

edit. 04.12.25

deton24's

Instrumental and vocal & stems separation & mastering

(UVR 5 GUI: [VR/MDX-Net/MDX23C/Demucs](#) 1-4, and BS/Mel-Roformer in [beta](#)
[MVSEP-MDX23-Colab](#)/[KaraFan/Drumsep/SCNet/Apollo](#)
[x-minus.pro \(uvronline.app\)](#)/[mvsep.com/Colabs/MSST](#)
[Gaudio/Dango.ai/Audioshake/Music ai](#))

[General reading advice](#) | [Discord](#) (ask, or suggest edits only there) | Table of [content](#) (up-to-date in Options>Document outline on PC or in the app) | [Training](#)

Straight to currently the best [models list](#)

Last updates and news

- cyatarow with unwa found a way to use MSST natively on Windows on RX 9060 using ROCm and released PyTorch for ROCm for Windows without having to use WSL - [click](#)
- Instruction for MSST-WebUI has been made too - [click](#)
- Our user had some success in porting our Colabs to Keggle. Inference Colab and Apollo ones have been made so far, but the porting process seems to be rather straightforward. Be aware that it might gradually fall behind with any newer models published in later periods.

<https://www.kaggle.com/code/zzryndm/music-source-separation-training-inference-webui>

<https://www.kaggle.com/code/zzryndm/apollo-colab-inference-i-fucking-give-up-with-this>

How it was done

"I didn't even have to debug anything. Pasting the Colab's code without changes worked miraculously.

Kaggle doesn't support markdown IN code cells but I found a workaround using U+200C for the variable names.

Also set the acceleration to GPU P100. It's better than t4x2. I learned that through the "patience is a virtue" route.

And trust me when i say it was NOT fun" - ryn (xxml)

- Gabox voc_fv7 beta 3 added to the [Colab](#)
- (MVSEP) "Four new models have been added:
1) MVSep Percussion (percussion, other) Demo:
<https://mvsep.com/result/20251128141738-f0bb276157-mixture.wav>
2) MVSep Keys (keys, other) Demo:
<https://mvsep.com/result/20251128142835-f0bb276157-mixture.wav>

For help and discussion, visit our Audio Separation Discord: <https://discord.gg/ZPtAU5R6rP> | Download [UVR](#) or [MSST-GUI](#)

For inst/voc separation in cloud, try out free Colabs: [BS/Mel-Roformer](#) | [MDX23](#) (2-4 stems) | [MDX-Net](#) | [VR](#) | [Demucs 4](#) (2-6)

3) MVSep Brass (brass, other) Demo:

<https://mvsep.com/result/20251128142905-f0bb276157-mixture.wav>

4) MVSep Woodwind (woodwind, other) Demo:

<https://mvsep.com/result/20251128143157-f0bb276157-mixture.wav>

- Unwa BS-Roformer HyperACE inst model has been added to a [separate Colab](#)

- Gabox released two new models (vocal and instrumental):

Mel [vocfv7beta3 | yaml](#)

Voc. fullness 21.82 | bleedless 30.83 | SDR 10.80

“beta 1 and 2... eh, pretty close to same instrumental bleed, but beta 3 def a step up from the two songs I compared (...) most songs so far, fv7beta3 is fuller than fv7beta1, def less robotic sounding at times (when a voice gets quiet/hard to capture, and it just fails). Just had another song where fv7beta1 was fuller than fv7beta3, but it was also a lot noisier large majority of the songs I tested, fv7beta3 was fuller... I think fv7beta3 is usually a bit noisier than fv7beta1? But also sounds fuller in those cases, I'd say it's generally worth it instrumental bleed, usually worse with fv7beta3 versus fv7beta1, but it depends fv7beta2 is always less full/less noise, but only slightly less instrumental bleed than fv7beta1” - rainboomdash

Mel [inst_fv7b | yaml](#)

Inst. fullness 27.07 | **bleedless 47.49** | SDR 16.71

“this may be the last beta before the final model” - Gabox

The highest bleedless metric out of all instrumental models so far. But fullness is worse than even most vocal Mel-Roformers (including BS-RoFormer SW and Mel Kim OG model).

“On the fuller side, somewhere around inst v1e+, maybe a tiny bit below. The main thing I notice is it captures more instruments than v1e+, but isn't muddy like [inst Resurrection] (which also captures more instruments) (...) It can add a lot of crackling noise, though, more than v1e+ (...) can be a little on the noisy side sometimes... but it at least isn't muddy and sounds natural (...) I'd still ensemble if you want the noise reduced - rainboomdash

([src](#))

- Unwa released [BS-Roformer-HyperACE](#) instrumental model | [separate Colab](#)

Inst. fullness 36.91 | bleedless 38.77 | SDR 17.27

(less fullness than v1e+: 37.89, but more bleedless: 36.53, SDR: 16.65)

Note: It uses its own inference script. “You can use this model by replacing the MSST repository's models/bs_roformer.py with the repository's bs_roformer.py.”

To not affect functionality of other BS-Roformer models by it, you can add it as new model_type by editing utils/settings.py and models/bs_roformer/init.py [here](#) (thx anvuew).

For error while installing the py file for HyperACE model in Scial's WebUI:

from models.bs_roformer.attend import Attend

ModuleNotFoundError: No module named 'models'"

The fix: "SUC-DriverOld/MSST-WebUI use the name "modules" and ZFTurbo/Music-Source-Separation-Training use the name "models". And Unwa's bs_roformer.py that you replace with, also use "models". So you'll have to do some coding and symlink to make it work." - fjordfish

"Currently, this model holds the highest aura_mrstft score on the instrumental side of the Multisong dataset. (...)

Some components in the SegmModel module were implemented based on this paper:

<https://arxiv.org/abs/2506.17733>

Simply put, it's a module that utilizes hypergraphs to capture global relationships and standard convolutions to capture local relationships, thereby generating the final "Correlation-Enhanced" feature map.

This weight is based on the following weights. Thank you, anvuew!

<https://huggingface.co/anvuew/BS-RoFormer>" - unwa

- The inference file got updated to fix error

Consider changing overlap from default 4 to 2 in the yaml of the model. The difference won't be really noticeable for most people, but it will be faster.

"thirty minutes of audio [on 4090]:

FNO was 71.38 seconds, HyperACE was 120.37

so HyperACE is about 2x longer than FNO (...) Does seem like HyperACE is picking up more instruments than v1e+

does seem like slightly worse vocal bleed overall (still need to test this more, though)... haven't encountered the super tinny vocal bleed like v1e+, at least

still fails to pick up that brass instrument on one song... Not really any worse than v1e+, though (...) resurrection inst does sound more muddy, but also a lot less noise... which makes sense... IDK, a little muddy for my tastes.

I did find one song/spot and resurrection inst was on par with HyperACE in picking up the wind instrument, v1e+ lost it for a bit.

I have found in the past that resurrection inst generally picks up more instruments than v1e+ (...) fullness of HyperACE is much closer to v1e+ than resurrection inst (...) it gets pretty staticy compared to v1e+ [on some drums] (...) v1e+ does this to a lot less extent it's not super common, though... (...) I'm very confident in saying HyperACE picks up more stuff than v1e+.

Resurrection inst does pick it up much better than v1e+, but I think it's still too quiet resurrection inst really does just pick up so much more instruments, despite having a lot less fullness" - rainboomdash

"fullness that is comparable to v1e+, but has significant more vocal crossbleeding in instrumental than BS Roformer Resurrection Inst, but still less than v1e+ and v1e" - dca100fb8

"the best instrumental model ive ever heard

Unbelievable how realistic it sounds

especially with bass and piano - PezZHasACat/pezz23

Might have problems with flute in specific songs - Hen

- (MVSEP) "We have released a new model 'MVSep Lead/Rhythm Guitar (lead-guitar, rhythm-guitar) '. It has two variants:
 - 1) Two-stage model (SDR: 9.21) - Best guitar model applied, and then 2-stem model is used which can separate lead/rhythm guitar.
 - 2) One-stage model (SDR: 9.02) - Single model is applied, which was trained on a 3 stem dataset.

They can give pretty different results, so worth trying both.

Demo: <https://mvsep.com/result/20251120090832-f0bb276157-mixture.wav>

- (MVSEP) We have released the "MVSep Plucked Strings (plucked-strings, other)" model.
 Demo: <https://mvsep.com/result/20251120092757-f0bb276157-mixture.wav> - ZFTurbo

- fr4z49 reported that they managed to use MSST with ROCm 7 and 6 on Linux and AMD RX 7600 for fast separations. Officially, it's not supported by AMD, but works, although your mileage might vary from GFX to GFX (range of GPU models inside various generations/archs).

- Probably, 5700 XT with some older ROCm versions (e.g. older than 6.33) might work too (e.g. HIP 5.7 and ROCm around 5.2.* - src, although you can try out 6.21 or 6.2.x to ensure, as it could happen that some earlier 6.x wasn't supporting RX 5700 XT correctly, while e.g. for RX 6000 ROCm 6.24 worked in some apps at some point, but more up-to-date information might be found in some ZLUDA guides, as it needs ROCm too - some suggestions here).

- The most performance gains on ROCm 7 might be potentially observed on officially supported GPUs like Instinct MI350 CDNA 4, providing even 3-7x performance gains over 6.0 in some applications (more).

- Official support for RX 400/500 (a.k.a. Polaris/GCN 4/GFX803) GPUs support was dropped, but you can follow this repo for unofficial ROCm 6 support.

Or for ROCm 5, this Ubuntu guide (it might even potentially work from Windows using WSL [if using at least Ubuntu 22.04 LTS] with almost no GPU performance overhead). There seemed to be some issues building Torch (UtilsAVX512.cc/tensorpipe) on Python 3.13, fixed on Python 3.10, and maybe 3.11.9.

Also, there seems to be some Arch Linux community package to install Pytorch still compatible for these GPUs (click).

Or also might be potentially supported with some other specific versions of ROC, e.g. 5.7.2 and also described above:

`export ROC_ENABLE_PRE_VEGA=1` (deprecated in ROCm 6; might help for lacking dependencies or wheel building issues). Or check out also this:

<https://github.com/AUTOMATIC1111/stable-diffusion-webui/issues/10435#issuecomment-1555399844>, or alternatively follow below instructions:

<https://pytorch.org/get-started/locally/> and then execute:

`pip3 install torch torchvision torchaudio --index-url`

<https://download.pytorch.org/whl/rocm5.4.2>

- Since then also ROCm 6.4.4 allowing using PyTorch natively on Linux and Windows on RX 7000 and 9000 was released ([more](#)), but it wasn't tested yet ([DL](#)). You might get the [error](#) using at least WebUI.
- Also, you might want to experiment with using ZLUDA in UVR (CUDA>ROCM translation layer - some suggestions [here](#)).

>fr4z49 ROCm report:

- "I managed to make [MSST-WebUI](#) work [on Linux] with:

Torch 2.10.0.dev20251110+rocm7.0

on RX 7600

(...) it seems like ROCm 7.0 is about a second faster [than 6.x]"

(probably by adding just pip install before it)

Turns out that if you do:

export TORCH_ROCM_AOTRITON_ENABLE_EXPERIMENTAL=1

it uses waay less VRAM and processes even faster.

inst_V1e_plus batch_size=2 overlap=3 chunk_size= 485100, 51.78s/it [3:50 of audio in 61 seconds]

For ROCm 6.x (a tad slower, might work on more GPUs) use:

torch 2.9.0+rocm6.3 torchvision0.24.0+rocm6.3 [-index-url

https://download.pytorch.org/whl/rocm6.3]

Or older version suggested above.

Thanks, fr4z49.

- yxlilc's harmonic noise separation VR (6 or 5.x model, unsure) if someone was interested:

https://github.com/yxlilc/vocal-remover/releases/tag/hnsep_240512 (July 2025)

- Gabox released beta 2 of vocfv7 Mel-Roformer "fullness went down a little bit"

Voc. bleedless: 31.55, fullness: 20.44, SDR: 10.87

<https://huggingface.co/GaboxR67/MelBandRoformers/blob/main/melbandroformers/experimental/vocfv7beta2.ckpt> | [yaml](#) | [Colab](#) (TL;DR in the vocals section)

"still quite a bit fuller than big beta 6x, but has less noise than even fv4 (also a bit less fullness, of course)" at least when the instruments are loud, fv7beta2 is usually quite a bit less noisy than fv4, while still maintaining a decent amount of fullness... it is a bit less, but not too much (...) both are pretty noisy with fv4 (...)

still gonna have an issue with backing vocals compared to fv7beta1 sometimes... (makes sense, it's a less full model). (...) Fv7beta2 has still been significantly better with BV than fv4, despite quite a bit less noise" but "significant issues on one song, while fv6/fv7beta1 didn't (...) Def an improvement over fv4. I'm really liking the balance of fullness and noise for most songs. fv4 and fv6/fv7beta1 are usually pretty noisy... this is less noisy, but still has a good amount of fullness." "Where the noise was undesirable and I ensembled fv4/fv6/fv7beta1 with big beta 6x, now I can just use this instead".

"Fv7beta2 has still been significantly better with BV than fv4, despite quite a bit less noise" but "significant issues on one song, while fv6/fv7beta1 didn't"

"If the noise isn't an issue, and you just want fullness, fv6/fv7beta1 are still the best models. I'd say fv6 and fv7beta1 are better models than fv4, fullness/noise aside. It depends with fv7beta1 versus fv7beta2, sometimes the noise can be pretty significant with fv7beta1, and fv7beta2 may have the fullness you desire."

fv6 is usually more noisy/full than fv7beta1, but it just depends... I've had instances where it's less noisy/full than fv7beta1. But if you really want high fullness, fv6 and fv7beta1 are the choices. Sometimes fv6 can be quite a bit more noisy and the gain in fullness isn't worth it".

- rainboomdash (thx).

"vocals sound very robotic with those models, however. Compared to fv4" - pipedream

- Anvew released a new BS-Roformer vocal model:

<https://huggingface.co/anvew/BS-RoFormer>

It also doesn't work on the UVR's RTX 5000 patch - then use [MSST](#) instead.

On an M1 Mac, you will probably need to decrease chunk_size in the yaml a bit.

"On one song it was on par with the 2025.07 BSRoformer model on MVSep, at least to my ears (Tentative by System Of A Down)

The other song [Linkin Park - Part of Me] has some background vocals that are hard to get for a lot of voc/inst models, 2025.07 manages to get them while this model doesn't.

The instrumentals and vocals seem pretty good other than that" - ryanz48

"Guessing it's not high enough fullness

the [lost](#) is extremely muddled, and the other chanting is just gone.

The lower harmony at 0:49-0:53 is mostly gone, making it sound very thin and a lot of the vocals just sound like they are breaking apart.

Hmm, big beta 6x is significantly better, it's def a fullness issue.. probably too high of a bleedless model for my tastes

big beta 6x still isn't super great here, the one I used for the other one I posted was fv7beta1, which is a fullness model.

yeah, big beta 6x seems more balanced, it's a bit fuller but not noisy to my ears, either but I'm not using headphones, so I won't hear any minor noise easily.

yoooo, it properly doesn't capture the instrument [here](#).

even FT2 bleedless gets tricked by this part, but this does just fine.

Maybe I'll try it out for this song.. most I'll still use higher fullness models" - rainboomdash

- Anvew's BS-Roformer Karaoke Model added to the inference [Colab](#)

- (MVSEP) "Eleven new models have been added:

1) MVSep Triangle (triangle, other) Demo:

<https://mvsep.com/result/20251104082053-f0bb276157-mixture.wav>

2) MVSep Sitar (sitar, other) Demo:

<https://mvsep.com/result/20251104082317-f0bb276157-mixture.wav>

3) MVSep Harpsichord (harpsichord, other)

Demo: <https://mvsep.com/result/20251104082809-f0bb276157-mixture.wav>

4) MVSep Tuba (tuba, other) Demo:

<https://mvsep.com/result/20251104082845-f0bb276157-mixture.wav>

5) MVSep Bassoon (bassoon, other) Demo:

<https://mvsep.com/result/20251104083211-f0bb276157-mixture.wav>

6) MVSep Congas (congas, other) Demo:

<https://mvsep.com/result/20251104083239-f0bb276157-mixture.wav>

7) MVSep Bells (bells, other) Demo:

<https://mvsep.com/result/20251104083305-f0bb276157-mixture.wav>

8) MVSep Ukulele (ukulele, other) Demo:

<https://mvsep.com/result/20251104083332-f0bb276157-mixture.wav>

9) MVSep Dobro (dobro, other) Demo:

<https://mvsep.com/result/20251104115825-f0bb276157-mixture.wav>

10) MVSep Wind Chimes (wind-chimes, other) Demo:

<https://mvsep.com/result/20251104115849-f0bb276157-mixture.wav>

11) MVSep Accordion (accordion, other)

Demo: <https://mvsep.com/result/20251104115916-f0bb276157-mixture.wav> - ZFTurbo

- "Yeah, I just tried [the bells] on a drum loop sample with sleigh bells I wanted to isolate, and I got a rude awakening lol"

- "That's why that model name is confusing, lol. What they mean is tubular Bells or chimes. There's currently no sleigh bells model, but the Tambourine model may work"

- "It worked (...) I used drumsep on it before"

"I finally was able to extract the lead guitar in this song using the dobro model, but i noticed how the bass synth is leaking in the dobro stem"

"So I thought, what if I just remove the bass in the source and try again? I did, and now it doesn't pick the lead guitar anymore"

- (uvronline) "Added two new models:

BS-RoFormer Kar (anvuel)

De-reverb Room (anvuel)" - Aufr33

The latter is also added to the [Colab](#).

- (MVSEP) "We added MVSep Synth (synth, other) model. Synth is included the following stems: Synth, Synthesizer, Synth Pad, Synth Bass, Synth Vocals, Synth Strings, Synth Percussion, Synth FX, Synth Keys, Synth Brass, Synth Guitar, Synth Flute, Synth Ambiant.

Demo: <https://mvsep.com/result/20251031214429-f0bb276157-mixture.wav> - ZFTurbo

"So, my initial thoughts. The model works great for certain kinds of sounds e.g. leads, pads, plucks. But it's tricky to predict what it'll do, so might be safer to get rid of the other stems first, and then using synth on what's left if you need further separation.

Some examples:

It isn't picking up synth basses for me, so use BS-Roformer SW for that.

It also sort of picks up synth brass, but wind model is catching that better, at least on the stuff I ran. The same could possibly be said for synth guitars." - Musicalman

"I tested it on a front channel rip from a cue from the TV show CHUCK, and it ripped the synth lead. But it did not isolate the synth "effect" at the end of it" - fal_2067

"is good, but sometimes it can't detect some bass synths for some reason, still not a big problem since an SW does the work almost every time. And sometimes it picks the strumming of guitars. And also it seems to fail if they are vocals or harmonies." - smilewasfound

"Most of the time gets more out of the song rather than taking out guitar, keys, bass separately until synths are left over. Sometimes very few misses or stem bleeds through, but overall very impressive!! It also picks up Vibraphone" - Tobias51

"Synth stem seems way too muddy on full songs, (...) But much better than I was expecting, I'll be honest. It messes up a lot on full songs, it seems. It seems like it's more like a stem remover than an isolator to me. The no synth stems sounds very clean. Tried it on Dolby stems, same thing, the no synth stem was very clean, synth stem sounded a bit muddy though. That's fine though. Gets very confused with bass thought.

Seems to also have a lot less of the classic MVSep phase issue, where for some reason half the stem is in the synth stem and the other half is in the no synth stem and inverting it cancels it out (literally almost all models have this issue on MVSep it's very strange). It's much less than the other models, but yeah, it still happens. (...) Using any other model in UVR or uvronline don't have this issue. (...) I put a bass guitar stem from Fortnite festival, it worked exceptionally well. Not much muddiness at ALL, really good. Separated the synth bass stem from bass really well. Ok wait, it seems very freaky, extremely freaky. It has the phasing issue. Yeah, WTF. It inverts" - Isling

"I put it the synth model through a really packed song with no synth to see if it would get tripped up, it didn't other than some bass at the end.

Which actually didn't get picked up by the bass model, so even that is a win" - dynamic64

- "Several SATB (soprano, alto, tenor, bass) choir models I trained ages ago, currently only a scnet_masked model is available, but I did have Demucs, MDX23C, and standard SCNet models that I will upload to this link when/if I find them, although I'm pretty sure the scnet_masked model was the best in the end:

<https://drive.google.com/drive/folders/1BpPgtlDk0yqrlArmrq9vnYErixb8l8zJ?usp=sharing> -

Dry Paint Dealer

Treat it as proof of concept.

"I've tried the SCNet one, it's really noisy, and it has a lot of bleed, it kinda works. I can see the potential on this kind of model ngl." - smilewasfound

"You can't install [the VR ones] into UVR since that only supports VR v5 [and 5.1] not [VR v6](#)"

- (MVSEP) Seven new models have been added:

1) MVSep Electric Guitar (electric-guitar, other). Demo:

<https://mvsep.com/result/20251031064813-f0bb276157-mixture.wav>

2) MVSep French Horn (french-horn, other). Demo:

<https://mvsep.com/result/20251031072529-f0bb276157-mixture.wav>

3) MVSep Banjo (banjo, other). Demo:

<https://mvsep.com/result/20251031095934-f0bb276157-mixture.wav>

4) MVSep Marimba (marimba, other). Demo:

<https://mvsep.com/result/20251031100024-f0bb276157-mixture.wav>

5) MVSep Glockenspiel (glockenspiel, other). Demo:

<https://mvsep.com/result/20251031100134-f0bb276157-mixture.wav>

6) MVSep Timpani (timpani, other). Demo:

<https://mvsep.com/result/20251031100232-f0bb276157-mixture.wav>

7) MVSep Harmonica (harmonica, other). Demo:

<https://mvsep.com/result/20251031100508-f0bb276157-mixture.wav>" - ZFTurbo

“The Harmonica model is hit or miss.” - musicbybrooks

“Wow, the electric guitar model is really neat. One thing I noticed is that it seems to be better than other models at picking up midi/synth lead guitars. At least on stuff I tried.

I think it also gets tripped up a bit more by weird FX and synth sounds being partially flagged as guitar. An interesting model, though, for sure.” - Musicalman

- (MVSEP) “The karaoke model by anvuew has been added under the algorithm "MVSep Karaoke (lead/back vocals)". It is available as the option "BS Roformer by anvuew (SDR: 10.22)" - ZFTurbo

For some reason it seems to give worse results than the ckpt anvuew shared.

- Dear friends at Apple Music. Please stop harassing labels and their sound engineers for making Atmos mixes using our and yours awesome AI models for audio separation. The artificial artifacts you're solely looking for in spectrograms in separate channels are inaudible in the entire tracks. The tracks are well mixed and accepted by major labels, but rejected by your lazy ass incompetent bullshit. The quality of Atmos mixes got better since the very beginning, and either your employees, or your algorithms, or both, do a lazy job without even hearing the shit on their own, while still rejecting stuff without sensible reason! You're making things nasty difficult for artists who lost their multitracks for certain legacy songs, rendering re-releasing of their albums in Atmos potentially impossible. Bring it up with the executives. Get your shit together, for fuck sakes!

- Full release of mesk's rifforge Mel-Roformer [model](#) focused on inst/voc separation for metal music

“The model can have some quirks (just like most models) but it's all around clean for me to release.”

Training details:

“Characteristics:

This is a dimension 512 depth 24 model (so fairly large file size at 1.9 GB!), with an SDR of 14.2436.

It's finetuned from an older Melband Roformer checkpoint with an SDR of 13.7.”

- Gabox released experimental BS-Roformer karaoke [model](#) | [metrics](#)

It gives the same error for RTX 5000 UVR patch users as the avuew's model.

- New ensemble (avg) of anvuew's and becruily & frazer karaoke models was evaluated on the leaderboard ([metrics](#) lower than BSkarfrazerBecruily+BSkarMVSEP+MBkarGaboxV2 SDR-wise). Probably you could make a [fusion model](#) out of the two to save on inference time in cost of slight SDR decrease (both use the same config so it might work).

- erosunica found out that BS-Roformer SW drums is “really good to remove some SFX and foley, way better than DnR v3”

- Gabox released voc_fv7 beta 1 Mel-Roformer [model](#) | [yaml](#) | [Colab](#)

Voc. fullness: 21.21, bleedless: 30.81, SDR: 10.96

"Just a better fv4 it seems, better bleedless" (fullness: 21.33, bleedless: 29.07, SDR 10.58) vs voc_fv4 "It is noisier.. Kinda closer to beta 5e?" "It's slightly less noise and fullness than beta 5e but picking up the backing vocals REALLY well, significantly better than beta 5e" But it's pulling the backing vocals out even better than 5e" "the backing vocals are so good! "it does have significant synth bleed, too... it at least wasn't coming through at full volume when I say fullness, I specifically mean how muddy it sounds" - Raiboom Dash

- Anvuew released a new Karaoke BS-Roformer model

https://huggingface.co/anvuew/karaoke_bs_roformer

https://mvsep.com/quality_checker/entry/9180

UVR users will encounter “ModuleNotFoundError: “No module named ‘torch._dynamo.polyfills.fx’” with this model (consider [MSST](#) instead). ~~Maybe users of RTX 5000 won’t encounter that issue due to newer PyTorch in dedicated patch.~~ Sadly not - even with CPU only. Even more, the issue seems to exist only on RTX 5000 patch.

“karaoke anvuew extracts lead vocals a bit better than karaoke becruily frazer, and in some parts, the lead vocals from karaoke anvuew still sound brighter compared to karaoke becruily frazer, which sounds a bit more compressed. oh, and for some reason, the becruily frazer model doesn’t detect vocals with radio effects, while anvuew’s model handles them just fine” - neoculture

“lead vocals leak into instrumental (...) Mel Becruily and Frazer’s BS don’t have this problem” In that case, maybe “isolate the acapella first in almost all cases of using a karaoke model” Demo:

<https://pillows.su/f/671f60ffdd615eb2613c78dca70319fe>

<https://pillows.su/f/391ec7ba8353a086989c9c0934321260>

- (MVSEP) “I added a new DeReverb model https://huggingface.co/anvuew/dereverb_room by avuew. It’s available in Reverb Removal (noreverb) [by choosing] DeReverb room by anvuew (BSRoformer). It works only for vocals. Since it is a mono model, it processes 2 stereo channels independently.” - ZFTurbo

Demo: <https://mvsep.com/result/20251017064532-53be20aa17-10seconds-song.wav>

- (MVSEP) Four new models have been added:

1) MVSep Tambourine (tambourine, other). Demo:

<https://mvsep.com/result/20251015221411-f0bb276157-mixture.wav>

2) MVSep Oboe (oboe, other). Demo:

<https://mvsep.com/result/20251015221618-f0bb276157-mixture.wav>

3) MVSep Clarinet (clarinet, other). Demo:

<https://mvsep.com/result/20251015221718-f0bb276157-mixture.wav>

4) MVSep Digital Piano (digital-piano, other). Demo:

<https://mvsep.com/result/20251015221944-f0bb276157-mixture.wav>

Sometimes it can also work better for normal piano and make even better work then SW if it works.

“Absolutely fantastic for an epiano. i just put your example song through mvsep with the bs sw piano model as a comparison and bs sw did terribly. Ofc your epiano model picked up all of the epiano in a clean way”

“Pretty impressive. Besides it being more full than other piano models (in most cases), it's also by far the only piano model that doesn't mistakenly pick up other instruments like tubular bells as piano.”

“From what i tested is more for midi piano, i tested with some tracks with that kind of midi sound and it worked way better than SW.”

- (training) Becruily made a modification of [dTtnet](#) arch working in MSST ([DL](#)).

“They report very good performance on vocals with low parameters” - Kim

Back in the end of 2023, one indie pop song from multisong dataset (of the two there) received the best SDR - Bas Curtiz

“Better than SCNet imo, remains to see if it can beat rofos” - Becruily

“Not fast to train. I'm back with vanilla mdx23c. Trying a config to train model with less than 4GB VRAM, (...) with my 1080Ti and batch_size=1, chunk_size is around 1.5sec” - jarreou
Installation instruction:

“In the latest MSST [at least for 13.10.25]

add the ddttnet folder to "models" and replace your settings file in utils with this”

The mod breaks compatibility with the authors' checkpoint.

“The weird thing is, it sounds like a fullness model despite not being one, I barely can find dips in instrumentals. [ddtnet vs kim melband](#), if anyone is curious” - bcr

“Also keep in mind authors trained with L1 loss only, default in MSST is masked loss”

“L1 loss when dataset is noisy, mse loss when dataset is clean”

“the loss is defined from msst, but in the original dttnet it was in the code itself you can just --loss L1_loss”

@jarredou “I copied your tfc and tfc_tdf classes to my files (and used that latest stft/istft I sent) - and seems to be better, just like the og dttnet

the tfc/tfd fixed the nan issue for me (...)”

Keep in mind, ddttnet was trained only with musdb and has 10-20x less params while being comparable in quality”

“the authors checkpoints had 16khz cutoff because dim_f was smaller than nfft/2”

if you want to train model with cutoff it's fine, if you want fullband then dim_f must be half of nfft + 1" - becruily

Hit our [#dev-talk](#) for more.

- New sites added to [Site and rippers \(deezmate.com and tidal.qqdl.site\)](#).

Qubuz remains or defunct/problematic for now.

- We have numerous reports about some models like Unwa Resurrection inst having problems on AMD (and probably Intel GPUs) in UVR, returning "Invalid parameter" error. In that case, uncheck GPU Conversion (but it will be slower). If you find a fix, please let us know on the Discord (link at the top of the doc).

- if you deal with slow separation times on becruily & Frazer karaoke model, decrease chunk_size to 160000 on 8GB GPUs. As long as decreasing chunk_size on CUDA (NVIDIA) doesn't seem to affect separation times, it's not the case with DirectML (AMD/Intel), if you're exceeding your VRAM, but it still doesn't crash.

- Added anview BS-Roformer Dereverb Room (mono) [model](#) to the inference [Colab](#)

- Sir Joseph released a Colab for A2SB: Audio Restoration NVIDIA's upscaler.

It's very slow - on 4070 Super it was slow already, and in free Colab we got Tesla T4 with RTX 3050 performance with 12GB of VRAM instead of 8 - memory issues might occasionally occur, Colab Pro recommended.

Only inpainting doesn't work (feature for filling silences if exist or missing parts) - "I couldn't fix the error. If anyone solves it, I'd be glad if they let me know so I can update it too.". A2SB should rather surpass AudioSR, Apollo and FlashSR (it does at least metrically).

https://colab.research.google.com/drive/1ThenZDCRTJKV1I_ax17XGWmkB1goKrFs?usp=sharing

- Gabox released inst_fv4 model. Don't confuse it with inst_fv4noise - the regular variant was never released before (and with voc_fv4).

[https://huggingface.co/GaboxR67/MelBandRoformers/blob/main/melbandroformers/instrumental/inst_Fv4.ckpt\(yaml\)](https://huggingface.co/GaboxR67/MelBandRoformers/blob/main/melbandroformers/instrumental/inst_Fv4.ckpt%28yaml%29) | [Colab](#)

"Seems to be erasing a xylophone instrument. Does sound not too noisy and not muddy, I like it. (...) A little noisy with piano (I split the song up and process with resurrection inst there). (...) Does have some issues that resurrection inst doesn't have, but it doesn't sound muddy! It usually works great. (...) In my opinion, fv4 still has vocal traces, I don't know if in all of its songs and v1e plus doesn't have them, but the noise can bother you even though it's not much. Does have more vocal bleed at times. I think a lot of what I thought was vocal bleed was a synth, it did a pretty good job... There was one segment on a song where it caught vocal residues, though" - rainboomdash

- neoculture released a Mel-Roformer instrumental model focused on preserving vocal chops Inst. fullness 39.88, bleedless: 32.56, SDR: 14.35

[https://huggingface.co/natanworkspace/melband_roformer/blob/main/Neo_InstVFX.ckpt\(yaml\) | Colab](https://huggingface.co/natanworkspace/melband_roformer/blob/main/Neo_InstVFX.ckpt(yaml) | Colab)

“great model (at least for K-pop it achieved the clarity and quality that no other model managed to have) it should be noted that it has a bit of noise even in its latest update, its stability is impressive, how it captures vocal chops, in blank spaces it does not leave a vocal record, sometimes the voice on certain occasions tries to eliminate them confusing them with noise, but in general it was a model that impressed me. It captures the instruments very clearly” - billieoconnell.

“NOISY AF, this is probably the dumbest idea ever had for an instrumental model. Don’t use it as your main one, some vocals will leak because I added tracks with vocal chops to the dataset. Just use this model for songs that have vocal chops” - neoculture

It was trained on only RTX 4060 8GB.

- Aname Mel trained a Mel-Roformer model called [Full Scratch](#)

Inst. fullness: 25.10, bleedless: 37.13, SDR: 14.32

Voc. fullness: 13.24, bleedless: 30.75, SDR: 8.01

(“trained from scratch on a custom-built dataset targeting vocals. It can be used as a base model or for direct inference. Estimated Training cost: ~\$100”)

For state_dict error, update MSST to the last repo version:

!rm -rf /content/Music-Source-Separation-Training

!git clone https://github.com/ZFTurbo/Music-Source-Separation-Training

“and you must reinstall main branch's requirement.txt. (before it, edit requirements.txt to remove wxpython)” - Essid

Kim Mel model for reference:

Inst. fullness 27.44, bleedless 46.56, SDR: 17.32

Voc. bleedless: 36.75, fullness: 16.26, SDR: 11.07

- (MVSEP) “1) Model by baicai1145 was added in Apollo Enhancers (by JusperLee, Lew, baicai1145) with name Universal Super Resolution (by baicai1145) (...)

2) New option added for Apollo Enhancers (by JusperLee, Lew, baicai1145) - Cutoff (Hz). Sometimes it can be useful to cut higher frequencies before applying model.” - ZFTurbo

- baicai1145 released their own Apollo vocal restoration model, which surpassed Lew's vocal V2 model metrically.

“with a 92-hour high-quality vocal dataset trained for 1 million steps.”

<https://huggingface.co/baicai1145/Apollo-vocal-msst/tree/main>

https://mvsep.com/quality_checker/entry/9105

21.24 vs 13.09 Aura MR STFT

(thx Essid)

ReminderL Apollo arch support was added to UVR too (acceleration work with NVIDIA GPUs only). Installing the model should be possible also there, although currently UVR

seems to be incompatible with the model outputting following error:

KeyError: "infos"

- (MVSEP) “Two new algorithms have been added:

1) MVSep Mandolin (mandolin, other). Demo:

<https://mvsep.com/result/20250927132339-f0bb276157-mixture.wav>

2) MVSep Trombone (trombone, other). Demo:

<https://mvsep.com/result/20250927132547-f0bb276157-mixture.wav>” - ZFTurbo

- ROCm 6.4.4 now allows using PyTorch natively on Linux and Windows on RX 7000 and 9000, so you don't need WSL with them - [link](#)

- BS-Roformer 6 stems added on uvronline

- “I added new version of MVSep Organ (organ, other) model: “BS Roformer (SDR organ: 5.08)”. SDR increased from 3.05 to 5.08.

Demo:

<https://mvsep.com/result/20250924223759-0ef607228-song-organ-000-mixture.wav>” -

ZFTurbo

“I think the model is remarkable improvement” - totalmentenormal

“the result is great, no bleed from what I tested it on” - dynamic64

“much better isolation of the Hammond organs compared to the previous model. In places where the organ sound was not picked up before, it is now separated in the track” - lukasz2286

- Google Colab now allows pinning your environment to specific version having the same versions of packages, so maybe your notebook won't break in the future due to changes in the environment introduced by Google in the Colab with package updates.

For now there is only 2025.07 and the latest environment to choose from, and it's hard to tell if e.g. 2025.07 environment will be gradually replaced along the time while new changes to the latest Colab environment will be made:

<https://developers.googleblog.com/en/google-colab-adds-more-back-to-school-improvements/>

To use it, go to Environment>Change environment type>Environment type version and choose 2025.07 option

- Essid reevaluated GAudio (a.k.a. [GSEP](#)) for the leaderboard.

https://mvsep.com/quality_checker/entry/9095

Inst fullness: 28.83, bleedless: 31.18, SDR: 12.59

The result would rather cover my observations that instrumentals rather have gotten worse over the years (at least since the last 2023 Bas' evaluation or even earlier, at least for certain songs). But it appears that the vocals might got better.

https://mvsep.com/quality_checker/multisong_leaderboard?algo_name_filter=Gsep&sort=instrument&ranking_metrics=

Despite the fact the metrics are worse than even the least bleedless free community models like even V1e, for specific songs where bleeding doesn't occur so badly, GSEP might be still interesting too try out to some limited extend, being a different architecture, sounding maybe less filtered. Also, mixdown of multi stem extraction instead, should rather have bigger bleedless metric, but since the appearance of instrumental Roformers, GSEP relevance for separation is rather faded.

- "Ensemble of 3 [karaoke] models "Mvsep + gabox + frazer/becruily" gives 10.6 SDR on leaderboard. I didn't upload it yet, but I had local testing." - ZFTurbo

- fabio06844 shared his method for "very clean and full" instrumental lately.

1) Go to MVSep and separate your song with the latest Karaoke BS-Roformer by MVSep Team

2) On its instrumental stem use DEBLEED-MelBand-Roformer (by unwa/97chris)

([model](#) | [yaml](#) | [Colab](#))

Despite the fact that "the MVSep Team Karaoke uses the MVSep BS model to extract/remove vocals, then applies [the] karaoke model to that", it was told to be not enough to just use BS 2025.07 model instead, leaving a little more residues.

- Aname released Mel-Roformer duality [model](#).

"it's odd why the model is named duality, but it has a single target (and the file size of the ckpt confirms it further)" - becruily

It's focused more on bleedless than fullness metric contrary to the unwa's duality v2 model, but with bigger SDR.

Inst. fullness 24.36, bleedless 46.52, SDR: 17.15

"instrumental is really muddy" - Gabox

For comparison -

Mel Duality v2 by unwa

Inst. fullness 28.03, bleedless 44.16, SDR: 16.69

MelBand Roformer vocals by Kim

Inst. fullness 27.44, bleedless 46.56, SDR: 17.39

Instrumental public models with the biggest fullness metric -

Gabox Mel Roformer Inst_GaboxFv7z

Inst. fullness: 29.96, bleedless: 44.61, SDR: 16.62

Unwa BS-Roformer-Inst-FNO

Inst. fullness: 32.03, bleedless: 42.87, SDR: 17.60

- (MVSEP) "I added new SCNet vocals model: SCNet XL IHF (high instrum fullness by becruily). It's high fullness version for instrumental prepared by becruily."

Inst. fullness 32.31, inst. bleedless 38.15, SDR 17.20

"One of my favorite instrumental models, Roformer-like quality.

For busy songs it works great, for trap/acoustic etc. Roformer is better due to SCNet bleed" - becruily

"It's better than BS Roformer (mvsep 2025.07 and inst Resurrection) at low frequencies, but bad at highs due to the bleeding. I think it has better phase understanding, because it keeps the harmonics that were masked behind vocals cleaner (but it might not necessarily be true to the source, it might just interpret/make up the harmonics instead of actual unmasking)" - IntroC

- Dry Paint Dealer Undr released Melband Roformer and Demucs ("or at least I think this is the correct model file") Lead and Rhythm guitar [models](#).

"My own very mediocre model for it that I never shared. It does work but has issues that I imagine any better executed model won't."

"Wait, I think it separated doubles in vocals" - isling

Demucs model doesn't work in UVR, as it was trained on MSST, and not the OG code (I tried to workaround the bag_num issue [before](#), and failed)

- (MVSEP) "Two new instrumental models have been added:

MVSep Harp (harp, other)

Demo: <https://mvsep.com/result/20250921131108-f0bb276157-mixture.wav>

MVSep Double Bass (double-bass, other)

Demo: <https://mvsep.com/result/20250921131129-f0bb276157-mixture.wav>" - ZFTurbo

"The BS-Roformer SW bass model should probably be used first to extract the double bass. Creates a better sound. This does not apply to bowed double bass.

Bowed double bass doesn't get picked up by BS-Roformer and therefore needs the double bass model. Good news is that bowed double bass is picked up in the strings stem so if you run the strings model you're good either way." - dynamic64

- (MVSEP) "A new saxophone model based on BSRoformer was added. It has a much better metric compared to the previous [model]. SDR grew from 7.13 up to 9.77.

It's available in "MVSep Saxophone (saxophone, other)" with option "BS Roformer (SDR saxophone: 9.77)

Demo: <https://mvsep.com/result/20250920151232-f0bb276157-mixture.wav>" - ZFTurbo

"after testing this on a song where trumpet and sax play in unison, doing the trumpet model is cleaner than doing the sax model" - dynamic64

"Amazing. Tested it on one song, it got every single Saxophone Part from the song it seems lit. Can hear one small little bitty part of it where it tries to come in off the Sax part, however I can barely hear it" - cali_tay98

- GAudio (a.k.a. GSEP) announced their SFX (DnR) model in their API:

"DME Separation (Dialogue, Music, Effects)"

So far it's not available for everyone on their regular site:

<https://studio.gaudiolab.io/>

But the link on their Discord redirects to the site with a form to write an inquiry:

<https://www.gaudiolab.com/developers>

Shortly after entering the one or both of the links and logging on the first, you might get an email that \$20 of free credits to access their API have been added to your account, and link to the API documentation:

<https://www.gaudiolab.com/docs>

- New Dango Karaoke model released

<https://tuanziai.com/en-US/blog/68ca20c87c8c85686c1b4511>

A lot of problems when songs don't have lead vocals in the center.

- (MVSEP) "I added new Karaoke model: "BS Roformer by MVSep Team (SDR: 10.41)" it's available under option "MVSep MelBand Karaoke (lead/back vocals)". [Metrics](#).

In contrast with other Karaoke models, it returns 3 stems: "lead", "back" and "instrumental".

Example: <https://mvsep.com/result/20250915192251-53be20aa17-10seconds-song.wav> - ZFTurbo

"If I had to compare it to any of the models, it is similar to the frazer and becruily model. Sometimes it does not detect the lead vocals specially if there's some heavy hard panning, but when it does, there is almost no bleed, and it works very well with heavy harmonies in mono from what I tested." - smilewasfound

"becruly & frazer is better a little when the main voice is stereo" - daylightgay

"On tracks I tested, harmony preservation was better in becruily & frazer (...) the new model isn't worse, I ended up finding examples like Chan Chan by Buena Vista Social Club or The Way I Are by Timbaland where it is better than the previous kar model. The thing is, with the Kar models, it's just track per track. Difficult to find a model for batch processing as it's really different from one track to another" - dca100fb8

"I also found the new model to not keep some BGVs, mainly mono/low octave ones, despite higher SDR" - becruily

"I think I've found a solution for people who don't like the new model.

If you put an audio file through the karaoke model and then put the lead vocal result through that, it usually picks up doubles.

Which you can then put in your BGV stem if you'd like" - dynamic64

"it's definitely not as good as the one by frazer and becruily. SDR can be misleading sometimes" - ryanz48

becruly ["our model] uses 11.9 SDR vocal model as a base"

ZFTurbo "I started from SW weights"

"I've had fantastic results with it so far. Much MUCH better at holding the 'S' & 'T' sounds than the Rofo oke (for backing vox). Generally seems to provide fuller results .. but also the typical 'ghost' residue from the main vox can end up in the backing vox sometimes, but it's usually not enough to be an issue. I won't go so far as to say that it's replacing the other backing vox models for me entirely .. but it feels like the best of both worlds that Rofo and UVR2 provide." - CC Karaoke

- (MVSEP) "We've added a mirror of MVSep (big thanks to okhostok):

<https://mirror.mvsep.com>

If you have a problem with upload/download speed or can't reach the main site then try the mirror.

Report please if it helped you to speed things up." - ZFTurbo

Some issues with being unable to click separate button for some users were fixed.

- Gabox released BS_ResurrectioN [model](#) | [yaml](#)

"It is a finetune of BS Roformer Resurrection Inst but with higher fullness (like v1e for example), it needs [MVSEP's] BS 2025.07 (as a source/reference) phase fix [so you "should process the instrumental result using BS 2025.07 then put [it] as source in UVR GUI phase fix tool"]. I requested it because I found some songs where Resur Inst was producing muddy instrum results (...) I requested it not just for me because I saw other people were looking for something like v1e++" - dca

- anvuew released BS-Roformer Dereverb Room [model](#) | [Colab](#)

"specifically for mono vocal room reverb." as most are recorded in mono.

Not that long inference compared to other Roformers.

"Really liking the fullness in the noreverb stem. Virtually all dereverb roformers I've tried sound muddy, but this one is just the opposite. (...) Other noises may interfere, and in my experience, makes the model underestimate the reverb. [The previous anvuew's mono model] is way different [from] this one in every way. So, like I say, worth a shot." -

Musicalman. "WOAH this is insane. This would go viral if someone implemented in a plugin"
- heuhew

We have reports about errors in UVR while using this model. Consider using [MSST](#) instead.

If you have stereo errors using MSST on stereo files, update MSST (git clone and git pull commands) or:

(it might work in your current version and not only in the linked repo too, but potentially the code will be located in a different line, the change will be pushed there later)

"Edit inference.py from my [repo](#) line 59:

Replace :

```
# Convert mono to stereo if needed
if len(mix.shape) == 1:
    mix = np.stack([mix, mix], axis=0)
```

by :

```
# If mono audio we must adjust it depending on model
if len(mix.shape) == 1:
    mix = np.expand_dims(mix, axis=0)
    if 'num_channels' in config.audio:
        if config.audio['num_channels'] == 2:
            print('Convert mono track to stereo...')
            mix = np.concatenate([mix, mix], axis=0)"
```

- jarredou

- BS-Roformer Karaoke [model](#) by becruily & frazer released | MVSEP | uvronline [Metrics](#) better than even fused model gabox + aufr33/viperx and SCNet IHF below).

Make sure you don't have the option "Vocals only" checked in UVR.

"After dozens of tests I can tell this (...) is the best (better harmony detection, better differentiation between LVs and BVs, sounds fuller, less background roformer bleed, better uncommon panning handling etc)" - dca

"it also can detect the double vocals" - black_as_night

It works the best for some previously difficult songs. Aufr33 and viperx model seems more consistent, but the new BS is still the best in overall - Musicalman

"my og Mel also catches some of the FX/drums, I guess quite a difficult one due to how it's mixed" - becruily

"it does do better on mono than previous

sometimes confuses which voice should be the lead, but all models do that on mono in the exact use-case I normally test" - Dry Paint Dealer Undr

"In my opinion, this model is in no way inferior to the ViperX (Play da Segunda) — it's really very good. (...) I noticed that in the separation, the first voice still appears mixed with the second. The second voice, however, stands out more, but not completely isolated—in some passages, it still appears alongside the first. In short: the model better separates the second voice, but still presents some mixing between them." - fabio5284

"The new karaoke model doesn't actually differentiate between lvs & bvs and there's some lead vocal bleeding in the instrumental stem" - scdxtheresvolution

Fixes and expansion to the dataset and retrain of the model possible in the future.

"The dataset isn't correctly labelled, so in some training examples it was literally training the model to treat the backing vocal as the lead" - frazer

VS the newer BS-Roformer MVSEP team model above: "sound isn't as clear, but it does an infinitely better job at telling lead/BGV apart"

Becruily:

"I want to remind something regarding my (and the frazer) models

they're made to separate true lead vocals, meaning either all of the main singer's vocals, or if it's multiple singers - theirs too

this means if the main singer has stuff like adlibs on top of the main vocals, these are considered lead vocals too - they go together

if there are multiple singers singing on top of each other, including harmonise each other, and if there are additional background vocals behind those - all the singers will be separated as one main lead vocal, leaving only the true background vocals"

think of them like concert ready models - the output instrumentals will be ready to play in cases where all main vocalists are going to sing on top of the karaoke instrumental

ps: and yes, double/stereo lead vocals are still lead vocals, they're not bgvs (only in rare cases)

ps 2: if there are two singers singing the same melody and they don't harmonise each other - the model will most likely consider both singers as one lead vocal (again in rare cases one singer could be left)

- (MVSEP) "New Karaoke model based on SCNet XL IHF was added on site in "MVSep MelBand Karaoke (lead/back vocals)". Name of model "SCNet XL IHF by becruily (SDR: 9.53, [metrics](#))". It has slightly worse metrics than the top Roformer model, but since it's different architecture it can give better results in some cases where the Rofo failed.

Demo: <https://mvsep.com/result/20250908072226-f0bb276157-mixture.wav>" - ZFTurbo lirc it's BVE or IHF unpublic ZFTurbo model retrain, and ckpt won't be public till further notice, as becruily said.

"SCNet is more bleedly in general despite me trying to reduce the leakage it's recommended for busy songs, often captures proper lead vocals better than Roformer. Another use case is to ensemble it with Roformer to improve fullness" - becruily
"Oh, might be related to the lead vocals panning, it seems this model doesn't like when it's not center (...) I'm indeed noticing this model works really great on some songs that the Mel Rofo Karaoke had trouble with (...) I noticed that, this model, instead of creating crossbleeding between LVs and BVs, make them both quieter. I prefer that compared to previous models Plus, it handle songs which have lead vocals in the sides and BVs also in the sides better"

To fix bleed in back-instrum stem, use "Extract vocals first, but, "I noticed a pattern that if you hear the lead vocals in the back-instrum track already (SCNet bleed), dont try to use Extract vocals first because there will be even more lead vocal bleed" - dca

"Separates lead vocals better than Mel-Roformer karaoke becruily. It's not perfectly clean, sometimes a bit of the backing vocals slips through, but for now, scent karaoke model still the most reliable for lead vocals separation (imo)

<https://pillows.su/f/df8c1791bceba5fe3ef6b16d310ec123>

<https://pillows.su/f/e1272a02c56e3d3eb7ba4007bbb0c4bd>" - neoculture.

"the model seems to handle mono vocals better than melband but isn't as clean, lot of bleed" (extract vocals first was also used to test this) - Dry Paint Dealer Undr

Since the Mel Kar Bechuily's model, the dataset is "larger" now, but still not "great", and it might get eventually fixed, becruily said.

- (MVSEP) Four "new models for independent instruments were added:

1) MVSep Viola (viola, other) Demo:

<https://mvsep.com/result/20250907234931-f0bb276157-mixture.wav>

2) MVSep Cello (cello, other) Demo:

<https://mvsep.com/result/20250907235225-f0bb276157-mixture.wav>

"quite impressive" - dynamic64

3) MVSep Trumpet (trumpet, other) Demo:

<https://mvsep.com/result/20250907235543-f0bb276157-mixture.wav>

"I can't get over how good the trumpet model is, it's so cleannn" - Shintaro

"trumpet struggles a bit on muted trumpet" - dynamic64

4) MVSEP Strings BS-Roformer (strings, other)

Demo: <https://mvsep.com/result/20250907225920-f0bb276157-mixture.wav>

The SDR has increased significantly compared to the previous MDX23C model, from 3.84 to 5.41. It is currently the best model on the leaderboard:

https://mvsep.com/quality_checker/leaderboard/strings/?sort=strings - ZFTurbo

“From some quick testing, it does not disappoint. Still playing with it, but atm it's exactly what I hoped for.” - Musicalman

“Yeah, I'm running some tests too with a few tracks that were really hard to separate, mostly ones with cellos or vocals that were too blended with the strings to isolate even with the latest inst/voc models, and it's been working out surprisingly well.”

- anview released experimental BS-Roformer vocal model (nfft 4096, stft_hop_length 1024 “so not that large”) with 12 SDR measured on musdb18hq dataset. Might be worth checking: [download](#).

11.60 SDR on the same test set was previously achieved by one of the first Mel-Roformers trained by Bytedance on musdbhq + 500 songs ([paper](#)), although it wasn't nfft 4096. It uses very high 1024000 chunk_size in the yaml, so consider decreasing it when having memory issues, 500MB ckpt size.

- introC released a python [script](#) to get rid of vocal leakage in v1e+ model

- iZotope released Ozone 12. Separation still has Spleeter-like quality, but “it's unclear what they use” - Spleeter references disappeared from their readme (jarredou).

“the stems sound very bleedly and not at all usable” - becruily.

A notable new feature working competitively is their Delimiter.

- Ableton received its own stem separation feature in Live 12.3. It's made in cooperation with Moises.ai. <https://www.youtube.com/watch?v=uSahY-HGKt4>

“doesn't even sound good” - isling

It doesn't use GPU, and has a slower High Quality setting too (single model for each stem opposing to default multi stem), but it can take even 20 minutes for 1 minute file on a slower CPU. At least default sounds more similar to Demucs than Roformers or SCNet archs, although files look like BS-Roformer judging by memory dump (model files are encrypted).

[Here](#) are the low default model stems metrics - e.g. vocals only 8.71 SDR, but HQ option has bigger SDR than public SCNet weights released by ZFTurbo in MSST repo, but they're on a bleedless metric side, fullness is lower than in the public undertrained SCNet XL [4 stem](#) model. Average SDR of the first 12 songs in the multisong dataset vs public SCNet XL: drums: 11.58 vs 11.22, bass: 12.25 vs 11.27 (thx jarredou).

“The boring thing is that you have to launch separation for each file manually (no batch processing). To nice things is that the separated stems are automatically saved individually in folder (no need to export them individually through Live's rendering and all issue that this can produce; different length...)” - jarredou

- It seems like we've received a step-by-step tutorial how to install the new Nvidia's upscaler: [click](#) (thanks Pipedream)

- "I added BS Roformer flute model. It's available in "MVSep Flute (flute, other)". It superior comparing to SCNet version. SDR: 9.45 vs 6.27. More than 3 SDR difference.

Example: <https://mvsep.com/result/20250830211041-f0bb276157-mixture.wav>" ZFTurbo

- Thanks to Essid, metrics for following instrumental models were added to the models [list](#): INSTV7N, inst_fv8 (v2), inst_gabox3, Rifforge model, older mesk's metal model, FVX, Bv1, Bv2 (b - bleedless, v - for version)

- "New Wind model based on BS Roformer has been added in MVSep Wind (wind, other):

Demo: <https://mvsep.com/result/20250829230056-f0bb276157-mixture.wav>

Results on quality checker: https://mvsep.com/quality_checker/entry/8933

It increased SDR +2.5 comparing to previous best model." - ZFTurbo

"this one does not disappoint. At least not with the stuff I've tried so far. (...) the improvement is most noticeable with orchestral music. In heavy mixes eg. with lots of strings, the old models trip out. [The] new one is a lot more robust." - Musicalman

"the model is not only cleaner but also detects some wind instruments that the previous one couldn't (specially baritone saxophones, I need to test it a bit more)" - smilewasfound

"the bs roformer wind model does really well with the other result and the violin model really is quite useful" - dio7500, dynamic64

- Suno now has stem separation feature "t's generative, so the separation isn't exact. Also, you apparently can't use it on like famous songs because they'll get flagged." - Musicalman
"it sounds like shit tbh, tried it out" - dynamic64

- Gabox released experimental inst Mel-Roformer [model \(yaml\)](#) called just "fullness".

"this isn't called fullness.ckpt for nothing." - Musicalman

Inst. fullness: 37.66, bleedless: 35.53, SDR: 15.91 (thx Essid)

- (MVSEP) "We added 2 new algorithms for Acoustic Guitar (based on BS Roformer) and for Flute (based on SCNet XL)

1) `MVSep Acoustic Guitar (acoustic-guitar, other)` Example:

<https://mvsep.com/result/20250825095613-f0bb276157-mixture.wav>"

"excellent, it's separating acoustic from electric very well, even in fuzzy, lo-fi recordings" -

Input Output (A5)

"outperforms moises' model like crazy" - Sausum

2) `MVSep Flute (flute, other)` Example:

<https://mvsep.com/result/20250825095856-f0bb276157-mixture.wav>" - ZFTurbo

"I tried the flute model on stairway to heaven and it was so disappointing" - santilli_

- We have the first lucky person on the server who succeeded to actually use the new NVIDIA's upscaler, and their messy AF code on Windows using Docker.
The output is mono, so you need to process each channel manually.
Also, it's extremely slow, even on 4070 Super, but results are "impressive". [More](#) (don't expect step-by-step tutorial for now because the guy is "not tech support").

- Unwa released [BS-Roformer-Inst-FNO](#) model (incompatible with UVR, use [MSST](#) and read special model installation instruction below).

inst. bleedless: 42.87, fullness: 32.03, SDR: 17.60

"very small amount of noise compared to other fullness inst models, while keeping enough fullness IMO. I don't even know if phase fix is needed. Maybe it's still needed a little bit." dca "seems less full than resurrection, which I would expect given the MVSEP [metric] results. (...) I'd say it's roughly comparable to gabox inst v7"

"I replaced the MLP of the BS-Roformer mask estimator with FNO1d [Fourier Neural Operator], froze everything except the mask estimator, and trained it, which yielded good results. (...) While MLP is a universal function approximator, FNO learns mappings (operators) on function spaces."

"(The base weight is Resurrection Inst)"

Installing the model - instructions:

1. "I had many errors with torch.load and load_state_dict, but I managed to solve them. PyTorch 2.6 and later have improved security when loading checkpoints, which causes the problem. torch._C._nn.gelu must be set to exception"
> "Add the following line above torch.load (at utils/model_utils.py line 479; 531/532 in updated MSST):

```
with torch.serialization.safe_globals([torch._C._nn.gelu])
```

- unwa, or:

- 1*. Or use PyTorch older than 2.6.
2. Read the linked [model card](#). "You need to replace the entire "MaskEstimator" class in original bs_roformer.py from ZFTurbo (in models/bs_roformer folder) with the code provided by unwa [indentation error fixed].
3. And probably install this lib <https://pypi.org/project/neuraloperator/> so:
"pip install neuraloperator".
- *. Use this [models_utils.py](#) if still nothing (neoculture)

4*. Seems like [MSST](#) might have some issues with GPUs other than corresponding archs to RTX 5000, 4000, 3000, H100, H200 or maybe using ROCm, resulting in SageAttention error, forcing slower CPU separation.

In that case, ensure you have compatible CUDA/torch/torchvision/torchaudio installed:
Compatible CUDA version requirement for GTX 1660 is 10 (e.g. on GTX 1060, Torch 2.5.1+cu121 can be used), but pip doesn't find such package of Torch. To fix it:

*a) Check out index-url method described below:

```
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
```

or

```
pip install torch==2.3.0+cu118 torchvision torchaudio --extra-index-url
```

```
https://download.pytorch.org/whl/cu118
```

or

```
pip install torch==2.3.0+cu118 --extra-index-url https://download.pytorch.org/whl/cu118
```

and

```
pip install torchaudio==2.3.0+cu118 --extra-index-url https://download.pytorch.org/whl/cu118
```

Replacing cu118 with newer cu121 or even 129 seems to give proper working URL too.

Maybe replacing 2.3.0 with 2.3.1 will work too.

*a2) Alternatively, you can try to install it from [here](#) from wheels by the following command:

“`pip install SomePackage-1.0-py2.py3-none-any.whl`” - providing full path with the file name should do the trick. Just for the location with spaces, you also need “ ”.

On GTX 1660 and Turing GPUs, you might seek for e.g. `cu121/torch-2.3.1`” and those various CP wheels (there are no newer versions).

JFYI, the official PyTorch page: <https://pytorch.org/get-started/previous-versions/>

lacks links for CUDA 10 compatible versions for older GPUs other than v1.12.1 (which is pretty old, and might be a bit slower if even compatible at all), so the only way to install newer versions for CUDA 10 is the --extra-index-url trick, as executing normally “`pip install torch==2.3.0+cu118`” will end up with the version not found error.

*b) You might still have SageAttention not found error. Perform the following:

“Had to replace `cufft64_10.dll` from

`C:\Users\user\AppData\Local\Programs\Python\Python313\Lib\site-packages\torch\lib`
by the one from `C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\bin`”

It is even compatible with the newest Torch 2.8.0 (if you followed the instruction to fix the dict issue above) if you grab that apparently “fixed version of `cufft64_10.dll` from CUDA v10.0” - dca

“I guess it's possible to use it with the Colab inference custom model one, you run install cell, install the neuralop thing with “`!pip`” in a cell code (on Colab, system command needs the “`!`” before them), then edit the existing Roformer code accordingly to unwa's guidelines on his repo” - jarredou

If you want to train with FNO1d or Conformer, you might check out this [repo](#).

- Turns out Duality model is very good for pops and clicks of 45 RPM vinyl, moving them to instrumental stem (bratmix)

- Google broke installing dependencies in many Colabs.

For the inference Colab by jarredou, see [here](#) for troubleshooting (pushed the changes - not tested, might be useful for other Colabs; fixed).

In the case of AudioSR, use [huggingface](#).

- NVIDIA released their own audio upscaler, with also an ability of inpainting (so it can fill short silences between damaged segments of audio).

<https://github.com/NVIDIA/diffusion-audio-restoration>

But maybe don't try it out the upscaler just yet, as the code is currently so messy and difficult to deploy e.g. on MacOS, that it took 9 hours for two of our users, and they still didn't succeed even with help of AI chats. And to make it work on Colab, the code needs to be completely rewritten, jarredou says.

- mesk released a beta version of his metal Mel-Roformer fine-tune instrumental model called "Rifforge" focused more on bleedless.

"training is still in progress, that's why it's a beta test of the model; It should work fine for a lot of things, but it HAS quirks on some tracks + to me there's some vocal stuff still audible on some tracks, I'm mostly trying to get feedback on how I could improve it" [known issues](#).

<https://drive.proton.me/urls/5XM3PR1M7G#F3UhCU8RDGhX>

- Custom model import Colab might have currently some issues with the model above. Probably, using that old version will work (at least locally).

"My old MSST repo I'm using, but I removed all the training stuff

<https://drive.proton.me/urls/P530GFQR4W#VCAAsF0E1TPje>

pip install -r requirements.txt (u gotta have Python and PyTorch installed as well) for the script to work.

You just gotta put all the tracks you want to test on in the ***tracks*** folder then double-click on ***inference.bat*** to run the inference script

it's like if you were to type in the command in cmd, but it's simpler, and I'm lazy" - mesk

- Shared bias added during weight conversion was removed from the SW model, making it compatible with UVR and normal MSST repo code (it was just a leftover not doing anything, just zeroes). Also, delete the shared bias line from the yaml.

Also, it was possible to trim the model size to have only vocals (although it probably can be achievable quicker in the config). mask_estimators.0 is responsible for vocals (each mask estimator is responsible for other stem).

- The new violin model on MVSEP sometimes does better than the strings model for strings (dynamic64)

- Aufr33's Mel-Roformer Denoise average variant ([link](#) | [yaml](#) | [Colab](#)) can be also used for crowd removal (Gabox)

- (MVSEP) "I released new MVSep Violin (violin, other). It based on BS Roformer model with SDR: 7.29 for violin on my internal validation.

Link: https://mvsep.com/home?sep_type=65

Example: <https://mvsep.com/result/20250809120109-f0bb276157-mixture.wav>"- ZFTurbo

"I've only played around with it a little bit, but it can even separate violin quartets from cellos, so cool." - smilewasfound

"Very neat model. (...) Sometimes the model does seem to pick up more than just violins imo, but yeah for separating high strings in particular it is really cool." - Musicalman

- MVSEP now has also official YouTube channel:

<https://www.youtube.com/@MVSEP>

- Issues with https://huggingface.co/spaces/TheStinger/UVR5_UI have been fixed.

Mirror is still functional: https://huggingface.co/spaces/qtzmusic/UVR5_UI

- Unwa BS-Roformer Resurrection instrumental model added on MVSEP and on uvronline with these links for [free/premium](#) accounts.

- Gabox released experimental voc_fv6 [model | yaml](#)

"Sounds like b5e with vocal enhancer. Needs more training, some instruments are confused as vocals" - Gabox. "fv6 = fv4 but with better background vocal capture" - neoculture
bleedless: 26.61 | fullness: 24.93 | SDR: 10.64

For comparison:

SCNet XL very high fullness on MVSEP has followin metrics:

Vocals bleedless: 25.30, fullness: 23.50, SDR: 10.40

- yt-dlp and their frontendeds like cobalt.tools are currently defunct. It might affect some Colabs YT downloading features, although JDownloader 2 still works.

- The below model added on x-minus/uvronline

<https://uvronline.app/ai?discordtest> - free accounts

<https://uvronline.app/ai?hp&test> - premium accounts

- Unwa released a new BS-Roformer Resurrection instrumental [model | yaml | Colab](#)

SDR: 17.25, bleedless: 40.14, fullness: 34.93

Compatible with UVR (model type v1). "Fast model to inference (204 MB only)".

"One of my favorite fullness inst models ATM. Sounds like v1e to me, but cleaner. Especially with guitar/piano where v1e tended to add more phase distortion, I guess that's what you'd call it lol. This model preserves their purity better IMO" - Musicalman

"the way it sounds, is indeed the best fullness model, it's like between v1e and v1e+, so not so noisy and full enough, though it creates problems with instruments gone in the instrumental sadly, but apparently it seems Roformer inst models will always have problems with instruments it seems, seems like a rule. (...) Instrument preservation (...) is between v1e and v1e+ (...) Fixes crossbleeding of vocals in instrumental in a lot of songs, compared to previous models (...) No robotic voice bug at silent instrumental moments" - dca100fb8

"Some songs leaves vocal residue. It is heard little but felt" - Fabio

"Almost loses some sounds that v1e+ picks up just fine" - neoculture

Mushes some synths a bit in e.g. trap/drill tune compared to inst Mel-Roformers like INSTV7/Becruily/FVX/inst3, but the residues/vocal shells are a bit quieter, although the clarity is also decreased a bit. Kind of a trade.

So far, none models work for phase fixer/swapper besides 1296/1297 by viperx and unwa BS Large V1 to alleviate the remaining noise. ~ dca. SW model not tested.

Less crossbleeding than paid Dango 11.

- Gabox released a bunch of new models:

a) Gabox Inst_ExperimentalV1 model | [yaml](#)

b) Gabox Kar v2 Mel-Roformer | [model](#) | [yaml](#)

SDR is very similar with the v1 Gabox model: 9.7699 vs 9.7661.

Lead:

bleedless: 27.58 vs 28.18, fullness: 15.24 vs 14.79

Back-instrum:

bleedless: 50.67 vs 50.74, fullness: 32.46 vs 32.84

(but you'll most likely get better results with Gabox denoise/debleed Mel-Roformer model instead ~Gabox, but it can't remove vocal residues

[model](#) | [yaml](#) | [Colab](#))

c) Gabox Lead Vocal De-Reverb Mel-Roformer | [DL](#) | [config](#) | [Colab](#)

“just use it on the mixture” - Gabox, “sounds great” - Rage313

Sometimes removes back vocals, especially if they're panned to the sides.

“(...) also a vocal/inst separator. Dry vocals go in vocal stem, everything else goes to reverb.

Don't think anvuew's models do that.

I might still preprocess with vocal isolation before dereverb. But only really worth it if you're after high fullness vocals.” - Musicalman

- Issues with https://huggingface.co/spaces/TheStinger/UVR5_UI occur.

“We have a problem with Zero GPU atm, waiting for a fix from HF staff isn't related to the code or last commit” - Not Eddy

Meanwhile, you can use: https://huggingface.co/spaces/qtzmusic/UVR5_UI

- (MVSEP) Now 32-bit float for WAV will be used only if gain level falls outside 1.0 range to prevent clipping, otherwise 16 bit PCM will be used, when it won't occur. If you really need it anyway, 32-bit float output for all files unconditionally is available for paid users.

If you have troubles with nulling due to the new changes in free version, consider decreasing volume of your mixtures by e.g. 3-5dB, and you won't be affected, although it might slightly affect separation results.

Also, FLAC now uses 16-bit instead of 24-bit.

- (MVSEP) “Sometimes we have complaints on speed from different parts of the world. The best way is to use VPN to solve them.” - ZFTurbo

- Gabox' voc_Fv5, Inst_GaboxFv7z, Unwa's voc Resurrection, voc_gabox2 and the new jarredou's drumsep 5 stem added to the [inference Colab](#)
 (Resurrection [now with the config] and Fv7z fixed). Also, previously released anvew's mono dereverb added.
 - If you feel overwhelmed by this GDoc's list, isling released their own, shorter version with recommended models for audio separation - [click](#).
 - Also, [here](#) you'll find an excerpt of the current document with only models and their links, if you find it hard to navigate through the whole document (edit. 24.07.25)
 - (MVSEP) BS-Roformer 2025.06 described previously below received two updates (11.81 -> 11.86 and 11.86>11.89) and it has been changed to:
 BS-Roformer 2025.07. [Full metrics](#).
 "All Ensembles and models where this model is involved improved a little bit too." - ZFTurbo MVSEP Multichannel BS feature started using 11.81 model at some point, now sure if now uses 11.89.
 If you tried achieving results any similar to BS-Roformer 2025.07, you could potentially try out [splifft](#) or its [Colab](#). If you fail to use Splifft check the [model](#) before conversion on MSST repo ([more](#))
 - (MVSEP) Gabox INSTV7 instrumental model added
 - (MVSEP) MelBand Karaoke (lead/back vocals) Gabox model added (SDR: 9.67)
 - Fused model of Gabox and Aufr33/viperx weights 0.5 + 0.5 added (SDR: 9.85)
 It gives maybe only slightly worse results than normal ensembling, but with separation time of just one model "it doesn't have the same quality and definition as Gabox Karaoke, fused doesn't separate well." - Billie O'Connell.
 You can perform fusion of models using [ZFTurbo script](#)^(src) or by [Sucial script](#) (they're similar if not the same). "I think the models need to have at least the same dim and depth but I'm not sure about that" - mesk.
 Despite the higher SDR, the fusion model seems to confuse lead/back vocals more.
 - The same goes to public Karaoke fusion models released by Gonzaluigi [here](#)
 - Gabox released Mel-Roformer [voc_gabox2](#) vocal model | [yaml](#) | [Colab](#)
 Vocal bleedless: 33.13, fullness: 18.98, SDR: 10.98
 - Unwa released a BS-Roformer vocal model called "[Resurrection](#)" | [yaml](#) which shares some similarities with the SW model (might be a retrain). The default chunk_size is pretty big, so if you run out of memory, decrease it to e.g. 523776.
 Vocal bleedless: 39.99, fullness: 15.14, SDR: 11.34.
- "Omg, this model is doing a really good job at capturing backing vocals (...)

Honestly, it sounds a bit muddy, and there's some instrumental bleeding into the vocal stems" neoculture

Not so good for speech denoising unlike some other models (Musicalman).

- mesk's training guide updated and [link](#) changed

- (MVSEP) New All-in and 5 stem ensembles have been added for paid users

- AudioSR WebUI [Colab](#) by Sir Joseph got fixed

- MVSep Ensemble 11.93 (vocals, instrum) (2025.06.28) added.

Eventually surpassed sami-bytedance-v.1.1 on the multisong dataset SDR-wise.

Instrumental bleedless: 47.65, fullness: 28.76, SDR: 18.24

Vocal bleedless: 36.30, fullness: 17.73, SDR: 11.93

- corrected typos in the metrics (thx yakomotoo)

- (MVSEP) MultiChannel now uses 11.81 BS Roformer model.

- (MVSEP) New BS Roformer model is now available on site - it's called 2025.06 (don't confuse it with SW).

Vocals bleedless: 48.59, fullness: 27.85, SDR: 11.82

Instrumental bleedless: 37.83, fullness: 17.30, SDR: 18.12

"It has +0.5 SDR to the previous best [24.08] model. We reached ByteDance's best model quality [only 0.1 SDR difference]. It is also TOP1 on the Synth dataset. It's balanced between both [instrumental and vocals]. I used metal dataset during training as well"

Compared to previous models, picks up backing vocals and vocal chops greatly where 6X struggles, and fixes crossbleeding and reverbs where in some songs previous models struggled before. Sometimes you might still get better results with Beta 6X or voc_fv4 (depending on a song). "Very similar to SCNet very high fullness without the crazy noise" - dynamic64, "handles speech very well. Most models get confused by stuff like birds chirping (they put it in the vocal stem), but this model keeps them out of the vocal stem way more than most. I love it!"

"not a fan of the inst result. I feel like unwa and gabox sound better despite being less accurate" - dynamic64. Might be better than Fv7n "I think gabox tends to sound better but the new BS-Roformer is more accurate" dynamic64, "instrumentals are muddy" - santilli_, "I think the Gabox [fv7n] model sounded more crispier than BS" - REYYY. "[voc_]fv4 sounds better" - neoculture, "instrumentals sound very good" - GameAgainPL.

"it did things i never thought it could before" "this model is insane wtf (...) never seen a model accurately do the ayahuasca experience before" - mesk.

"the first model to not produce vocal bleed in instrumental for "Supersonic" by Jamiroquai (not even Dango does it). It is also the case with "Samsam (Chanson du générique)" and "Porcelain" by Moby." and "In the Air Tonight" by Phil Collins, also "removes very most of Daft Punk vocoder vocals" - dca. "my new favorite for vocals. It sounds fantastic" -

dynamics64. “for the first time ever it managed to remove the reverb from one specific song. it is not perfect, but still much better than previous attempts” - santilli_
“It even seems to handle speech very well. Most models get confused by stuff like birds churping (they put it in the vocal stem), but this model keeps them out of the vocal stem way more than most. I love it!”. “sometimes 6x is better sometimes bs is better” - isling “for me it's picked up a lot that 6x hadn't for backing vocals

- Using this [repo](#), you can convert Mel-Roformers, HTDemucs and Apollo models to OpenVINO (so to onnx)
- Lew, if you read it, some guy wants to add your Apollo uni model into a plugin for OpenVINO and Intel's HF, but the model lacks an open source licence. If you could re-release it with the proper licence, it would be appreciated. [More](#)
- (x-minus/uvronline) “I added two new models to remove vocals and hid a few old ones.

So there are now only three main models in the menu for different purposes:

Mel-RoFormer by Gabox Fv7z - best bleedless, good fullness, almost noiseless

Mel-RoFormer by unwa v1e+ - best fullness, average bleedless

Mel-RoFormer unwa big beta6x - best vocals

Older models are still available at the link:

<https://uvronline.app/ai?hp&test> (premium)

<https://uvronline.app/ai?test> (free)” - Aufr33

“Oh! Lead vocal panning has been added for Mel Kar Old! (...)

Along with MDX Kar old and UVR Kar old to the test page!!” - dca

- Gabox released a new experimental [Karaoke model](#). It's one stem target so keep extract_instrumental enabled for the rest stem.
“really hard to tell the difference between this and becruily's karaoke model” minus the latter has more target stems.
- jarredou released his new MDX23C drumsep 5 stem model, which is public for everyone to [download](#). All SDR metrics are better than the previous model (“on kick/snare/toms it's around +2 SDR better than previous version”):
SDR: kick: 16.66, snare: 11.54, toms 12.34, hihat: 4.04, cymbals: 6.36 ([all metrics](#)).
Metric fullness for snare: 25.0361, bleedless for hh: 12.3470, log_wmse for snare: 13.8959
“Quite cleaner than the previous one”, “it's more on the fullness side than bleedless”,
From all the metrics, only bleedless for snare is worse than in the previous model:
26.8420 vs 30.4149 and indeed “snare has a bit of bleed sometimes” - isling, “as well as cymbals bleed in hi hat track, but the stems sound clean” - dca.
“a lot noisier than other drumpsep models, but that's not necessarily a bad thing.”

“Surprisingly, it's the 2nd best model for hi hat and 2nd best model for cymbals on mvsep leaderboard. It's a bit biased because ZF's top mel model is 4 stem only.”

For comparison, [metrics](#) of the old 6 stem jarredou/Aufr33 MDX23C model (which has cymbals divided into ride and crash which are not evaluated):

SDR: kick: 14.55, snare: 9.79, toms: 10.64, hihat: 3.20, cymbals: 6.08

Metric fullness for snare: 25.0361, bleedless for hh: 10.2765, log_wmse for snare: 12.4258

The model was trained with a lightweight config to train on a subpar T4 GPU on free Colabs and 10 accounts (“CRAAZY fast” for inferencing). The metrics do not surpass exclusive drumsep Mel-Roformer and SCNet models on MVSEP, but at least you can use this one locally.

“Most of the issues with my model are already known issues with mdx23c arch, it's bleedly and has band splitting artifacts. Like I said a few days ago, if I would have to redo it now, it would have probably gone with SCNet Masked. It's using 4x times lower n_fft resolution than InstVocHQ while using 2 times longer chunk_size (and with MDX23C, whatever number of stems, it's the same inference speed). A bit like the fruit's model is doing”.

Trained on 511 tracks, MVSEP models were trained on almost the same dataset.

Maybe if we separate just snare with the old MDX23C model from an already separated drums stem, and mix/invert to get the rest, then pass it through the new model, the bleed would be gone.

Remember that you need already separated [drums](#) in one track to use this model effectively. About used dataset: “It was around 2/3 acoustic drums and 1/3 electro drums dataset at start of training, I've added more electro drums at end of training to balance it a bit more.” - jarredou

- septicoco released [macvsep](#) which is “macOS client for the Mvsep music separation API”

- Added Clear Voice in [speech separation](#)

- Gabox released [Inst_GaboxFv7z](#) Mel Roformer | [yaml](#)

Inst. fullness: 29.38, bleedless: 44.95

“Focusing on the less amount of noise keeping fullness”

“The results were similar to INSTV7 but with less noise” - neoculture

Metrically better bleedless than Unwa v2 (although it's even more muddy), for comparison:

Fullness: 31.85, bleedless: 41.73

- (MVSEP) “I added a new SCNet vocal model. It's called SCNet XL IHF. It has a better SDR than previous versions. Very close to Roformers now”.

Vocal bleedless is the best among all SCNet variants on MVSEP. [Metrics](#).

IHF stands for “Improved high frequencies”.

Vocal bleedless 28.31, fullness 17.98

“certainly sounds better than classic SCNet XL (...) less crossbleeding of vocals in instrumental so far, and handle complex vocals better (...) problems with instruments, compared to high fullness one. XL high fullness remain the one without too many

instruments cut”, but some difficult songs used with previous models can yield better results

- dca

- Great news! MVSEP now allows sorting scores on the [Multisong Leaderboard](#) by SDR, fullness, bleedless, aura_stft, aura_mrstft, log_wmse, l1_freq, si_sdr.

Be aware that Gabox (and probably sometimes becruily) used to give funny names to their evaluations, so finding proper model names on the leaderboard is sometimes impossible. But I've tracked down all possible models with their metrics and proper names in the [instrumentals](#) and [vocal](#) models section, so no worries.

Also, metrics beside SDR are not available for old evaluations where they weren't listed in the model details yet. You can find more info about bleedless/fullness metrics [here](#).

Log WMSE metric is good “at least for drums or anything rich in low frequency content” - jarredou

- Our server members send their warm regards to A5 whose account disappeared for the ~5th time :) And later reappeared weirdly mutated.

- “Dango launched their new instrumental model

<https://tuanziai.com/en-US/blog/684841907c8c85686c1b3da6>” It's version 11.

“there is no opportunity to try at least 3 complete tracks for free.”

Some crossbleeding issues from v10 are still present, plus some songs are even getting worse results than in v10. You might want to use v1e (with phase fix) + Becruily vocal model (Max Spec) instead, although some people might still like Dango anyway.

“Some tracks are fuller than Gabox v8”. Conservative mode is less full than V1e.

“They have a tool called "edit & improve" [or "Advanced Repair tool"] that lets you use 'Conservative mode' for some of more complex parts of a song and 'Smart mode' for other parts. I find that way more convenient than processing the entire track in 'Conservative' mode.”

They plan to release a karaoke model in two months.

- Gabox released a “[small](#)” version of Mel instrumental model for faster inference | [yaml](#)

Be aware that it can have some audible faint constant residues.

- ZFTurbo: “I added BS Roformer SW to "MVSep Piano", "MVSep Guitar", "MVSep Bass", "MVSep Drums" algorithms. For Bass and Drums available new Ensembles.”

“drum SDR jumped by .6 on the ensemble! Atho fullness took a hit” - heuhew

“Same with bass, sdr leaped but fullness shot down 4 points” - dynamic64

- BS-Roformer SW 6 stem model replaced the old one.

lirc, the model didn't change, just the inference code. SW stands for “shared weights”

“I got better drums/bass separation with that model than with any others when input is some live/rehearsal recordings with shitty sound”

Also, there's better SDR and fullness for instrumentals when you invert vocals against mixture instead of mixing down drums/bass/other stems.

- undef13 released these “bs-roformer weights stored in fp16 precision, half the size of frazer's initial version. quality is the exact same as the fp32 version”.
- First vocal retrain was published shortly after a day “May not perform very well” (at least for now)

If you were to fine-tune it “this model generalizes like crazy (...) hasn't failed yet to confuse instruments and just chews through whatever you put through it (i ignore the overall mudiness)” you can retrain it to just being inst/voc model “I'm currently training it to my 2 stem (...) dataset (...) I was pleasantly surprised” iirc on even laptop RTX 3070...
- Added on MVSEP as BS-Roformer 6 stem (no clicking issues)

- The new Logic Pro model has been reversed/cracked and shared as a standalone model for inference. Full metrics for all stems added later below. It has the best SDR on multisong dataset for all stems besides vocals (but still not bad).

It uses a BS-Reformer arch. .MIL (CoreML) model file was converted to .PT.

“The only change they made was a global parameter for bias which I've never seen before so I guess it's apple secret sauce”. No quantization was used “they had a shared bias across QKV and the out_proj”.

“It is wonderful to achieve such results with dim 256. It seems that what was still needed was depth.”

A bit scared to share it, but seek and ye shall find.

Usage:

```
python inference.py --audio_path="./sample.flac"
```

For: ModuleNotFoundError: No module named 'hyper_connections'

Run: pip install hyper_connections

“looks like chunks aren't overlapping? Getting clicks in output.”

“A very small edit, line 13:

```
parser.add_argument('--chunk_size', type=int, default=588800)
```

- this produces 99% identical results with the DAW.

Previous 117760 chunk size was adding clicks and was lower quality in general.”

Still, the code doesn't use overlap, and it will result in click, just less than before.

Also, you can run out of memory with 588800 with 5GB VRAM free.

882000 was tested to have the biggest SDR in that model (not lower or higher).

On a CPU without an Nvidia GPU it will probably be long.

The inference script and model probably still needs the validation to ensure the metrics are the same with the [validation](#) made from DAW lately, but it's rather the same (at least other inference code got 0.03 SDR difference or same results based on the same converted weights).

To use it with ZFTurbo MSST repo:

“You need to replace bs_roformer.py in the repo with file from the archive (...) and change line 8 to:

from models.bs_roformer.attend import Attend" and then use separately shared config for the MSST repo and the model. Using MSST repo for inferencing fixes the clicking issue. For "unrecognized arguments" issue, "you must put your path inside quotation marks or apostrophes".

"If you're using the script GUI, be aware that the browser popup window when choosing checkpoint has some predefined extension and .pt is not part of it"

Since then, weight compatible with UVR with deleted shared bias was shared (there were actually only zeroes). Also, with mask estimator method, just one stem file can be extracted out of the full weight. Vocals only were shared, but config for UVR will rather require some tweaks.

- Bebruily guitar model added to inference [Colab](#), bleed suppressor by unwa/97chris model fixed, denoise-debleed by Gabox added, Revive 3e fixed, Revive 2 added

- Gabox released [instv7plus](#) bleedless model (experimental)
fullness: 29.83, bleedless: 39.36, SDR 16.51

- And [Inst_FV8b](#)

fullness: 35.05, bleedless: 36.90, SDR 16.59

"Very clean" although muddier than V1E+.

- wesleyr36/Dry Paint Dealer Undr HTDemucs Phantom Center model was added to the [inference Colab](#)

- (Unwa) "After a long time, I'm uploading a vocal model specialized in fullness.

Revive 3e is the opposite of version 2 — it pushes fullness to the extreme.

Also, the training dataset was provided by Aufr33. Many thanks for that."

[bs_roformer_revive3e](#) | [config](#) | [Colab](#) (should be fixed now)

Voc. SDR: 10.98, fullness: 21.43, bleedless: 30.51

- Logic Pro updated their stem separation feature, which now incorporates guitar
Overall, it's "surprisingly good" - dynamic64. And a piano separator was also added to it.

[More](#)

"Guitar & Piano separation seems to be really on point. So far it separated super well, also didn't confuse organs for guitars and certain piano sounds as well." - Tobias51

"guitar model sounds better than demucs, mvsep, and moises" - Sausum

"it's not a fullness emphasis or anything, but it's shockingly good at understanding different types of instruments and keeping them consistent sounding" - becruily

You don't need to process L and R for bleeding across channels like in other models, there isn't any in this one - A5

Full [evaluation](#) on multisong dataset (besides instrumental):

SDR piano 7.79, bleedless 31.96, fullness 14.42

SDR other 19.90, bleedless 58.68, fullness 49.85

SDR guitar 9.00, bleedless 31.54, fullness 15.95

SDR other 15.94, bleedless 49.36, fullness 31.57

SDR drums 14.05 (although lower fullness than MVSep SCNet XL drums 14.26 vs 21.21),

SDR bass 14.57 (-||-), other 8.66, vocals 11.27 (only that is not SOTA)

MVSep Piano Ensemble (SCNet + Mel) has only other fullness higher: 56.96 ([click](#))

- Since 23.05.25 jarredou (Discord: rigo2) and dca100fb8 (Discord) also have writing privileges to this document. You can find it mirrored to this date [here](#) in docx, pdf and html.

- Bebruily released Melband guitar [model](#) | [Colab](#)

“Not SOTA, but much more efficient and comparable to existing guitar models, and for some songs it might work better because it picks up more guitars (though it can also pick some other instruments).

For better results you might try first removing vocals.”

- (MVSEP) “We added a new GUI example to work with the MVSep API. Now it allows to use multiple files and multiple algorithms at once.

It exists as standalone .exe file, so it doesn't require python installation

Repository: <https://github.com/ZFTurbo/MVSep-API-Examples>

Exe for Windows:

https://github.com/ZFTurbo/MVSep-API-Examples/raw/refs/heads/main/python_example5_gui/mvsep_client_gui_win.exe - ZFTurbo

TL;DR “You can process a song with multiple models, and process multiple songs”

- Scial released v1/2 de-breath VR models:

<https://huggingface.co/Scial/De-Breathe-Models/tree/main>

Alternatively, for this purpose you can also try out free/abandonware:

<https://archive.org/details/accusonus-era-bundle-v-6.2.00>

- (x-minus/uvronline) Aufr33 added new Lead and Backing vocal separator.

~~It uses big beta 5e model as preprocessor for bebruily Mel Karaoke model~~ “In fact, the big beta 5e model is run after bebruily Mel Karaoke” Aufr33 (so you don't need the additional step to use this separator), plus it also allows controlling option for lead vocal panning like for BVE v2 (it's to “to “tell” the AI where the main vocals are located (how they are mixed).”).

Bebruily's model “doesn't even need Lead vocal panning a lot of the time, [the] ability to recognize what is LV and what is BV [is] impressive” - dca).

The difference from using single bebruily Kar model (without preprocessor) is that, here, “you get the third track, backing vocals.”.

“The new separator is available in the free version, however, due to its resource intensity, only the first minute of the song will be processed.” if you don't have premium.

Bebruily:

“Probably too resource-intensive, but you could try adding demudders to each step

1) karaoke model + demudding

2) separate vocals of bgv + demudidng
But not sure how much noise this will bring
(Or even a 50:50 ensemble with BVE OG")

- Unwa released [Revive 2](#) variant of his BS-Roformer fine-tune of viperx 1297 model
Voc. bleedless: 40.07, fullness: 15.13, SDR: 10.97
“has a Bleedless score that surpasses the FT2 Bleedless” and fullness lower by 0.64.
“can keep the string well” better than viperx 1297 (...) in my country they have some song with Ethnic instruments. Only 1297 and Revive2 can keep them in Instrumental while other model notice them as Vocal” ~daylight
“it does capture more than viperx's” - mesk
It's depth 12 and dim 512, so the inference is much slower than with some newer Mel-Roformers like voc_fv4 (even two times), with the exception of Mel 1143 which is as slow as BS 1297 (thx dca, neoculture).

- BS-Roformer Revive unwa's vocal [model](#) (viperx 1297 model fine-tuned) was released.
Voc. bleedless: 38.80, fullness: 15.48, SDR: 11.03
“Less instrument bleed in vocal track compared to BS 1296/1297” but it still has many [issues](#), “has fewer problems with instruments bleeding it seems compared to Mel. (...) 1297 had very few instrument bleeding in vocal, and that Revive model is even better at this (...). Works great as a phase fixer reference to remove Mel Roformer inst models noise” it doesn't seem to remove instruments like FT3 Preview for phase fixing (thx dca100fb8)
Added to [phase fixer Colab](#).

- [Inst GaboxFv8](#) model | [yaml](#) | [Colab](#) checkpoint has been updated, metrics could have changed, but most of the model qualities might remain similar

- (MVSEP) “I added new Drumsep MelBand Roformer (4 stems) model on MVSep (old one was removed). It gives the best metrics with big gap for kick, snare and cymbals.” - ZFTurbo ([metrics](#); only toms are worse SDR-wise vs previous SCNet Drumsep models)

- Gabox released [voc fv5](#) vocal model | [yaml](#)
voc bleedless: 29.50, fullness: 20.67, SDR: 10.56
“fv5 sounds a bit fuller than fv4, but the vocal chops end up in the vocal stem. In my opinion, fv4 is better for removing vocal chops from the vocal stem” - neoculture. [Examples](#)
“v5 is slightly fuller, v4 is less full but also slightly more careful about what it considers as vocals. I think b5e is the fullest overall, but it's a bit much sometimes. Pretty sure the gabox models are a little more accurate with vocal/instrument detection.” Musicalman
Passes the Gregory Brothers - Dudes a Beast test (before - trumpets in vocal stem at 0:51; unwa's beta4 and inst v1e tested) - maxi74x1

- *Some of our less active users have been accidentally kicked out of our Discord server during some administrative tasks. You're free to rejoin using [this](#) invite link (unless you were banned before in some other unrelated event).*

- Dry Paint Dealer Undr (a.k.a. wesley36) released new Phantom Centre Models:
HTDemucs Similarity/Phantom Centre Extraction model:
https://drive.google.com/drive/folders/10PRuNxAc_VOcdZLHxawAfEdPCO6bYIi3?usp=sharing (it tends to be more “correct” in center extraction than the last MDX23C model)
The Demucs model won’t work with UVR giving bag_num error even with the yaml prepared in the same way as for Imagoy Drumsep and after renaming ckpt to th (it’s probably because it needs ZFTurbo inference code).
SCNet Similarity/Phantom Centre Extraction model:
<https://drive.google.com/drive/folders/1CM0uKdf60vhYyYOCg2G1Ft4aAiK1sLwZ?usp=sharing>

And also, difference/Side Extraction model based on SCNet arch was released:
<https://drive.google.com/drive/folders/1ZSUw6ZuhJusv7HE5eMa-MORKA0XbSEht?usp=sharing>

- Aufr33 released his UVR Backing Vocals Extractor v2 [model](#), previously available only on x-minus/uvronline (VR arch).

“Note that this model should be used with a rebalanced mix.

The recommended music level is no more than 25% or -12 dB.

If you use this model in your project, please credit me.”

Should work in UVR. Just place the model file in Ultimate Vocal

Remover\models\VR_Models and [config](#) file in lib_v5\vr_network\modelparams. Then pick “4band_v4_ms_fullband.json” when asked to recognize the model (it has the same checksum as in lib_v5\vr_network\modelparams folder if it’s there already). Also, I think it’s not VR 5.1 model. And it was used with vocal model as preprocessor.

More about its usage in [Karaoke](#) section (scroll down a bit).

- squid.wtf doesn’t work anymore “it just downloads 30 seconds of a song, just a random 30 second snippet” lucida works.

- [USS-Bytedance](#) Colab has been fixed (Python “No such file or directory” fix) - thx epiphery.
https://colab.research.google.com/drive/1rfI0YJt7cwdxT_pQlgobJNuX3fANyYmx?usp=sharing

- (MVSEP) “I added a new MVSep Saxophone (saxophone, other) model. It has 3 versions:
SCNet XL (SDR saxophone: 6.15, other: 18.87)

MelBand Roformer (SDR saxophone: 6.97, other 19.70)

Ensemble Mel + SCNet (SDR saxophone: 7.13, other 19.77)” ZFTurbo

“SCNet XL take[s] wurlitzer as sax tho. Mel Rofo one (...) didn’t” - dca

- (x-minus) Server code updated [might fix the issue with bleeding at first seconds in e.g. Mel Decrowd; edit. it didn’t]

Added Lead vocal panning setting for Mel-RoFormer Kar by becruly model.

[It's] "to "tell" the AI where the main vocals are located (how they are mixed).

Added Demudder for the Mel-RoFormer Kar by becruily model." - Aufr33

"doesn't even need Lead vocal panning a lot of the time, [the] ability to recognize what is LV and what is BV [is] impressive" - dca

- Anjok released a new UVR Roformer patch #15 fixing CUDA for RTX 5000 Series GPUs and Windows users (it's based on CUDA 12.6 and newer PyTorch). It might not be backward compatible with older GPUs, so be aware ([src](#)).

[Download](#)

- (MVSEP) "I added becruily Karaoke model. It's available as option in MelBand Karaoke (lead/back vocals) algorithm." ZFTurbo

- (MVSEP) Since at least February there's a normalization for all input unless WAV is chosen as output format.

Sometimes it can be "annoying when you have to combine the outputs later".

"No, if you turn off normalization, FLAC will cut all above 1.0

And if it was normalized, it means you had these values."

FLAC doesn't support 32-bit float, it's 32 int, so normalization is still needed."

So if your stems don't invert correctly, just use WAV output format - it's 32-bit float.

- Audioshake now have strings model

- [Fast Separation](#) Colab by Sir Joseph has been updated with the following models:
MelBand Roformers: FT 3 by unwa, Karaoke by becruily, FVX by Gabox, INSTV8N by Gabox, INSTV8 by Gabox, INSTV7N by Gabox, Instrumental Bleedless V3 by Gabox, Inst V1 (E) Plus by Unwa, Inst V1 Plus by Unwa

- (stephanie/UVR) "Those of you on Linux running the current *roformer_add+directml* branch that cant get becruily's karaoke model working due to the same error:

it seems editing line 790 in separate.py setting the keyword argument strict to False when calling load_state_dict seems to make the karaoke model load and infer properly, so i think it will work

`model.load_state_dict(checkpoint, strict=False)`

I don't know if this is a robust workaround, but I haven't observed anything behaving differently than it should yet, so if you want to give it a shot I think it will work

TL;DR change line 790 in separate.py to the codeblock and then run again and karaoke model should work"

- Aname's Mel-Roformer 4 stems Large added to inference [Colab](#)

- Apollo Lew Uni model can be also used as denoiser.
It tends to smooth out some noise in higher frequencies, making the spectrum more even there, smoothing out the sound in general ([example](#)).
More about the model and its usage - [click](#).

- Bebruily's Mel-Roformer Karaoke model added on x-minus/uvronline under "Keep backing vocals" option and in the inference [Colab](#)

Most likely, you'll have ""norm"" AttributeError when trying out that model in UVR. Read [here](#) for troubleshooting. Use melband-roformer model type, not v2.
Make sure you use the latest UVR Roformer [patch](#) - older patches like #2 will show RuntimeError about layers.

- (bebruily) "I'm releasing my first karaoke [model](#).
It's a dual model trained for both vocals and instrumental. It sounds fuller + understands better what is lead and background vocal, and to me, it is better than any other karaoke model."

"Compared to Aufr33's Melband model, it can achieve e.g. cleaner pronunciation in some songs ([examples](#)) - neoculture "It is the best available, better than Mel Kar, UVR BVE v2, lalal.ai, Dango..." - dca "This sounds amazing" - Rege 313 "It performs very well with male/female duets, nice work" - Gabox

"Important note: This is not a duet or male/female model. If 2 singers are singing simultaneously + background vocals, it will count both singers as lead vocals. The model strictly keeps only actual background vocals. The same goes for "adlibs" such as high notes or other overlapping lead vocals.
The model is not foolproof. Some songs might not sound that much improved compared to others. It's very hard to find a dataset for this kind of task.

Tip: For even better results, first extract the vocals with a fullness model (like mine) and combine the results with a fullness instrumental model." bebruily

The model outputs 2 stems like duality models, so you might end up with three outputs if you check the option to invert stem - don't use it, it will rather have worse quality than what the model outputs.

- (MVSEP) "I added 2 more models for DrumSep based on MelBand Roformer architecture."
a) 4 stems (kick, snare, toms, cymbals) - average [SDR](#) of hihat ride, crash is 11,52 (but in one stem) and so far it's the best SDR out of all models (even vs the previous ensemble consisting of three MDX23C and SCNet models).
b) 6 stems (kick, snare, toms, hihat, ride, crash) - average SDR of hihat ride, crash is 8.18 (but from separated stems), while

The snare in a) has the best SDR out of all available models.

Kick and toms are still the best SDR-wise in the previous 3x MDX23C and SCNet ensemble (new ensemble with these new Mel-Roformers so far)

- The new models “are very great for ride/crash/hh. And overall they have the best metrics almost for all stems.” - ZFTurbo

- Aname released two 4 stems Mel-Roformer models:

<https://huggingface.co/Aname-Tommy/melbandroformer4stems/tree/main>

a) Large (4GB) SDR drums: 9.72, bass: 9.40, other: 5.11, vocals 8.65 (multisong dataset)

b) XL (7GB) SDR drums: 9.83, bass: 9.37, other: 5.31, vocals 8.57 (multisong dataset)

The latter doesn't work in the custom model import Colab with at least the default chunk_size, and works slow on e.g. 3060 (?12GB). Both models were trained with chunks set to 15 seconds (chunk_size = 661500).

“I tried a song on 4070 Super it took like 6 mins on XL 4 stems compared to 30 seconds on Large 4 stems” On 3060 XL is very slow.

Despite lower AVG SDR on musdb18 dataset vs demucs_ft (8.54 vs 9), it seems to outperform that model (SDR is only better in other stem), public SCNet, SCNetXL, BS-Roformer have better [metrics](#) (still musdb18 dataset, not multisong on MVSEP)

“Drums are sounding really good in particular, tested a couple songs with the large model after using unwa's v1e+ for instrumental” “drums are absolutely the standout”

“Large works in like 99% use case” “Large split sounds amazing so far tho”

XL “result would take so much longer, but the large results sounded better imo” 5B

“The Colab is forcing a different value than the one from the config. You can try to edit the inference cell code and add 661500 as possible value and see if it goes better.

The Colab only changes chunk_size (value from GUI), batch_size (forcing =1) and overlap (value from GUI), it doesn't touch other settings from config.” - jarredou

“It may change audio setting, chunk_size=485100, n_fft=2048 will work, but it will go lower SDR maybe” while the lowest reasonable value will be rather 112455 (2,5 s).

Large model uses 7GB VRAM on Nvidia GPU in UVR with default config settings.

- Sir Joseph released [SESA Fast Separation](#) Colab based on UVR. It's faster than the regular [SESA](#) Colab (which now has “added Apollo to Auto Ensemble and fixed a few technical glitches. It's running smoother now!”)

More changes in the fast Colab:

V1e+ and Gabox inst fv8 are missing because the model list cannot be updated in the Fast Colab yet.

“auto-ensemble feature is included here too.

Background noise suppression is a bit more polished.

You can specify unwanted stems to filter out.”

- (x-minus/uvronline) “1. A new Mel-RoFormer by unwa v1e+ model has been added. It removes vocals very gently while preserving instruments. It is recommended to use it with correct_phase post-processing.

2. Mel-RoFormer by Kim & unwa ft3 and some other models are hidden. As before, you can find them here: <https://uvronline.app/ai?hp&test>" - Aufr33

"The only problem is the phase correction, it still uses FT2 as a reference [for phase fixer], and FT2 cuts instruments still, so I'm waiting for FT3 release by unwa so it can be added as phase fixer reference and preserve instruments well" dca

"Results are still better with phase fixer though, right"

Make sure you're "clicking on "Ensemble"? It should "reveal" that option" since the last website layout changes.

"phase fixer [on the site] swaps the v1e+ vocals with the ft2 vocals"

lirc phase fixer feature requires premium.

- SESA Colab by Sir Joseph is back! The Colab link has changed - [click](#)

Apollo Integration: Added Apollo audio enhancement feature. Supports Normal and Mid/Side methods.

UI Updates: Added new Apollo settings components under the Settings tab.

Bug Fixes:

Fixed Apollo output not showing in the terminal.

Corrected "Phase Remix" and "Overlap Info" display in the UI.

Translation Updates: Added new translation keys for Apollo, removed unused keys.

Colab Support: Added 10 new languages: EN_US (English), TR_TR (Turkish), AR_SA (Arabic), RU_RU (Russian), ES_ES (Spanish), DE_DE (German), ZN_CN (Chinese), HI_IN (Hindi), JA_JP (Japanese), IT_IT (Italian).

and new models added

Note: Enhanced UI and processing stability.

- Gabox released [Inst_GaboxFv8](#) model ([yaml](#)) [weight has been replaced by v2]

Inst. bleedless: 38.06, fullness: 35.57, SDR: 16.51 [outdated]

Might have some "ugly vocal residues" at times (Phil Collins - In The Air Tonight) - 00:46, 02:56 - dca.

VS v1e "it seems to pick up some instruments better" Gabox

"a bit cleaner-sounding and has less filtering/watery artifacts.

Both models are prone to very strange vocal leakage ["especially in the chorus."].

And because Fv8 can be so clean at times, the leakage can be fairly obvious. For now, my vote is for Fv8, but I'll still probably be switching back and forth a lot" - Musicalman

"sometimes v1e+ have vocal residues which sound like you were speaking through a fan/low quality mp3" - dca

- Added Mesk Metal Model Preview, Unwa v1+ Preview, and Unwa v1e+ Mel instrumental models and Beta6X and FT3 Preview by Unwa vocal models, and Bandit v2 multilingual model to inference [Colab](#)

- Unwa released a new V1e+ Mel-Roformer instrumental [model](#) | [yaml](#) | [Colab](#)

Inst bleedless: 36.53, fullness: 37.89, SDR: 16.65

Less noise than v1e (esp. in the lower frequencies), but it's also less full - "somewhere between v1 and v1e." It has fewer problems with quiet vocals in instrumentals than the V1+, "issues with harmonica, saxophone, elec guitar and synth seem to have been fixed.

Theremin and kazoo are still problematic [like] for models from MDX-Net or SCNet [archs]). Only dango seems to correctly detect kazoo as an instrument it seems" - dca, "The loss function was changed to be more fullness-oriented, and trained a further 50k steps from the v1+ test." Unwa

"v1e keeps better instruments like trumps than v1e+

With v1e+ there is less noise, but some instruments are hidden" koseidon72

"v1e+ has a strange problem of almost vocoding the vocals and keeping them in quietly" even with phase fixer

"has some problems with cymbals bleed in vocals (not the case with other instrumental roformer models)" dca

"trained with additional phase loss which helps remove some of that metallic fullness noise, and also has higher sdr I believe" - becruily

- Unwa released V1+ Mel-Roformer instrumental [model](#) | [yaml](#) | [Colab](#)

Inst. bleedless: 38.26, fullness: 35.31, SDR: 16.72

"It is based on v1e, but the Fullness is not as high as v1e, so it is positioned as an improved version of v1." Unwa

"very nice model, the multistft noise is gone"

It's probably due to:

"Unwrapped phase loss function added" Unwa

BTW. It was already proven before, that adding artificial noise to separations was increasing fullness metric.

"Seems to have significantly less sax and harmonica bleed in vocal, which is an awesome thing (...) It still struggles with other things like FX and Kazoo." dca

"It sounds clean. The only thing [is] that some instruments are deleted, and in some tracks leaves remnants of voice in the instrumental." Fabio

"Screams are not removed from the track" Halif

Training details

"I made a small improvement to the dataset and trained about 50k steps with a batch size of 2.

8192 was added to multi_stft_resolutions_window_sizes.

As it was, the memory usage increased too much, so it was rewritten to use hop_length = 147 when window_size is 4096 or less and 441 when window_size is greater than that."

Unwa

- Mesk released a preview of his instrumental model retrained from Mel Kim on metal dataset consisting of a few thousands of songs.

https://huggingface.co/meskvilla33/metal_roformer_preview/tree/main | Colab

These are not multisong metrics, but made with private dataset!

Instr bleedless: 48.81, fullness: 42.85, SDR: 13.7621

"currently restarting from scratch because I think I know what all the problematic vocal tracks were, and I removed them, we'll see if it's gonna be better"

"vocals could follow if requested.

Should work fine for all genres of metal, but doesn't work on:

- hard compressed screams
- some background vocals
- weird tracks (think Meshuggah's "The Ayahuasca Experience")

P.S: Use the training repo ([MSST](#)) if you want to [separate] with it. UVR will be abysmally slow (because of chunk_size [introduced since [UVR Roformer beta #3](#)])

- Yusuf fixed [Apollo](#) and [AudioSR](#) WebUI Colabs and mid/side method of upscaling was added to Apollo
- Unwa released Big Beta 6X vocal [model \(yaml\)](#)
Vocal bleedless: 35.16, fullness: 17.77, SDR: 11.12

"it is probably the highest SDR or log wmse score in my model to date."

Some leaks into vocal might occur.

"dim 512, depth 12.

It is the largest Mel-Band Roformer model I have ever uploaded."

"I've added dozens of samples and songs that use a lot of them to the dataset"

- (MVSEP) "I added new Apollo model with Aura MR STFT: 22.42

It's available under "Apollo Enhancers (by JusperLee and Lew)" with option:

"Universal Super Resolution (by MVSep Team)".

It requires a hard cutoff on frequency for best experience." - ZFTurbo

lirc, it was trained by his student.

"It's doing well on more transient stuff like snare hits, but it seems to really struggle to actually add harmonics. Has this really weird quality of sounding high quality and low quality at the same time"

"It doesn't seem to like 8 kHz cutoff, it has generated almost nothing"

"I tried with a 10 kHz cutoff and just got quiet-ish transients"

"Lew told me the same while training his, the model would learn transients/drums but struggle with harmonics. Maybe it's an Apollo limitation. I don't recall if the OG model by jasper lee has this issue too, since it rarely works"

Advice

You might want to process your song even 4 times to potentially get better results.

Also, you can split mids and sides, and upscale them separately to get better results, although it's not always better solution ([spectrograms](#) | [tutorial](#)), thx AG89.

Using e.g. MDX23C Similarity/Phantom Centre extraction model instead with 2x slowdown (to reduce smearing artefacts) gives less high-end recovery, but less noise resulting in more proper cancelling of both channels ([spectrograms](#) by AG89).

Avg ensemble will be rather diminishing returns, so consider manual weighted ensemble in DAW.

Getting rid of noise or dithering above real frequencies by making cutoff can make a night and day difference for the result ([example](#))

Sometimes cutting off some more existing frequencies might be beneficial too (the model was trained with hard cutoff)

For noise artefacts after upscaling you can use some [denoisers](#)

- Gabox released new [INSTV8N](#) instrumental model in experimental folder ([yaml](#)) "noticed too many vocal residues. (...) there is no noise" although N stands for noise in its name.
- Some upscaling Colabs are also affected by the last runtime changes in Colab made by Google. Maybe downgrading !pip install torch==2.5 would help.
- We're aware of the issues in some Colabs like [MDX by HV](#) (numpy errors related to its wrong version). Any fixing will be announced. Stay tuned.
- Fixed, but initialization is slow till further notice, and you need to click initialization cell second time when you're prompted to restart environment.
- Fixed, but now you need to click the initialization cell again after Numpy has been installed (happens briefly after launching the initialization cell).
- Unwa's FT3 test vocal model added on x-minus/uvronline
"make vocals sound a bit lower at chorus compared to other parts of songs", doesn't happen with big beta 5e - oak
- ZFTurbo: "I added 2 new super resolution algorithms on MVsep in Experimental section:
1) AudioSR. Metrics: https://mvsep.com/quality_checker/entry/8067
2) FlashSR. Metrics: https://mvsep.com/quality_checker/entry/8071"
Be aware that both can give some errors occasionally. Some problems with mono audio were fixed already.
- Unwa released FT3 preview vocal [model](#) | [yaml](#)

Vocal bleedless: 36.11, fullness: 16.80, SDR: 11.05

“primarily aimed at reducing leakage of wind instruments to vocals.

I will upload a further fine-tuned version as FT3 in the near future.”

For now, FT2 has less leakage for some songs (maybe till the next FT will be released).

- Gabox added some new experimental instrumental models in a separate repo [folder](#).
They are called V8, V9, V10, don't consider them as newer/better, but forgotten to upload in the meantime.
They're less full than V7, but have less vocal residues. Also, the results from V8 and V10 are the same (“inverted polarity between 2 results, and it's just silence”), and also for V9.
“Both remove some instruments from the music, like V7.
As for noise, however, they are less noisy”

- Gabox released [inst_gaboxBv3](#) instrumental model (B for bleedless)
Inst. bleedless: 41.69, fullness: 32.13
“can be muddy sometimes”

- [mesk's training model guide](#) link has been changed (the previous one has been deleted)

- Apart from new drumsep models on MVSEP, also moises.ai has their own drumsep model (paid).

Probably their base drums model used for drumsep is not better than other solutions, so check [this](#) section of the doc to get better drums to separate first to test it out, although one user reported that moises' drums model (free), probably vs Mel-Roformer on MVSEP or x-minus (not sure) can give “better results (...) if the input material is for example cassette-tape sourced or post-FM).

- Joseph made the SESA Colab private till some stuff will be fixed in the future.
Consider using [this](#) Colab with newer models added at this time.

- (x-minus) Inst V7 model by Gabox replaced v1e model by Unwa.

It can be still accessed by these links:

<https://uvronline.app/ai?hp&test> (premium)

<https://uvronline.app/ai?test> (free)

(v1e might be still fuller, and impair fewer instruments in cost of more noise, also be aware that separation on x-minus might differ from Colabs, MSST or UVR, possibly due to different inference parameters)

- Training (and inferencing) locally on Radeon using [MSST](#), specifically RX 7900 XTX, was confirmed to work by Unwa on Ubuntu 24.04 LTS using Pytorch 2.6 for ROCm 6.3.3.

Currently, officially [supported](#) consumer GPUs with ROCm are:

RX 7900 XTX, RX 7900 XT, RX 7900 GRE and AMD Radeon VII. But in fact, there are more consumer Radeons confirmed to work already too.

“No special editing of the code was necessary. All we had to do was install a ROCm-compatible version of the OS, install the AMD driver, create a venv, and install ROCm-compatible PyTorch, Torchaudio, and other dependencies on it.” [More](#)

“So far I have not had any problems. Running the same thing appears to use a little more VRAM than when running on the NVIDIA GPU, but this is not a problem since my budget is not that large and if I choose NVIDIA I end up with 16GB of VRAM (4070 Ti S/4080 S). Processing speeds are also noticeably faster, but I did not record the results on the previous GPU, so I can't compare them exactly.” [More](#)

- [Inst_GaboxFVX](#) model was released (which is “instv7+3” - so probably fuller than instv3) and

- [INSTV7N](#) (so more noisy than INSTV7; “it's [even] closer to fv7 than inst3”) [yaml](#)

- Gabox Karaoke model got updated (links have been replaced, and the old deleted from the repo),

- and also final [INSTV7](#) was released (“I hear less noise compared to v1e, but it has worse bleedless metric”)

- Gabox released instv7 [beta 2](#)

Inst. bleedless: 34.66, fullness: 38.96

and instv7 [beta 3](#)

“Both are noisy with small vocal residuals in places where music is low and deletions of some musical instruments.”

- New 4 stem drumsep SCNet model (kick, snare, toms, cymbals) has been added on MVSEP (best SDR for kick and similar for toms to previous 6s model -0.01 SDR difference), and also 8 stems ensemble of all other drumsep models (besides the older Demucs model by Imagoy) [metrics](#)

- Gabox released [instv7beta](#) model [yaml](#)

Inst. bleedless: 35.01, fullness: 38.39

“sound is good, but sometimes some instruments are lowered or deleted”

“while the annoying buzzing/noise is still present, it seems to be more contained.”

- Gabox released Mel [KaraokeGabox](#) model (uses Aufr’s [config](#)) | [Colab](#)

“the lead vocals are good and clean!

While the backing tracks are lossy for this model, [it still] provide[s] great convenient for those who need LdV”

“The model doesn’t keep the backing vocals below the main vocals, sometimes the backing vocals will be lost even though there are backing vocals there.”

- New [FullnessVocalModel](#) ([yaml](#)) vocal model was released by Aname | [Colab](#)

Voc. bleedless: 32.98 (less than beta 4), fullness: 18.83 (less than big beta 5e/voc_fv4/becruily, more than beta 4)

“While it emphasizes fullness, the noise is well-balanced and does not interfere much. (...) in sections without vocals, faint, rustling vocals can be heard.”

We have some report of very long separation of this model in UVR on Macs.

> Try to change chunk_size: 529200 to 112455 for that model/yaml (but it's dim_t 256 equivalent, so something higher to test might be a better idea too)

- (SESA) No audio file found bug fixed

- “In my testing, I've found that SCNet very high fullness (on mvsep) put through Mel-Roformer denoise (average) and UVR denoise (minimum) has the best acapella result would love to see people's thoughts” dynamic

- Gabox released [voc_fv4](#) | [yaml](#) | [Colab](#)

Voc. bleedless 29.07, fullness 21.33

“Very clean, non-muddy vocals. Loving this model so far” (mrmason347)

“lost some of the trumpet sound while on Becruily model can keep it, but some also was lost”

- Joseph fixed some bugs and errors in SESA Colab, and also added new interface
There are still some issues with auto ensemble till further notice.

- Fixed

- unwa released Big Beta 6 vocal [model](#) | [yaml](#) | [Colab](#)

“Although it belongs to the Big series, the characteristics of the model are similar to those of the FT series. (...) this model is based on FT2 bleedless with the dim increased to 512”
Muddier than Big Beta 5, might be better than FT2 at times.

“If you liked the output of the Big Beta 5e model, you may not like 6 as much; it does not have the output noise problem of 5e, but instead sacrifices Fullness. (...) Simply put, it is a more conservative model” unwa

- To get rid of noise in INSTV6N, use [denoisedebled.ckpt](#) ([yaml](#)) on mixture first, then use INSTV6N - “for some reason it gives cleaner results” (Gabox)

- New Gabox model released: [INSTV6N](#) (noisy) | [yaml](#) | [Colab](#) | [SESA](#) | [metrics](#):

inst bleedless: 32.63, fullness: 41.68 (more than v1e)

Interestingly, some people find it having less noise vs v1e, and more fullness.

Also, it has more fullness vs INSTV6, and more noise.

“v1e sounds like an “overall” noise on the song, while v6n kind of mixes into it.

v6n also sounds like two layers, one of noise that's just there. And the other one mixes into the song somehow.

Using the phase swap barely makes it any better than phase swapping with v1e though” - vernight

Also Kim model for phase swap seems to give less noise than unwa ft2 bleedless

- Demudder in UVR using at least DirectML (Intel/AMD) works only if "Match freq cut-off" is enabled in MDX settings. Otherwise, you'll get "Format not recognised" error.

- SESA Colab might undergo some issues with hyper_connections at the moment.
It might be fixed tomorrow.

- Done

- [SESA](#) Colab update:

Voc_Fv3 (by Gabox)
dereverb_mel_band_roformer_mono (by anvuew)
MelBandRoformer4StemFTLarge
INSTV5N (by Gabox)
denoisedebleed (by Gabox)

- Gabox released [denoisedebleed.ckpt](#) | [yaml](#) | [Colab](#) for noise from fullness models (tested on v5n) - it can't remove vocal residues

- Aname released small inst/voc 200MB Mel-Roformer with null target stem ([link](#))

- [v5_noise](#) inst model released | [yaml](#) | [metrics](#) | [Colab](#)

- New Gabox vocal model released: [voc_Fv3.ckpt](#) | [yaml](#) | [Colab](#)
Enthusiastic opinions so far

- [INSTV6](#) by Gabox and De-reverb (Mono) by anvuew models added on x-minus | [Colab](#)
V6 "is slightly better than v5" (although not for everyone), but "v1e still gives better fullness, but noise [in v1e] is a problem" old viperx 12xx models have less problems with sax.

- (added on MVSEP as SDR 13.72) ZFTurbo trained new SCNet XL model for drums.

"I have 2 versions: one is slightly higher SDR and avg Bleedless.

Second is better for fullness and L1Freq.

Previous best SDR model had 13.01 (it's SCNet Large)." [Metrics](#)

15.7180 (13.72) one has much better fullness metric.

"It's far superior to the other one, but I still hear some weird parts.

It still messes up on some percussion.

The drums stem sounds really weird.

The no drums is alright except for some bleeding but yeah the drums is quite muddy" - insling

- Gabox released new fine-tunes of his inst Mel-Roformer models ([click](#)):
[inst_gabox2.ckpt](#), [inst_gabox3.ckpt](#), [INSTV5.ckpt](#), [INSTV6.ckpt](#)

with one opinion that the last one is his best inst model so far.

"seems like a mix between brecuily and unwa's models"

"confuses way less instruments for vocals than v1e, but it's still not as full as v1e (...) But it's a very good model"

Rarely it can give "Run out of input error" in UVR when installing using the new Model install option (moved model has 0 bytes), while V5 worked correctly, then move the ckpt to Ultimate Vocal Remover\models\MDX_Net_Models manually.

- We're aware that the x86-64 version of the latest UVR patch for Mac went offline.
Anjok was pinged about it.

- anvuew released new [dereverb_mel_band_roformer_mono_anvuew_sdr_20.4029](#) model.

"supports mono, but ability to remove bleed and BV is decreased

should not matter whether it's singing or speech, because my dataset contains speech."

[Colab](#)

- MedleyVox Colab is currently broken (you can use MVSEP instead)

> fixed:

https://colab.research.google.com/drive/10x8mkZmpqiu-oKAd8oBv_GSnZNKfa8r2?usp=sharing (although initialization now takes 7 minutes, GDrive integration added)

- Phase remix functionality was added to SESA model inference Colab

https://colab.research.google.com/drive/1U28JyleuFEW6cNxQO_CRe0B2FbNoiEet

- (MVSEP) ZFTurbo added new SCNet XL "high fullness" and "very high fullness" models on the site ([metrics](#)).

They're good for both vocals and instrumentals, and sometimes are fuller than v1e, although with more noise, which can be too strong for some people, but not all.

"very high fullness" variant have both vocals and instrumental fullness and bleedless metric better than the "high fullness".

"they also correctly detect "complex" (for the AI) instruments as part of the instrumental track rather than in vocals (like flute or sax for example), which isn't the case for v1e and Fv5.

Example: sax solo of Shine On You Crazy Diamond detected the sax solo as part of acapella using v1e or Fv5 or brecuily inst." dca100fb8

The noise "gonna go nuts with distortion, compression and other vct plugins" John.

Both variants "have the same amount of buzzing noise"

"Instrumentals are very good. It's holy shit level. Unwa v1e/Gabox Fv5 are still amazing, it's just nice to have such a decent model like these new ones on a different arch" dca

From the songs I've tested, SCNet is incredible. Very full sounding" mrmason347

"Regular high fullness though has a less full instrumental but quite good acapella"
theamogusguy

VHF leaves some vocal residues in metal, but seems to do well for e.g. alt-pop.

"scnet doesn't pick up the drone backing vocals, but 10.2024 has mad violin bleed in the vocals" dynamic64

For "mainly orchestral tracks with choir" "it gave me noticeably fuller results than v1e"
Shintaro5034.

"For noisy/dense mixes though, Roformers are probably better, especially for inst.
scnet seems better at preserving treble in some vocals. These high fullness models
especially so. So maybe teaming SCNet up with Roformer might give a nice middle ground"
"Rofos are really bad for some kinds of EDM that are very aggressive (Dubstep, Trance,
Breakcore, etc...), also it has a very hard time with Experimental (IDM)"

VHF seems to have more crossbleed in some songs, along with also basic XL model. Some
songs which sound full enough even with basic SCNet XL. While others sound muddy (dca)

- (X-Minus) Mel Kim model has been replaced for phase correction by Unwa's Kim FT2
model for premium users

- New [sites and rippers](#) added:

<https://yams.tf/> (Qobuz, Tidal, Spotify, Apple Music [currently 320kbps], Deezer) - for URLs
<https://us.deezer.squid.wtf/> (Deezer only) - for queries

<https://github.com/lmAiiR/QobuzDownloaderX> (local ripper for premium accounts or provided
ARLs)

- [FlashSR](#) has been released ([Colab](#) with chunking and overlap by jarredou).

It's a diffusion distillation of AudioSR, and has lower Aura MR STFT [metric](#), and usually
lower quality as well, but it might give better results for music for some people

- (MVSEP) "We trained new DrumSep models (5 stem and 6 stem) based on SCNet XL.

* 5 stems: cymbals, hh, kick, snare, toms

* 6 stems: ride, crash, hh, kick, snare, toms"

Both have better SDR than the previous MDX23C model by jarredou and Aufr33.

The 5 stems variant has e.g. better snare SDR than the 6 stems variant. [Full metrics](#).

It doesn't work correctly on the site yet, it will be announced in the link above by ZFTurbo
when it will be fixed.

- They work already

- Unwa released ft2 bleedless vocal model | [Colab](#)

<https://huggingface.co/pcunwa/Kim-Mel-Band-Roformer-FT/tree/main>

voc bleedless 39.30 | fullness 15.77 | SDR 11.05

- instv5 model released by Gabox (39.40 inst fullness | inst. bleedless 33.49) [link](#) | [yaml](#) |
[Colab](#) | x-minus

"it seems that most vocal leakage is gone, and the noise did significantly decrease, although
there's still a bit more noise presence than v1e."

In terms of fullness though, for some reason it sounds as if it's actually less full than v1e,
despite the higher instrumental fullness SDR.

Despite v4's significant amount of noise, it seems to be the only model that gave me a fuller
sounding result compared to v1e that's actually perceivable by my ears." Shintaro

- New inst/voc SYH99999 models released
<https://huggingface.co/SYH99999/MelBandRoformerSYHFTB1/tree/main>
- (x-minus) Phase fixer added for Gabox fv3 and becruily models for premium users
- For vocals, you can alleviate some of the noise/residues in unwa's 5e model by using phase fixer/swapper and using becruily vocals model as a reference (imogen).
- For instrumentals, you can try unwa's v1e with phase swap at 500/500 with original mel band of Kim. It consistently gives less noise - midol

"500 / 500 means you use original phase below 500 Hz and hard cutoff/swap to transferred phase above 500hz. (this can potentially create phase artifacts at 500Hz because of hard swap)

500 / 20000 means you use original phase below 500hz and progressively crossfade to transferred phase until 20000hz and transferred phase is used above 20000hz. So it's softer phase swap below 20kHz" - jarredou

"using 500 on both parameters really does make me have the illusion that I have produced the official instrumental. Even tho it's unofficial haha" - midol
- Deezer on Lucida doesn't work. Doubledouble.top came back (probably temporarily), but returns mp3 128kbps from Deezer now. Also, it supported Apple Music unlike Lucida, but now it doesn't work, (check current services [status](#)). Besides, occasionally it can happen that rips from Amazon only on doubledouble have quality higher than 44/16. Plus, downloading full albums frequently fails, while single songs downloading works.
- Lots of new Gabox models added since then, including:
 - a) BS-Roformer instrumental variant, which doesn't struggle so much with choirs like most Mel-Roformers, although may not help in all cases ([link](#))
 - b) [inst_gaboxFv3.ckpt](#) - like v1e when it comes to fullness (added on x-minus)
 Inst SDR 16.43 | inst. fullness 38.71 | inst. bleedless 35.62
 It might pick up entire sax in vocal stem.
- Gabox models have been added to SESA [Colab](#) (you'll find more info about them later below).
- Along with UVR Roformer beta patch #14, Anjok released the long anticipated **demudder**. It's in Settings > Advanced MDX options (so works only for Roformers and MDX models). It consists of three methods to choose from (each separates your twice):
 - Phase Rotate

- Phase Remix (Similar to X-Minus) - “the fullest sounding, but can leave a lot of artifacts with certain models. I only recommend that method for the muddiest models. Otherwise, Combined Methods is the best” “I don't recommend using phase remix on the Instrumental v1e model. I recommend combined methods or phase rotate for models produce fuller instrumentals.” Anjok

It might leave some choruses when using V1E (Fabio)

- Combine Methods (weighted mix of the final instrumentals generated by the above). More in the [full changelog](#). You cannot use demudder on 4GB AMD GPUs with 800MB Roformers with even 2 seconds chunk size set (memory allocation error).

“It's meant to solely target instrumentals. The vocals should stay exactly as before.

For Roformer models, it must detect a stem called "Instrumental" so for some models like Mel-Kim, you need to open model's corresponding yaml, and change “other” to “instrumental”.

“I've noticed with the few amounts of tracks I've tried, demudding can sometimes accentuate instances of bleeding or otherwise entirely missed vocal-like sounds”

In case of file not found error on attempt of using demudder, reinstall UVR.

“I put the demuddled instrumental in the bleed suppressor, and it sounds really good, almost noise free. I either do a bleed suppressor or a V1/bleed suppressor ensemble” gilliaan

“With the new config editor feature you could probably edit the configs of models to have the vocal stem labelled as the Instrumental stem so the demudder demuds the vocal stem, it definitely still makes a difference.

I accidentally did this when installing another model, but it seems to actually have an effect on vocal stems too.

You just change the target instrument from vocals to instrumental I think (don't move the stems around)

You can verify it works if the stems are the other way around when processing (vocals are in the file labelled as Instrumental). Then you can use the demudder on the vocals that way I think. If you want to use the demudder with other models that aren't labeled with instrumental, you'll have to select the stem you want to demud and replace it with Instrumental.

Though demudding the vocal stem will definitely make it quite noisy depending on what model you use, though there [appears](#) to be instances where demudding the vocal stem can mildly help with certain effects but i did not test this enough” stephanie

Anjok: “Just a few quick notes on the Demudder:

It works best on tracks that are spectrally dense (ex. Metal, Rock, Alternative, EDM, etc.) I don't recommend it for acoustic or light tracks.

I don't recommend using it with models that emphasize fuller instrumentals (like Unwa's v1e model).

I do plan on adding options to tweak the phase rotation.
I also plan on adding another combination method that may work better on certain tracks.”

UVR_Patch_1_21_25_2_28_BETA:

Small patch (you must have a [Roformer Patch](#) [e.g. #13] previously installed for this to work):
[Link](#)

Also, minor bugs fixed, calculate compensation for MDX-Net v1 models added.

The MacOS version will be released later ([observe](#)).

Be aware that at least Phase Rotate doesn't work on AMD and 4GB VRAM GPUs on even 88200 chunk size (prev. dim_t 201 - 2 seconds) and 800MB Roformers like Becriuly's, while 112455 (2,55s, prev. dim_t = 256) works just fine for normal separation.

- BS-RoFormer 4 stems model by yukunelatyh / SYH99999 added on x-minus
Since then, a new version was added (later epoch, but it has lower SDR for all stems).

<https://uvronline.app/ai?discordtest>

Some people liked the v1 more than Demucs, but “it's like demucs v4 but worse i think the vocals have a ton of bleed, the bass is disappointing tbh the other stem has a ton of bgv and adlib bleed in it” lsling

It has [SDR metrics](#) for all stems worse than 4 stem BS-Roformer by ZFTurbo and demuics_ft.

- Aname also released 4 stem BS-Roformer [model](#) | [yaml](#)

It has better SDR than the above (as in the SDR metrics link above), but worse than the other two mentioned

- Gabox released Mel-Roformer instrumental model (Kim/Unwa/Becriuly FT):

<https://huggingface.co/GaboxR67/MelBandRoformers/tree/main/melbandroformers>

inst bleedless: 37.40 (better than v1e by 1.8), fullness 37.07 (better than unwa inst v1 and v2)

“It's like the v1 model with phase fixer, but it gets more instruments, like, it prevents some instruments getting into the vocals”, “sometimes both models don't get choirs”.

- instrumental variant called fullness v1 (“noisier but fuller”)

inst bleedless: 37.19, fullness 37.26

(thanks for evaluation to Bas Curtiz and his [GSheet](#) with all models.)

- fullness v2 released

- fullness v3 released

- B (bleedless) v1/v2 variants released

- voc_gabox.ckpt:

voc bleedless: 34.66 (better than 5e), fullness 18.10 (on pair with beta 4)

- Vocal model F v1

- Vocal model F v2

voc bleedless: 33.4013, fullness: 19.3064

- Issues with dataset 4 in MSST repo were fixed

"I think that issue could also explain why training de-reverb models with pregenerated reverb audio files was not working that well, as reverb was not aligned with clean dry audio as it should have been." jarredou ([more](#))

- Aufr33 BS-Roformer Male/Female beta ([model](#) | [config](#) | [config](#) for UVR | tensor match error [fix](#)) added on [Colab](#) (based on BS-RoFormer Chorus Male Female by Suciial) along with Unwa's Kim FT2

- Anjok released the MacOS versions of UVR Roformer beta patch #13.1 applying hotfix to address a few graphics issues:

- Mac M1 (arm64) users - [Link](#)
- Mac Intel (x86_64) users - [Link](#)

- Anjok released UVR beta Roformer patch #13 for Mac (Windows further below):

UVR_Patch_1_15_25_22_30_BETA:

- Mac M1 (arm64) users - [Link](#)
- Mac Intel (x86_64) users - [Link](#)

- mesk wrote a good comprehensive [training guide](#) for beginning model trainers. Later you can proceed to read further [section](#) of this doc for more details and arch explanations

- Apple Music bot link added in [this](#) section (thx mesk)

- mrmason347 and Havoc shared an interesting method to get cleaner vocals. The last point of tips to enhance separation [here](#):

Separate with becruily Mel Vocal model and its instrumental model variant, then get vocals from the vocal model, and instrumental from instrumental model, import both stems for the DAW of your choice (can be Audacity) so you'll get a file sounding like original file, then export - perform a mixdown of both stems, then separate it with vocal model

- If you somehow still struggle with "norn" issues in UVR, see at the bottom of the section [here](#)

- Dango released "Reverb Remover" - [click](#)

"it's very similar to RX11 Dialogue Isolate, good/real-time set to 5
it's like listening to the same inference files" John; probably also works in mono, you can get 30 seconds for free)

- [filegarden](#) added to the [list of cloud services](#) (seems to be unlimited, registration required, link shortener with custom name available)
- [your-good-results](#) and [your-bad-results](#) channels have been reopened on the server, but you need to paste links to uploads instead of uploading audio files directly on Discord due to copyright issues the server was undergoing

- If you want to use Phase fixer Colab with cut-offs suggested by CC Karaoke, check [here](#)
- Unwa's Kim FT2 model added to the [inference Colab](#) (both inst and voc becruily models are added too)
- jarredou released [Custom Model Import](#) Version of the inference Colab. You can use it if we don't add any new model to the main Colab on time, or you test your own models.

Just make sure that pasted link haven't "downloaded the webpage presenting the model instead of the model itself."

So, e.g. for yaml pasted from GH, use:

https://raw.githubusercontent.com/ZFTurbo/Music-Source-Separation-Training/main/configs/config_vocals_mdx23c.yaml

Instead of:

https://github.com/ZFTurbo/Music-Source-Separation-Training/main/configs/config_vocals_mdx23c.yaml

And for HF, follow the pattern presented in the Colab example (so with the resolve in the file address)

- [model_fusion.py](#) by Social

This script seems to save the weighted ensemble of three models into a checkpoint called "fused". The result is not bigger than a single model.

Probably you could basically create one checkpoint getting the same or similar results of manually weighted models, and not inference every of them one by one.

- Beccruily models added on MVSEP and instrumental on variant on x-minus
- ZFTurbo "added new organ model: MVSep Organ (organ, other).
Demo: <https://mvsep.com/result/20250116160630-f0bb276157-mixture.wav>"
- Anjok released a patch #13 fixing following issue with no sound on some Roformer models (like avvew's and social's de-reverb) on GTX 10XX or older (Windows):

UVR_Patch_1_15_25_22_30_BETA:

- Full Install: [Link](#)
- Patch Install (use if you still have non-beta UVR installed): [Link](#)
- Small Patch Install (have a Roformer patch previously installed for this to work): [Link](#)

The issue was some older GPU's are not compatible with Torches "Inference Mode," (which is apparently faster) so it's now using "No Grad" mode instead. Users can switch back to using "Inference Mode" via the advanced multi-network options.

The MacOS version will be released in a few days. I just need to finish testing out all the models and networks and ensure all the kinks are worked out." [More](#)

- Users undergo some issues (no sound) with Mel-Roformer de-reverb by anvew (a.k.a. v2/19.1729 SDR) since the latest UVR beta #11/12 updates (the issue seems to occur only on GTX 10XX series, and maybe older). Anjok's working on the issue.
You should be able to use more than one UVR installation at the same time when one's been copied before updating (patch #10 still works) or use MSST repo and/or its GUIs.

- Anjok released patch #12 which is a hotfix for the [4 stem](#) BS-Roformer model by ZFTurbo (trained on MUSDB)

[UVR Patch 1_13_0_23_46_BETA_rofo_fixed.exe](#) (Windows only)

- Anjok released a new UVR beta Roformer patch #11 (Windows only for now):

[UVR_Patch_1_13_0_23_46_BETA_rofo](#)

It fixes 4 bugs: with VR post-processing threshold, Segment default in multi-arch menu, CMD will no longer pop-in during operations, and error in phase swapper.

[More](#) details/potential updates.

Standalone (for non-existent UVR installation)

[UVR_1_13_0_23_46_BETA_full.exe](#)

For 5.6 stable (so for non-beta Roformer installation)

[UVR_Patch_1_13_0_23_46_BETA_rofo.exe](#)

Small (for already existing Roformer beta patch installation)

[UVR_Patch_1_13_25_0_23_46_rofo_small_patch.exe](#)

- New beta UVR Roformer patch #10 released by Anjok (for now, only small patch for already existing [beta Roformer](#) installation is available, and only for Windows, check [here](#) for Mac later)

[UVR_Patch_1_9_25_23_46_BETA_rofo_small_patch](#) - [Link](#)

Added SCNet and Bandit archs with models in Download Center (SCNet models using ZFTurbo's unofficial code update will not work since they appear to require a library "mamba_ssm" that is only available in Linux), fixed compatibility with some newer Roformer models (wesley's MDX23C and Roformer Phantom center models, and 400MB inst small by Unwa), new Model Installer option added, model configuration menu enhanced, allowing aliases to selected models, added compatibility for Roformer/MDX23C Karaoke models with the vocal splitter, VIP code issue is gone, issues with secondary models options and minor bugs and interface annoyances are addressed, "improved the "Change Model Settings"

menu. Now, any existing settings associated with a selected model are automatically populated, making it easier for users to review and adjust settings (previously, these settings were not visible even if applied).”.

If you have Python DLL error on startup, reinstall the last beta update using the full package instead, then the small installer from the newer patch.

“If you see a different usage of VRAM than with previous Roformer beta version, it could also be because the new beta version doesn't rely on 'inference.dim_t' value anymore (if you were using edited "dim_t" value)

You have to edit audio.chunk_size now (see [here](#) for conversion between dim_t and chunk_size” it's “In model yaml config file, at top of it, chunk_size is first parameter (...) you can edit model config files directly inside UVR now.”

“Unfortunately, SCnet is not compatible with DirectML, so AMD GPU users will have to use the CPU for those models.

Bandit models are not compatible with MPS or DirectML. For those with AMD GPU's and Apple Silicon, those will be CPU only.

The good news is those models aren't all that slow on CPU.” - Anjok

Annoying CMD window will randomly pop up again when ffmpeg and Rubber Band are used. Regression will be fixed.

- Newer Mel-Roformer Male/Female model was added by ZFTurbo on MVSEP (SDR: 13.03 vs 11.83 - the previous SCNet one, and much better bleedless metric 41.9392 vs 26.0247 with only 0.2 fullness decrease)

“I find it acts differently from Rofo or UVR2. Sometimes it's the one of the three that gets it right., and not strictly for male/female.” CC Karaoke

- Aufr33 released his own BS-Roformer Male/Female (currently beta) model based on BS-RoFormer Chorus Male Female by Sucial.

“this model only works with vocals. You need to pre-isolate the vocals.”

Added on MVSEP and x-minus for premium (in the new Other menu).

Weights:

https://mega.nz/file/XZwV2QwB#5nvWpmvtoBMTJkpor-IMUZCbBZWDH-3i52ELJS_JmcU

Config:

https://huggingface.co/Sucial/Chorus_Male_Female_BS_Roformer/blob/main/config_chorus_male_female_bs_roformer.yaml

- Unwa released a new version of his Mel-Kim fine-tune (ft2)

<https://huggingface.co/pcunwa/Kim-Mel-Band-Roformer-FT/tree/main>

It tends to muddy instrumental outputs at times, similarly like the OG Kim's model was doing, which didn't happen in the previous ft model by Unwa.

[Metrics](#). PS. All unwa models were trained on 3060 Ti!

- Unwa released 400MB experimental BS-Roformer inst model

<https://huggingface.co/pcunwa/BS-Roformer-Inst-EXP-Value-Residual>

It's using a new Value Residual Learning added to Roformer arch by Lucidrains in the OG Roformer. If it wasn't made compatible with MSST repo already, replace bs_roformer.py from this [repo](#) and

from bs_roformer.attend import attend

↓

from models.bs_roformer.attend import attend
in bs_roformer.py file

"I think it sounds better than large rn but still not good, needs some [more] epoch[s]!"
[later the VRL was added as Roformer v2 in [UVR](#) so it's compatible with the model]

- New dereverb model(s) released by Sucial - "fused": [model](#) | [yaml](#)

"trained two new models specifically targeting large reverb removal. After training, I combined these two models with my v2 model through a blending process, to better handle all scenarios. At this stage, I am still unsure whether my new models outperform the anvuew's v2 model overall, but I can confidently say that they are more effective in removing large reverb." [More](#)

- Beccruily Mel inst and voc models added on MVSEP and inst variant on x-minus

- ZFTurbo released new models on MVSEP:

a) a new Male/Female separation model based on SCNet XL

SDR on the same dataset: 11.8346 vs 6.5259 (Sucial)

Model only works on vocals. If the track contains music, use the option to "extract vocals" first. Sometimes the old Sucial model might still do a better job at times, so feel free to experiment.

b) SCNet XL (vocals, instum)

Inst SDR: 17.2785

Vocals have similar SDR to viperx 1297 model,
and instrumental has a tiny bit worse score vs Mel-Kim model.

- "All Ensembles on MVSep were updated with latest release [SCNet XL] increasing vocals SDR to 11.50 -> 11.61 and instrum SDR: 17.81 -> 17.92".

- ([MSST](#)) You can now inference mono files without any issue

- You can now use "batch_size=1 without clicks issues (with overlap >= 2 of course)" - jarredou

- *Beccruily's released instrumental and vocal Mel-Roformer models* | [Colab](#) | [UVR](#) beta | [Instrumental](#) model files | Inst SDR 16.4719 | inst fullness 33.9763 | bleedless 40.4849 [Vocal](#) model file | Vocals SDR 10.5547 | voc fullness 20.7284 | bleedless 31.2549 | [config](#) with ensemble fix in UVR.

Instrumental model is as clean as unwa's v1, but has less noise and, and it can be got rid well by Mel [denoise](#) and/or Roformer [bleed suppressor](#). Inst variant "removed some of the faint vocals that even the bleed suppressor didn't manage to filter out" before". Doesn't require phase fix from Mel-Kim like unwa models below.

"it handles the busy instrumentals in a way that makes VR finally an arch of the past"
Correctly removes SFX voice. More instruments correctly recognized as instruments and not vocals, although not as much as Mel 2024.10 & BS 2024.08 on MVSEP, but still more than unwa's inst v1e/v1/v2. (dca100fb8).

Trumpet or sax sound which on unwa model was lost, can be recovered on becruily's model (hendry.setiadi)

The instrumental model pulled out more adlibs than the released vocal model variant - it pulled out nothing (isling).

"Vocal model pulling almost studio quality metal screams effortlessly. Wow, I've NEVER heard that scream so cleanly" (mesk)

The model was trained on dataset type 2 and single RTX 3090 for two days (although with months of experimentation beforehand). SDR metrics are lower than Mel-Kim model.

If you use lower dim_t like 256 at the bottom of config for slower GPU, these are the first models to have very bad results with that setting.

You can experiment with phase fixer with santilli_ suggestion "Using becruily's vocals as source and inst [model] as target, and changing high frequency weight from 0.8 to 2 makes for impressive results".

- [Phase fixer Colab](#) (update 2) by santilli_ released - it can use e.g. Mel-Kim model phase for unwa's v1e/v1/v2 models to automatically get rid of some noise during separation (it might no longer work due to the last changes in MSST repo), it includes also becruily models

- A small UVR Roformer beta patch #9 fixing mainly Apollo arch released also for Mac (UVR_Patch_12_8_24_23_30_BETA):

Mac M1 (arm64) users - [Link](#)

Mac Intel (x86_64) users - [Link](#)

- New "MVsep Bass (bass, other)" SCNet model available on MVSEP

"It achieved SDR: 13.81. In Ensemble it gives 14.07 - which is a new record on the Leaderboard." ZFTurbo

"It passes Food Mart - Tomodachi Life test. That's the first model to."

"All bass models have problems with fretless bass"

There's already an option to combine all SCNet+BS Roformer+HTDemucs bass models for 14.07 SDR.

Ensembles have been updated with this model too.

- Reverb removal by Sacial v2 (Mel-Roformer) [model](#) added on MVSEP (update of the previous model)

- Lew universal upscaling [model](#) has been added on x-minus/uvronline too (premium users). Just a reminder - it's not for badly mixed music, it's for lossy files (also on [Colab](#)/MVSEP/UVR [beta](#) [at least support for a model file])

- ZFTurbo released a new 4 stem XL [model](#) trained on SCNet. "I have great results comparing with SCNet Large model (by starrytong)."

SCNet Large MUSDB test avg: 9.70 (bass: 9.38, drums: 11.15 vocals: 10.94 other: 7.31)
SCNet XL MUSDB test avg: 9.80 (bass: 9.23, drums: 11.51 vocals: 11.05 other: 7.41)

SCNet Large Multisong avg: 9.28 (bass: 11.27, drums: 11.23 vocals: 9.05 other: 5.57)
SCNet XL Multisong avg: 9.72 (bass: 11.87, drums: 11.49 vocals: 9.32 other: 6.19)

A new SCNet bass model is incoming and already surpassed metrics of ZFTurbo's HTDemucs and BSRoformer bass models.

- Anjok released a small UVR Roformer beta patch #9 fixing mainly Apollo arch:
[UVR_Patch_12_8_24_23_30_BETA](#)

Windows only for now: [Full](#) | [Patch](#) (Use if you still have non-beta UVR installed) | [Small Patch](#) (You must have a Roformer patch previously installed for this to work)

Changelog:

Apollo fixes: "Chunk sizes can now be set to lower values (between 1-6)

Overlap can be turned off (set to 0)"

Fix both for Apollo and Roformers: now 5 seconds or shorter input files no longer cause errors.

OpenCL was wrongly referenced in the UVR. It was actually DirectML all the way, and Anjok changed all the OpenCL names in the app into DirectML.

- Unwa released a new Kim-Mel-Band-Roformer-FT vocal [model](#) | [Colab](#)

It enhances both our new bleedless (36.95 vs 36.75) and fullness (16.40 vs 16.26) [metric](#) for vocals vs the original Mel Kim model. [SDR](#)-wise it's also a tad lower (10.97 vs 11.02) (thx Bas Curtiz)

- Male/female BS-Roformer separation model has been released by Sacial

<https://github.com/ZFTurbo/Music-Source-Separation-Training/issues/1#issuecomment-2525052333>

If they sing at intervals (one by one), they cannot be separated.

Works pretty good, bleed might occur occasionally. Also, it seems to pick up various people dialogues.

If you want to use the model in UVR, use [this](#) config (thx Essid)

If you have "The size of tensor a (352768) must match the size of tensor b (352800) at non-singleton dimension 1" e.g. in python-audio-separator, use [this](#) config (thx Eddycrack864)

- Anjok released UVR Roformer beta patch #8 for Win: [full](#) | [patch](#) | Mac: [M1](#) | [x86-64](#)

- UVR_Patch_12_3_24_1_18_BETA

Apollo arch was made compatible with MacOS MPS (metal) and OpenCL, but with it, it might be unstable and very RAM intensive - use chunk size over 7 to prevent errors (currently it's not certain that all models will work with less than 12GB of VRAM).

Apollo is now compatible with all Lew models (fixed incompatibility with any other than previously available in Download Center). Fixed (presumably regression with) Matchering.

How would I assign a yaml config to an Apollo model on the new UVR [patch]?

“1. Open the Apollo models folder

2. Drop the model into the folder

3. From the Apollo models folder, drop the yaml into the model_configs directory

4. From the GUI, choose the model you just added and if the model is not recognized, a pop-up window will appear, and you'll have the option to choose the yaml to associate with the model.” - Anjok

“I found some overlapping issues in the UVR [using Apollo vs Colab]. like some short parts sounding duplicated overlaid” The issue is caused by different chunking, which on Colab is preconfigured to use 15GB of VRAM. “chunk_size has influence on results, colab uses 25sec (or 19 for latest lew model)”

- Anjok released UVR Roformer beta patch #7 for Windows: [full](#) | [patch](#)

- UVR_Patch_12_2_24_2_20_BETA (version for Mac probably tonight, [observe](#) or on [GH](#))

It introduces support for Apollo arch. The OG mp3 enhancer and Lew v1 vocal enhancer were added to Download Center. Probably, now you'll be able to add newer Lew uni enhancer and v2 vocal enhancer manually. The arch is located in Audio Tools. Sadly, this arch cannot be GPU-accelerated with OpenCL, so using AMD and Intel cards (you're forced to use CPU, which might be long).

Also, “Phase Swapper” a.k.a. Phase fixer for Unwa inst models was added to Audio Tools.

- New SCNet and MelBand DnR v3 (SFX) models were added on MVSEP (along with optional ensemble). “The metrics turned out to be better than those of the similar model Bandit v2” (25.11)

- We fixed some issues (IndexError) with jarredou’s inference Colab due to the recent updates in the ZFTurbo code (thx for the heads-up, MrG).

- Anjok released UVR Roformer beta patch #6 for MacOS as well: [M1](#) | [x86-64](#)

- Lew released a new Apollo [universal model](#) for upscaling various lossy audio files (added on MVSEP and x-minus premium).

Unlike the previous mp3 model, it’s able to enhance any formats and not only mp3, including files with hard cutoff like in AAC 128 kbps ([see](#)), it struggles with 48 kbps files.

"If anyone wants to run the new model in [Colab](#) [already added], set chunk_size to 19. Then the model uses 14.7GB VRAM" (Essid). Sometimes a lower setting is necessary (e.g. for a 3 minute song, otherwise memory error will appear).

As for 27.11.24 it doesn't work with MSST yet, later added support in UVR [patch](#).

"Actually much better than the original Apollo model. It handles artifacts really well and also noise, it understands noise while [the] OG model doesn't for some reason" John UVR/simplcup

Specifically for any muddy Roformer vocals, still use Lew vocal enhancer v1/2 as they're better for this task, though they can be noisy (available in the [Colab](#)).

"I also included [checkpoint](#) you can continue training from" ([mirror](#))

Q: segments: 5.4 - can I assume that chunck_size if 5.4 * 44100?

A: Yes, and dims 384

The smaller one for inference and bigger one for training

Q: Does that model has a dataset of a wide variety of compression noise and artifacts?

A: mp2, mp3, ogg, wma, aac, opus, low band width wavs. Random speed change augmentations were used too." "It was mostly trained on music" Lew

- Two weeks ago, unwa and 97chris released a [bleed suppressor](#) Mel Roformer model dedicated for instrumentals (made with unwa v1 in mind). It can work with e.g. v1 and v1e. Sometimes it can remove some bleed also after using [phase fixer](#) (by becruily) dedicated for v1 model, or used also on x-minus for premium users

- Anjok [released](#) a new UVR Roformer beta patch #6

UVR_Patch_11_25_24_1_48_BETA (Windows: [standalone](#) | [patch](#) | Mac: [M1](#) | [x86-64](#)) addressing "All stem" error issue with viperx' models.

And with it, a long anticipated MDX-Net HQ_5 model has been released | [Colab](#) | MVSEP (it's also added to Download Center in previous UVR patch versions).

~~New version of the HQ_5 model is announced to be released in two weeks already.~~

Instrumentals are slightly muddier than in HQ_4, but vocal residues are also a bit quieter (although rather still present where they were before, maybe with some exceptions).

E.g. some hi hats might get a tad quieter in the mix.

The new model variant in two weeks was said to have fuller instrumentals.

vs unwa's v1e "HQ5 has less bleed but is prone to dips in certain situations. (...). Unwa has more stability, but the faint bleed is more audible. So I'd say it's situational. Use both. (...) Splice the two into one track depending on which part works better in whichever part of the song is what I'd do." CC Karaoke

[Model](#) | config: "compensate": 1.010, "mdx_dim_f_set": 2560, "mdx_dim_t_set": 8, "mdx_n_fft_scale_set": 5120

- We have some reports about user custom ensemble presets from older versions no longer working (since 11/17/24 patch and in newer ones). Sadly, you need to get rid of them (don't restore their files manually) or the ensemble will not work and model choice will be greyed out. You need to start from scratch.

- Sucial released a new Mel-Roformer dereverb/echo model ([model](#) | MVSEP). It's good but doesn't seem to be better than the more aggressive variant of Mel anvew's model ([models list](#)).

Still, might depend on a use case.

- People experience All stems error with viperx'12xx models in newer versions of UVR Beta Roformer patch (patch #2 was the last confirmed to work with these older models)

- Lucida.to is undergoing some issues with Qobuz links. Tidal and Deezer work, but poorly, occasionally giving errors too, just retry. Doubledouble redirects to Lucida now. In case of problems with accessing the domain in your country, check out lucida.su or VPN. If you have any problems during downloading files, try out in incognito mode without any browser extension, also download accelerators might cause issues too (FAQ).

- Unwa released a new beta 5 model dedicated for vocals | [Colab](#) | [MSST-GUI](#) | [UVR instr](#) <https://huggingface.co/pcunwa/Mel-Band-Roformer-big/tree/main> | yaml: big_beta5e.yaml
It seems to fix some issues with trumpets in vocal stem (maxi74x1).
It handles reverb tails much better (jarredou/Rage123).

“It’s noisy and, IDK, grainy? When the accompaniment gets too loud. (...) Definitely not muddy though, which is a welcome change IMHO. I think I prefer beta 4 overall” - Musicalman

“to me, the noise sounds similar to how VR arch models sounded, except it’s not poor quality”

“Perhaps a phase problem is occurring (...) The noise is terrible when that model is used for very intense songs” - unwa

Phase fixer for v1 inst model doesn’t help with the noise here (becruily).

“it’s a miracle LMAO, slow instrumentation like violin, piano, not too many drums... it’s perfect... but unfortunately it can’t process Pop or Rock correctly” gilliaan

“feel so full AF, but it has noticeable noise similar to [Apollo] lew’s vocal enhancer”

“the vocal stem of beta5e may have fullness and noise level like duality v1, but it may also suffer kind of robotic phase distortion, yet may also remove some kind of bleed present in other melrofo’s.” Alisa/makidanyee

“bigbeta5e is particularly helpful when you invert an instrumental and then process the track with it. It really keeps the quality. Even if the instrumental was a lossy mp3 inverted to a lossless flac file, it cleans it up without making a mess. (...) some songs gets their

instrumentals leaked online. And a lot of the time it's a lossy 160kbps mp3 file or even worse, you invert that instrumental file to the real song and process the result using bigbeta5e [to clean the invert] gilliaan/heauxdontlast

"Ensemble AVG Big Beta 4 + Big Beta 5e is really good to reduce the noise while keeping the fullness" - heauxdontlast

- Unwa released a new Inst v1e model ("The model [yaml] configuration is the same as v1")
<https://huggingface.co/pcunwa/Mel-Band-Roformer-Inst/tree/main> | [Colab](#) | [MSST-GUI](#) | [UVR instructions](#) (added in Download Center) | x-minus ([link](#) for premium users) | MVSEP

"The "e" stands for emphasis, indicating that this is a model that emphasizes fullness." "However, compared to v1, while the fullness score has increased, there is a possibility that noise has also increased." "lighter compared to v2." While SDR-wise it's [worse](#) than previous unwa's models, it has the best full [fullness](#) factor (you can read more about this new method of evaluation later in [this](#) section). The phase fixer doesn't really fix the noise in this model like in v1.

Like other unwa models, this can also confuse flute, trumpets and saxophone with vocals.
- You might want to use this max ensemble by dca100fb8 (e.g. the BS model here is capable of detecting flute correctly and the Mel - sax and trumpet):
unwa's v1e + Mel 2024.10 + BS 2024.08 (Max FFT; the latter models on MVSEP, also sometimes unwa's big beta5e can also retrieve missing instruments from v1e when those two fails)

- You might want to check max ensemble of instv1, instv2 and inst v1e - erdzo125 (for even better fullness but more noise - you can consider the [phase fix](#) for instv1)

- Anjok released a new beta Roformer [patch](#) #5 for UVR (Windows only):
[UVR_Patch_UVR_11_17_24_21_4_BETA_patch_roformer](#)
"- Fixed OpenCL compatibility issue with Roformer & MDX23C models.
- Fixed stem swap issue with Roformer instrumental models in Ensemble Mode."
The patch is rather not standalone like patch #3, so have a previous UVR installation.

- Anjok released a new beta Roformer patch #4 for UVR:
[UVR_Patch_11_17_24_21_4_BETA](#) (Windows: [full](#) | [patch](#) | Mac: [M1](#) | [x86-64](#))
Minor [bug fixes](#). Most importantly, MacOS version fix:
"Roformer checkbox now visible for unrecognized Roformer models" so now you can use custom Roformer models on MacOS Roformer patch without copying/modifying configuration files from Windows version or other users in order to circumvent the lack of option from Windows version to set that the recognized model is Roformer, so separation will work on that model. Plus it includes all the previous fixes in the previously released patch (so overlap code fixed, so no stem misalignment should occur on certain overlap settings - probably higher overlap now means longer separation)

- Anjok released a new beta Roformer patch #3 for UVR (Windows version for now) [UVR Patch 11 14 24 20 21 BETA patch roformer](#) - “this is a patch and requires an existing UVR installation” (so either [previous](#) beta Roformer patch or stable 5.6 version). The new patch fixes the issue with stem misalignment when using incorrect overlap setting for Roformers. Now it uses ZFTurbo code (also for MDX23C), meaning that probably now increasing overlap for Roformers will result in increasing separation times and potentially better SDR (the opposite of what it used to be in the previous beta Roformer patch). Potentially, it might allow using faster settings without stem misalignments or segment popping (when overlap and dim_t was set to 201 and overlap 2) for 4GB VRAM cards and some heavier models.

Among other minor fixes: “Roformer stem naming issues resolved. Fixed manual download link issues in the Download Center. Roformer models can now be downloaded without issue.”. Implementation of SCNet and Bandit archs is still in works.

[Full changelog](#).

- Bebruily made a Python script fixing phase with unwa v1 model, so it removes its noise.

[Download](#)

You need to run: pip install librosa
in case of “no module named librosa found” error.

“The results are almost, if not the same as x-minus' phase correction.

To use, you need to have the song separated with Kim's melband model and unwa's v1 model.” 32 bit output switch added

“the output length is few ms shorter than the input
the output has little popping in the end”

- SYH99999/yukunelatyh released a MelBandRoformerSYHFTV3Epsilon [model](#).

VS previous SYH's models “this version is more consistent with separation. It's not what I'd call a clean model; It sometimes lets background noise bleed into the vocal stem. But only somewhat, and depending on how you look at it, it can be a good thing since it makes the vocals sound less muddy.” Musicalman

Since then, there was also a newer [MelBandRoformerBigSYHFTV1Fast](#) model released.

- Lew released a v2 of the vocal enhancer model for Apollo trained on Roformer vocal outputs

Added for paid users on x-minus in the Ensemble menu or in the Restoration menu (formerly De-noise) and on [Colab](#). Model [files](#) | [config](#).

Works the best potentially on BS and Mel Roformer ensemble, but it might add some noise as well.

The model stopped progressing during training, so probably there won't be any newer epoch of this model.

- Unwa released v2 version of the inst Mel-Roformer model.
“Sounds very similar to v1 but has less noise, pretty good”
“the aforementioned noise from the V1 is less noticeable to none at all, depending on the track”.

“V2 is more muddy than V1 (on some songs), but less muddy than the Kim model.

(...) [As for V1,] sometimes it's better at high frequencies” Aufr33

Also, SDR got a bit bigger (16.845 vs 16.595)

<https://huggingface.co/pcunwa/Mel-Band-Roformer-Inst/tree/main> | [Colab](#) | [MSST-GUI](#)

“It's the same size as the big model with depth 12 and mask_estimator_depth 3.

The improvement was stagnant with the same model size as v1.” - unwa

- The model has been added to UVR [Beta Roformer](#) Download Center and x-minus.

- MSST-GUI is now included in ZFTurbo's repo, it's the "gui-wx.py" file" just don't run it by double-clicking, but run it from CMD.

GPU acceleration working only Nvidia GPUs will give out of memory errors on 4GB VRAM GPUs for Roformers (you can use CPU instead).

“UnicodeEncodeError” means there is disallowed character in your input file name, e.g. “doesn't work with [and] in the foldername - known bug”.

- Both duality models and inst v1/2 are now added to UVR [Beta Roformer](#) Download Center (problems with duality models in UVR have been fixed)

- Unwa released v2 version of the duality model, slightly a bit better SDR and fewer residues (available in the link below)

“[other](#)” is output from model

“[Instrumental](#)” is inverted vocals against input audio.

The latter has lower SDR and more holes in the spectrum.

So using MSST-GUI, leave the checkbox “extract instrumental” disabled for duality models.

- Unwa released a new inst-voc Mel-Roformer called “duality”, focused on both instrumental and vocal stem.

<https://huggingface.co/pcunwa/Mel-Band-Roformer-InstVoc-Duality/tree/main> | [Colab](#)

Vocals sound similar to beta 4 model, instrumentals are deprived of the noise present in inst v1 model, but as a downside, they don't sound similarly muddy to previous Roformers.

You can use it in the [MSST-GUI](#) for ZFTurbo script (already added) or with the OG ZF repo code. The model will now work in UVR (added in Download Center, but the problem was also fixed by Anjok and added in the OG repo's yaml)

- New Ensemble button added on x-minus for premium users for the new inst unwa's model. It corrects the phase and almost removes the noise existing in this model

“This post-processing uses Kim's model. After post-processing, the vocals will be replaced with those of this model.” [Examples](#)

Using Mel-Roformer de-noise might be better alternative:

"removes more noise from the song, keeping overall instrument quality more than the new button" koseidon72. But the more aggressive variant of the model sometimes deletes parts of the mix, like snares.

- New Bas Curtiz fine-tuned on MVSEP and unwa's inst Mel-Band added on MVSEP and x-minus.

Although there were only 5 submission sent to ZFTurbo for fine-tuning, and 30+ is needed, so there is not so much of a difference in the new FT.

"I suggest to all of you, if there is any voice left [in inst v1], use the Mel-Roformer de-noise with minimal aggression. "not only for little voices left, but also for some background noise. Unfortunately, this new [unwa's] model doesn't eliminate vocoder voices well from an instrumental"

The model is much faster than beta 4.

- unwa released a new Mel-Roformer model focused on instrumental stem this time (a.k.a. v1):

<https://huggingface.co/pcunwa/Mel-Band-Roformer-Inst/tree/main> | [Colab](#) | UVR [instructions](#) | MVSEP

"much less muddy (...) but carries the exact same UVR noise from the [MDX-Net v2] models"

But it's a different type of noise, so aufr33 denoiser won't work on it.

"you can "remove" [the] noise with UVR-Denoise, aggr. -10 or 0" although at least with -10 it will make it sound more muddy like Kim model and synths and bass are sometimes removed with the denoiser (~beacruly). UVR-Denoise-Lite doesn't seem to damage instruments that badly, but still more than Mel denoise (recommended aggr. - 4, with 272 vs 512 windows size it's less muddy, TTA can stress the noise more, somewhere above 10 aggr. it gets too muddy). UVR-Denoise on x-minus is even less aggressive (it's medium aggression model for free users without aggression pick), but it might catch ends of some instruments like bass occasionally. Premium minimum aggression model is somehow more muddy, but doesn't damage instruments. Minus the noise, this is a **groundbreaking** instrumental model among public models or existing Roformers.

(more [training details](#))

"Flipping the target seems to definitely have effect on the instrumental part!" Bas Curtiz

"I got an error when I set num_stems to 2." unwa

You can use "target_instrument: null" instead, which is also required for multistem training like on [this example](#) ~jarredou

"It's because of the PHASE. I found a way to fix it. Today I will add a new ensemble button."

- Similarity / Phantom Center Extractor [model](#) by wesleyr36 added on MVSEP (Experimental section) and x-minus.pro (Extract backing vocals).

"This model is similar to the Center Channel Extractor effect in Adobe Audition or Center Extract in iZotope RX [and Audacity/Bertom], but works better."

Although it does not isolate vocals, it can be useful." Aufr33
You can find more on the topic in [Similarity Extractor](#) section.

- ZFTurbo released MVSEP Wind model on the site (MelBand/SCNet/ensemble)
Some songs might be separated better vs the model on x-minus, not all.
- A GUI for ZFTurbo's Music Source Separation script for inference called MSST-GUI was released by Bas Curtiz (link with instruction in the description):
<https://www.youtube.com/watch?v=M8JKFeN7HfU> (reupload)
It has screen reader compatibility, although people can't navigate with the arrow keys in the web view for now, but at least you have HTML source of the page so you can just download models from there.
 - Multiple updates were made since that excerpt was written and new models were constantly added
 - If you have "ERROR: Could not build wheels for diffq, pesq, which is required to install pyproject.toml-based projects" then
"Edit the requirements.txt file and remove or comment out that line with asteroid" [click](#)
Then rerun pip install -r requirements.txt
 - If you have decent Nvidia GPU, and no GPU acceleration maybe "Check these commands to install torch version that handle cuda":
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
or
pip install torch==2.3.0+cu118 torchvision torchaudio —extra-index-url
https://download.pytorch.org/whl/cu118
or
pip install torch==2.3.0+cu118 --extra-index-url https://download.pytorch.org/whl/cu118
or
pip install torchaudio==2.3.0+cu118 --extra-index-url https://download.pytorch.org/whl/cu118
- New beta 4 of unwa's Mel-Roformer fine tune of Kim's voc/inst model released:
<https://huggingface.co/pcunwa/Mel-Band-Roformer-big/tree/main> | [Colab](#)
Be aware that the yaml config has changed, and you need to download the new beta4 yaml.
"Metrics on my test dataset have improved over beta3, but are probably not accurate due to the small test dataset. (...) The high frequencies of vocals are now extracted more aggressively. However, leakage may have increased." - unwa
"one of the best at isolating most vocals with very little vocal bleed and still doesn't sound muddy" "gives fuller vocals". Can be a better choice on its own than some ensembles.
- ZFTurbo, the owner of MVSEP, seeks help on improving Bas Curtiz' ft Mel-Roformer model on MVSEP. How can you help?
 - 1) Find a badly separated song with this model (e.g. bleeding)
 - 2) Find other model which separates your song correctly
 - 3) Send the good results (instrumental stem + vocal stem in stereo/44kHz with the same length) to ZFTurbo on Discord (or [email](#))

The result will be used as input for training a new, fine-tuned model.

- "1) All [MVSEP] ensembles now use Bas Curtiz MelRoformer model. SDR for Multi dataset stayed almost the same, but greatly increased for Synth
- 2) Drums model were updated for all Ensembles too.

https://mvsep.com/quality_checker/entry/7197" ZFTurbo

- New SCNet Large Drums model added on MVSEP

- <https://studio.gaudiolab.io> introduced new Noise Reduction feature

- For those having problems with too slow functioning of lucida.to, you can use <https://mp3-daddy.com/>. Sometimes FLAC option might not work, then download mp3 first, and then FLAC will work (the files have full 22kHz spectrum). Although, sometimes it may fail anyway (not always in incognito mode and with third party cookies allowed, and after long wait after error appeared). Downloading might be possible by manual download with Inspect option in your browser (it starts downloading and interrupts like on GSEP in the old days). Don't even bother reading their site description - it's full of AI-written sh&t. Contrary to what they say, it doesn't support YT or YT Music/Tidal/Deezer links, so you need to use their search engine. So probably the max output quality is 44kHz/16 bit. It doesn't seem to use Tidal (maybe Deezer).

<https://doubledouble.top/> is now also back online and supports Apple Music unlike Lucida, but it might be slower and go offline eventually as before.

- Strings model based on MDX23C arch added on MVSEP. It has low SDR yet (3.84), so it's hit or miss whether it will work for your song, but some people had even good results at times. ZFTurbo plans to work on it further.

- Finally, HQ_4 released in March has been added also on MVSEP (it was also added on x-minus/uvronline.app not long ago via [this](#) link at least)

- Beta 3 of the unwa's Mel-Roformer fine-tuned Kim's model released. Fine-tuning was started from scratch on enhanced dataset made with help of Bas Curtiz. As the result, the model is free from the high frequency ringing present in the previous beta models.
"I've added hundreds of GB worth of data to my dataset".

[Download](#) | [Colab](#)

"definitely better than Kim's now" although vocal residues might occur yet, and then use unwa's BS-Roformer fine-tune instead. SDR is slightly lower than Kim Mel-Roformer. It's good for RVC.

- The Lew's model was added to jarredou's [Colab](#)

- Lew released a model for Apollo, serving to enhance vocal results of Roformers
<https://ufile.io/09560o34>

"You can use it in Music Source Separation Training [repo](#), and it should be compatible with jarredou Apollo Colab" Links a bit below (not compatible with UVR).

- Beta 2 of the unwa's Mel-Roformer fine-tuned model released ([Colab](#)).
Be aware that both models have some ringing issues in higher frequencies. Hard to say if it will be fixed in the further training, Unwa explaining said it was mainly made with vocals in mind so it's not sure.
- Unwa released beta version of still-in-training Mel-Roformer fine-tuned model of Kim's. Not tested SDR-wise, but might give better results than the old Unwa's BS-Roformer model already. Download:
<https://huggingface.co/pcunwa/Mel-Band-Roformer-big/tree/main>

In UVR consider using dim_t = 1501 at the bottom of the yaml (can be slow), but 1333 or 1301 can be better for e.g. 40 second snippets, while the biggest SDR is for 1101 for all Roformers, but it still depends on a song what gives the best results (in reality, even SDR for each song is different, and bigger SDR not always means better quality, the quality using specific parameters might even differ in certain fragments).

- (uvronline.app/x-minus) New electric and acoustic guitar models by viperx' added on the site for premium users.
Acoustic seems to be good, while electric might be more problematic at times.
- Now lalal.ai have some voc/inst models sounding like some ensemble of public Roformers, but still not as good, although close. Some of their specific models are worth trying out, e.g. lead guitars - the model got better by the time, or also piano model

- <https://github.com/JusperLee/Apollo> | jarredou [Colab](#)
jarredou: "New tool for heavily compressed mp3 restoration, using bandsplitting and roformers. It does work really great if the audio was compressed at 44.1khz sample rate, whatever bitrate [<=128kbps]. BUT if there was some resampling leading to hard cutoff, it will wrongly behave."

The current model of Apollo was only trained on mp3 compressed audio. If you use ogg/opus/m4a/whatever else compressed audio as input, it's not guaranteed that it will work as expected."

It was also added on MVSEP as "Apollo MP3 Enhancer":

Demo: <https://mvsep.com/result/20240919224117-f0bb276157-mixture.wav>

Advice

"[Good use case](#):

Input has no hard cutoff (quality slowly degrades toward high freq).
Generated output is as expected. It can fill holes, and it can remove artifacts (and probably bleeding too) and is working great with highly degraded audio here. If trained on clean

source vs separated stem, which is not as much degraded content than 32kbps mp3 like previous example, I think it could work really great” [\[bad use case\]](#)

“so far the overlap magic is needed, cause u hear the transition”

“It seems to alter the tempo. It’s not a constant alteration, it just shifts stuff, and you can’t invert” becruily

> “I’ve seen this too in my tests, but it seems to happen only at the end of the chunk.

In the updated version (in which the end of chunks is ditched), I haven’t seen that issue again.

> Overlap feature added [to the Colab].

New inference.py created for easier local CLI use.

I have set chunk_size at 3 seconds as default in the Colab because it was the chunk_size used to train the model, but it seems that the highest is the best.” jarredou

It was also added into ZFTurbo’s training dataset (read [more](#)).

Also, for non-mp3 input files, you might want to experiment with compressing them to 64kbps first.

- Also, TS-BSmamba2 was added to the repo. So it’s available for training now too. But currently it works only on Linux.

- Aufr33 added MDX HQ4 to x-minus/uvronline via this link:

<https://uvronline.app/ai?hp&test-mdx>

- (x-minus/uvronline.app) “viperx has updated the piano model!

I just replaced it on the website.” Aufr33

“The new piano model is incredible, I have even been able to separate a harpsichord by passing it over and over again through the model until the other instruments are left alone and it doesn’t sound bad at all.”

There was some update to MVSEP piano models lately too, and there are SCNet and viperx models and ensemble with metrics added on the website (at least on separate page beside multi song dataset chart).

“both similar but mvsep has a teeny bit more bleed during the choruses and whatnot”

- (MVSEP) “I added possibility to use Bas Curtiz’ MelRoformer model with large score on Synth dataset. You must choose it from MelRoformer options. By default, my model is used. The problem with Bas’s model is that it’s very heavy and slow, with almost the same score on Multi dataset.” Aufr33

“I’ve tried some songs and have great result! Music sounds fuller than original Kim’s one & the finetuned version from ZFTurbo. Even [though] the SDR is smaller than BS Roformer finetuned last version, but almost song has the best result in instrumental.

1 song I found is bad result is from Wham - Where did your hearts go. The trumpet or sax whatever sound was lost, the model detects it as vocal, and the 1st beginning of vocal still heard. On other mel roformer, that trumpet or sax sound can still separate it as well." Henry

- (MVSEP) "Guitar model was updated. I added BSRoformer model by viperx with SDR: 7.16. And

- I replaced [guitar] Ensemble. Earlier it was MDX23C + Mel. Now its BS + Mel. SDR increased from 7.18 to 7.51.

Demo: <https://mvsep.com/result/20240914110542-7ab0356600-song-000-mixture.wav>

All these models are available for all users." ZFTurbo

- The new MDX HQ5 beta model is now online!

Use this link to access it:

<https://uvronline.app/ai?hp&test-mdx> - link for premium users

Go to "music and vocals" and there you will see it (scroll down).

It's not a final model yet, and the model is in training from April and is still in progress.

It seems to be muddier than HQ_4 (and more than Kim's and MVSEP's Mel-Roformer), it has less vocal bleeding than before, but more than Kim Mel-Roformer. Sometimes struggles with reverb.

"Almost perfectly placed all the guitar in the vocal stem" it might get potentially fixed in the final version of the model, which is planned for release in the mid-November as Anjok said at 04.11.24.

The model is not available in UVR yet (only on uvronline.app)

- Using the [UVR Roformer](#) beta patch for Mac doesn't allow you to choose the Roformer parameter to check for manually copied Roformer models to UVR like: Kim Mel-Roformer or unwa's Roformer, and only config name can be chosen, but no confirm button is available to make the model work. Place [corresponding](#) hash-named file to models\MDX_Net_Models\model_data after placing model file to MDX_Net_Models and non-hased model's yaml to mdx_c_configs and start the UVR.

- Aufr33 released files for the new UVR de-reverb model made with jarredou (based on VR 5.1 arch).

1. Download [this](#) and unzip into your Ultimate Vocal Remover folder

2. Select VR architecture and DeReverb model from the menu

3. Set the parameters as shown [here](#)"

(PS: Dry, Bal: 0, VR 5.1, Out:32/128/Param: 4band_v4_ms_fulband -

An already existing json config file in modelparams folder has the same checksum)

Bas Curtiz' "Conclusion so far:

- MDX[23C] De-Reverb seems to be cleaner, takes the reverb away, also between the words,

whereas VR leaves a little reverb

- [The new] VR De-Reverb seems to sound more natural, maybe therefore actually.
Also, MDX tends to 'pinch' some stuff away to the background, which sounds unnatural.

This is just based on my experience with 3 songs/comparisons, but both points are a pattern.
Overall, they're both great when u compare them against the original reverbed/untouched
vocals." [Video](#)

- SCNet Large vocal model on MVSep published.

Multisong dataset:

SDR vocals: 10.74

SDR other: 17.05

"just like the new bs roformer ft model, but with more bleed. [BS] catches vocals with more harmonies/bgv" isling

- Cyrus repaired pip issues with [Medley Vox](#) Colab

- Aufr33 released MDX23C de-reverb model files

https://a19p.uvronline.app/public/dereverb_mdx23c_sdr_6.9096ckpt | [config](#)

"If you will use this model in your project, please credit us (me and jarredou)"

Also added on MVSEP.

UVR instruction:

1. Just copy model to Ultimate Vocal Remover\models\MDX_Net_Models

2. Copy .yaml config to Ultimate Vocal

Remover\models\MDX_Net_Models\model_data\mdx_c_configs

3. When opening UVR, selecting dereverb_mdx23c_sdr_6.9096 from the MDX-Net process method, don't click 'RoFormer model' cause it's not.

4. Select config_dereverb_mdx23c from the dropdown. Done." ~Bas Curtiz

5*. In case of "no key" error in UVR, changed line 30 in the config to:

No dry

But it doesn't happen to everyone.

- New UVR Dereverb model added on uvronline.app for premium users.

It seems to handle room reverb better than the previous MDX23C model, and the Foxy's model sometimes cut "way too much" than this new model.

- People cannot separate using Ripple since longer than August 12th. There's an error "couldn't complete processing please try again"

- (x-minus.pro/uvronline.app) "Hipolink was a temporary solution. Now I can accept payment via Patreon as well." Aufr33

- MDX23C De-reverb model by Aufr33 released for premium users of uvronline.app.
"Thanks to jarredou for helping me create the dataset"

- Jarredou released v. 2.5 of MDX23 Colab adding the new Kim Mel-Roformer model. Final SDR is higher (17.64 vs 17.41 for instrumentals, with 2024.08.15 [MVSEP](#) Ensemble being 17.81).

https://colab.research.google.com/github/jarredou/MVSEP-MDX23-Colab_v2/blob/v2.5/MVSEP-MDX23-Colab.ipynb

"Baseline ensemble is made with Kim Melband rofo, InstVocHQ and selected 1296 or 1297 BS Rofo" switching from 1296 to 1297 produces more muddy/worse instrumentals in this Colab (more sudden jumps of dynamics from residues)." VitLarge is no longer used by default.

"I've opened a donation account for those who would want to support me:

<https://ko-fi.com/jarredou>"

- unwa's fine-tuned BS-Roformer model released (12.59 for instr) - worse SDR than other fine-tuned models on MVSEP by ZFTurbo, but better SDR than Kim's MelRoformer and viperx base model

https://drive.google.com/file/d/1Q_M9rIEjYIBZbG2qHScvp4Sa0zfdP9TL/view

- Mel-RoFormer Karaoke / Lead vocal isolation model files released by Aufr33 and viperx
"If you will use this model in your project, please credit us" ([download](#))

[UVR instructions](#). Be aware that online version on uvronline/x-minus seems to work better.

- doubledouble.top will be soon replaced by <https://lucida.to/>

- Kimberley Jensen released her Mel-Band Roformer vocal model publicly ([download](#))
(simple [Colab/CML inference/x-minus/MVSEP/jarredou Colab](#) too now)
Works in UVR [beta Roformer](#) ([model](#) | [config](#) - place the model file to
models\MDX_Net_Models and config to model_data\mdx_c_configs subfolder and "when it
will ask you for the unrecognised model when you run it for the first time, you'll get some box
that you'll need to tick "roformer model" and choose it's yaml".
Use overlap 2 for best SDR, or 8 for faster inference in UVR)

- SCNet model published on MVSEP. Similar metrics to MDX23C model, but seems to leave lot of vocal residues.

" it is based on SCNet-small config from the paper, the SCNet-large config is almost 1 SDR above in the reported eval, so hopefully, next SCNet model trained by ZFTurbo with that large config will be better too." So far he had some problems training on large config, sadly.

- Slightly better Roformer 2024.08 model (0.1 SDR+) was added on MVSEP vs 2024.04 model “it seems to be much better at taking the vocals when there are a lot of vocal harmonies.”
- (x-minus.pro/uvronline.app) “In the new interface, the BS-RoFormer model now also has De-mudder
Select the Music and vocals, BS-RoFormer and after processing you will see the De-mudder button appear.” Aufr33
It works for premium.
- If you got an error while using jarredou’s Drumsep Colab (*object is not subscriptable*):
change to this on line 144 in inference.py:

```
if type(args.device_ids) != int:
    model = nn.DataParallel(model, device_ids = args.device_ids)
```

(thx DJ NUO)
- (GSEP) I received email about deletion of my files on one of my accounts which is inactive, if I don’t buy premium (haven’t received it on my main account with premium), so it’s probably due to inactivity and no premium. It’s probably for accounts not using the service since the release of the new paid site and/or maybe didn’t have premium since then (email is from July 9th, so 3 months after the release of the new site, so possibly your files can get deleted after 3 months after premium was disabled on your account). Normally, new separations for free users are deleted after 3 days now, but older files were preserved at least for accounts using beta till now. The account wasn’t used since the end of October 2022.
Check your mailbox to ensure, I didn’t find that mail in spam on the main account with premium, so hopefully it’s not for everyone (at least not for those with premium or who used the site since the last 3 months):
“All files from Gaudio Studio will be deleted on August 7, 2024 [Wednesday]. (...) If you purchase a Studio Plan, your files will be preserved.”
Be aware that they function in Japan, which is GMT+9, so it’s 6 hours sooner than CEST (Warsaw, Skopje, Zagreb).
~~If you currently have premium, you can download all your previous separations in WAV without any charge (at least that’s how it used to be), without premium it says (misleadingly, I assume) “The song processed in the beta service do not support WAV file downloads.” but probably you’ll be able to do that if you buy premium if nothing has changed. It’s no longer possible, and there are no references to WAV in dev tools as before.~~
- Aufr33 released files of his Mel-Roformer de-noise models publicly:
[Less aggressive](#) & [More aggressive](#) | [yaml file](#)
“If you will use this model in your project, please credit me”
Added in jarredou [Colab](#) too (and on x-minus.pro/uvronline.app for premium users and MVSEP).

Both models work in UVR too (don't forget setting overlap to 2 to avoid stem misalignment issues like for other Roformers in UVR Roformer beta, overlap 3 or above will break separation)

- Jarredou released manual ensemble Colab with drop-down menus (based on ZFTurbo code)

https://colab.research.google.com/github/jarredou/Music-Source-Separation-Training-Colab-Inference/blob/main/Manual_English_Colab.ipynb

- To fix issues with BS variant of anview's de-reverb model in UVR "change stft_hop_length: 512 to stft_hop_length: 441 so it matches the hop_length above" in the yaml file. It doesn't happen on (thx lew).

If that line is not present in your model config go to the settings, then choose MDX In the advanced menu, then click the "clear auto-set cache" button.

Then go back to the main settings, click "reset all settings to default" and restart the app (thx santilli_).

- If you still have error on every attempt of using GPU Conversion in UVR on AMD GPU (you might potentially use outdated drivers and/or Windows), go to Ultimate Vocal Remover\torch_directml and replace DirectML.dll from C:\Windows\System32\AMD\ANR (make backup before). Experimentally, you can use this older [1.9.1.0](#) version of the library. Restart UVR after replacing the file!

Be aware that results achieved without GPU Conversion that way, at least on certain configurations, might have noisy static instead of bleeding in less noisy parts of stems vs when using only CPU (basically, MDX noise can be somehow different on GPU and denoise standard only alleviates the issue to some extent, and you need to use Denoise Model option to get rid of this noise, or better solution - min spec manual ensemble of denoise disabled result and denoise model to get rid of more noise. Aufr's Mel-Roformer minimum denoise works worse for it.

- GSEP introduced a new model called "Vocal Remover" dedicated for vocal extraction and is only used for vocals, instrumental stem still uses the old model. Might be good at extracting SFX as well. (becruily/wancite)

- ([uvronline.app](#)) Mel-Roformer De-noise released for premium users.

"This model is optimized for music and vocals. You can choose between two aggressiveness settings:

minimum - removes fewer effects such as thunder rolls

average - usually removes more noise"

"The new model works as good as my UVR De-noise model, or even better."

- [drumsep](#) model by aufr33 and jarredou added on [MVSEP](#) and [uvronline.app](#) too

- Not Eddy's multi-arch Colab released in form of UI (like in e.g. KaraFan)
https://colab.research.google.com/github/Eddycrack864/UVR5-UI/blob/main/UVR_UI.ipynb

In case of “FileNotFoundError: [Errno 2]” try other location than “input”, or other Google account in case of ERROR - mdxc_separator (helps for both).

- New Mel-Roformer de-reverb model by anvew was released
<https://github.com/ZFTurbo/Music-Source-Separation-Training/issues/1#issuecomment-2226805511>
(to make it work with UVR, delete “linear_transformer_depth: 0” from the YAML file, copy the model to MDX_Net_Models and YAML config to model_data\mdx_c_configs)
Also added on MVSEP.

“I’m definitely hanging onto it. It reminds me of the equivalent dereverb mdx model, which I’ve always liked (when it works). The roformer model is cleaner in some ways, though slightly more filtered and aggressive.

Neither the roformer or mdx models respond to mono reverb. However, adding a stereo reverb on top solves that, especially with roformer.” (Musicalman)

“anvew’s models can remove reverb effect only from vocals. Old FoxJoy’s model works with full track.”

- BS-Roformer -||- - a bit better SDR
<https://github.com/ZFTurbo/Music-Source-Separation-Training/issues/1#issuecomment-2229279531>
(To fix “The size of tensor a”... error with BS variant of anvew’s de-reverb model “change stft_hop_length: 512 to stft_hop_length: 441 so it matches the hop_length above” in the yaml file.) thx lew

Added in the Colab:

[https://colab.research.google.com/github/jarredou/Music-Source-Separation-Training-Colab-Inference/blob/main/Music%20Source%20Separation%20Training%20\(Colab%20Inference\).ipynb](https://colab.research.google.com/github/jarredou/Music-Source-Separation-Training-Colab-Inference/blob/main/Music%20Source%20Separation%20Training%20(Colab%20Inference).ipynb)

- The below model has been added. Ensembles updated as well.
Some users report “bleed from some synths and bass guitar” “Some drums instruments are low volume on drums only. While mel roformer makes a good clean one” “On some parts it’s almost like it doesn’t separate anything for a few seconds and on some other parts, it’s working just really great. The demucs one is way more stable when listening to individual model separations on the same song.” (or simply older ensemble)

- (MVSEP) I finished my drums models. Results:

MelRoformer SDR: 12.76

Demucs4 (finetuned) SDR: 12.04

Ensemble Mel + Demucs4 SDR: 13.05

for comparison:

Old Best Demucs4 SDR: 11.41

Old Best Ensemble SDR: 11.99

New models will be added on site soon.” ZFTurbo

For comparison, the Mel-Roformer available on x-minus trained by viperx has 12.5375 SDR.

- (for models trainers) “Official SCNet repo has been updated by the author with training code: <https://github.com/starrytong/SCNet>”

“ZF’s script already can train SCNet, but currently it doesn’t give good results”

<https://github.com/ZFTurbo/Music-Source-Separation-Training/releases/>

The author’s checkpoint:

<https://drive.google.com/file/d/1CdElIqsoRfHn1SJ7rccPfyYioW3BIXcW/view>

“One diff I see between author config and ZF’s one, is that dev has used learning rate of 5e-04 while it’s 4e-05 in ZF config. And main issue ZF was facing was slow progress (while author said it worked as expected using ZF training script

<https://github.com/starrytong/SCNet/issues/1#issuecomment-2063025663>)”

The author:

“All our experiments are conducted on 8 Nvidia V100 GPUs.

When training solely on the MUSDB18-HQ dataset, the model is trained for 130 epochs with the Adam [22] optimizer with an initial learning rate of 5e-4 and batch size of 4 for each GPU. Nevertheless, we adjust the learning rate to 3e-4 when introducing additional data to mitigate potential gradient explosion.”

“Q: So that mean that you have to modulate the learning rate depending on the size of the dataset?

I think it’s the first time I read something in that way.

A: Yea, I suppose because the dataset is larger you need to ensure the model sees the whole distribution instead of just learning the first couple of batches”

- jarredou/frazer

SCNet paper: <https://arxiv.org/abs/2401.13276>

On the same dataset (MUSDB18-HQ), it performs a lot better than Demucs 4 (Demucs HT).

“Melband is still SOTA cause if you increase the feature dimensions and blocks it gets better you can’t scale up scnet cause it isn’t a transformer. It’s a good cheap alt version tho”

Still, it might potentially give interesting results when training will be mastered to the point when e.g. SDR will be in pair with at least MDX-Net models as they can still be better than Roformers for instrumentals in many cases (e.g. MDX-Net tend to have less muddy

instrumentals - every arch can have its own unique sound characteristics and might be potentially useful for ensembling).

- (jarredou) "I've released the Drums Separation model trained by aufr33 (on my not-that-clean drums dataset).

Stems: kick, snare, toms, hihat, ride, crash

It can already be used, but training is not fully finished yet.

The config allows training on not so big GPUs [n_fft 2048 instead of 8096], it's open to anyone to resume/fine-tune it.

For now, it's struggling a bit to differentiate ride/hh/crash correctly, kick/snare/toms are more clean.

Download

[attached config includes also necessary training parameters for training further using ZFTurbo [repo](#)]:

https://github.com/jarredou/models/releases/tag/aufr33-jarredou_MDX23C_DrumSep_model_v0.1

Use on Colab:

[https://colab.research.google.com/github/jarredou/Music-Source-Separation-Training-Colab-Inference/blob/main/Music_Source_Separation_Training_\(Colab_Inference\).ipynb](https://colab.research.google.com/github/jarredou/Music-Source-Separation-Training-Colab-Inference/blob/main/Music_Source_Separation_Training_(Colab_Inference).ipynb)

It works in UVR too. All models should be located in the following folder:

Ultimate Vocal Remover\models\MDX_Net_Models

Don't forget about copying the config file to: model_data\mdx_c_configs

The model achieved much better SDR on small private jarredou's evaluation dataset compared to the previous drumsep model by Inagoy which was based on a worse dataset and older Demucs 3 arch.

The dataset for further training is available in the drums section of [Repository of stems/multitracks](#) - you can potentially clean it further and/or expand the dataset so the results might be better after resuming the training from checkpoint. Using the current dataset, the SDR might stall for quite some amount of epochs or even decrease, but it usually increases later, so potentially training it further to 300-500-1000 epochs might be beneficial.

"I've had models where SDR changes by 0.01 but fullness/bleedless change with 10-15 points, I wouldn't trust it that much" - becruily

Current model metrics:

"Instr SDR kick: 18.4312
Instr SDR snare: 13.6083
Instr SDR toms: 13.2693
Instr SDR hh: 6.6887
Instr SDR ride: 5.3227
Instr SDR crash: 7.5152
SDR Avg: 10.8059" Aufr33

And if evaluation dataset hasn't changed since then, the old Drumsep SDR:

"kick : 13.9216
snare : 8.2344
toms : 5.4471
(I can't compare cymbals score as it's different stem types)" - jarredou

After initial jarredou's training in Colab, Aufr33 decided to train the model for additional 7 days, to at least above epoch 113 (perhaps around 150, it wasn't said precisely), while using the same config, but on a faster GPU (2x 4090).

Even epoch 5 trained on jarredou's dataset casually in free Colab (which uses Tesla T4 15GB with performance of RTX 3050, but with more VRAM) with multiple Colab accounts and very light and fast training settings, already achieved better SDR than Drumsep:

"epoch 5:

Instr SDR kick: 13.9763
Instr SDR snare: 8.4376
Instr SDR toms: 6.7399
Instr SDR hh: 0.7277
Instr SDR ride: 0.8014
Instr SDR crash: 4.4053
SDR Avg: 5.8480

epoch 15:

Instr SDR kick: 15.3523
Instr SDR snare: 10.8604
Instr SDR toms: 10.3834
Instr SDR hh: 4.0184
Instr SDR ride: 2.7248
Instr SDR crash: 6.1663
SDR Avg: 8.2509"

Don't forget to use already well separated drums (e.g. from Mel-Roformer for premium users on x-minus) from well separated instrumental as input for that model, or Jarredou MDX23 Colab fork v. 2.4 or MVSEP 4/+ ensemble (premium).

Purely for drums separation from even instrumentals, the model might not give good results. It was trained just on percussion sounds and not vocals or anything else.

Also, e.g. the kick and toms might have a bit weird looking spectrograms. It's due to: "mdx23c subbands splitting + unfinished training, these artifacts are [normally] reduced/removed along [further] training." [Examples](#)

- BTW, just for inference (separation), "ONNX and Demucs models don't work with multi-GPU"
- In the "experimental" section of MVSEP, there's been added a new multispeaker model at the bottom.
E.g. it can work well splitting rapping and singing overlapped in the same, previously well separated vocal stem, but:
"It works more or less ok on my validation [5 quite different "songs"], but it's a disaster on real data. I opened it for everyone, but don't expect really good results" ZFTurbo
- Also, there has been added a new multichannel section in "experimental" it's just for songs with 3 or more audio channels like e.g. Dolby Atmos (FLAC/WAV input supported). It's just BS-Roformer and there's "no reason to process stereo tracks with it". Also, the original sample rate of the input file is preserved here.
- One of MVSEP's GPU died recently, so the separations will be probably slower than usual.
- jarredou updated his [AudioSR Colab](#). Now "each processed chunk is normalised at same LUFS level (fixes the volume drop issue)" plus "input audio is resampled accordingly to 'input_cutoff' (instead of lowpass filtering)"
Now also some errors associated with mono files are fixed.
- New drums model available on x-minus.pro
"SDR is: 12.4066.
Thanks to @viperx for model training! The model is trained on 995 songs. A small number of my pairs were included in the dataset." Aufr33
Very positive reviews so far.
- New guitar model added on MVSEP
"Previous old model mdx23c: 4.87
New mdx23c model: 6.34
New MelRoformer model: 6.91
Ensemble MDX23c + MelRoformer: 7.10
Extract vocals and after apply ensemble MDX23C + MelRoformer: 7.28"

- If x-minus.pro site doesn't work for you, use the clone instead:

<https://uvronline.app/ai?hp>

- Demudder on x-minus was updated on 13.06 (cosmetic differences)

- "Some interesting updates to SL 11

<https://www.youtube.com/watch?v=2BoEgBGiafM>

Seemingly, separation features got better. Coming on 19th June.

Their new algo was [evaluated](#), and SDR is a bit worse than [htdemucs](#) 4 stem non ft model.

Every stem has some bleed, vocals are decent, and actually have better SDR than Demucs_ft. GPU processing in options has low utilization and is slow, they say it's planned to be fixed in patch. 16GB VRAM recommended at least while using brass and saxophone. Around 18 models can be used in total.

Unmix Multiple Voices is for speech case, not for singing case.

Unmix Drums option can serve for further separation of drums

"the residual kick/snare problem is much better, but the cymbal split does still contain bleed from the rest of the song sadly" [vs drumsep] - jasper waffles

- [Multi-arch Colab by Not Eddy](#)

incorporates: MDX-Net, MDX23C, Roformers (incl. 1053), Demucs, and all VR models, YouTube support and batch separation. It uses broken overlap from OG beta UVR code. Use the one below for just Roformers and now also 1053 instead:

[Colab with Roformers](#)

- New Mel-Roformer De-Crowd model released on MVSEP. It slightly surpassed SDR of the previous MDX23C model.

It's also available publicly in the repository below:

<https://github.com/ZFTurbo/Music-Source-Separation-Training>

To use it in UVR, Go to UVR\models folder, and paste [that](#) there.

Then change "dim_t" value to 801 at the very bottom of:

"model_mel_band_roformer_crowd.yaml" in mdx_c_configs subfolder. Don't use overlap above 4.

- Drums Roformer model shared publicly by Yolkis

<https://github.com/ZFTurbo/Music-Source-Separation-Training/issues/1#issuecomment-2156069553>

Not totally bad results as for 7.68 SDR, but it was trained on subpar GPU for Roformers for only 5 days. To use it in UVR, delete linear_transformer_depth line in the config.

- (x-minus.pro) "The new Strings model by viperx has been added!" based on Mel-Roformer arch.

Good results reported so far.

Sometimes it can pick up brass.

- (x-minus.pro) “Demudder has been added!

This only works with the mel-roformer inst/vocal model. You need premium to use it.” It works only for instrumentals. Vocals are unaffected. It fills holes in the spectrum, basing on both vocals and instrumental stems (e.g. it won’t serve to just recover lossy mp3). The option shows after you uploaded/processed a track (at least with mel-roformer model). It’s capable of providing better results than max_mag a.k.a. BS and Mel Roformer ensemble (premium), depends on a song. SDR-wise, it’s not much worse than original model results (16.48 vs 17.32).

- “New VST for real time source separation (probably same models [like in] MPC stems)

<https://www.youtube.com/watch?v=0Js5bWQWY7M>

<https://products.zplane.de/products/peelstems/>”

- Mel-RoFormer de-crowd by aufr33 and viperx model files have been released publicly.

DL: <https://buzzheavier.com/f/GV6ppLupAAA>

Conf: <https://buzzheavier.com/f/GV6psmJpAAA>

“You can use ZFTurbo's code [check his GitHub] to run this model. If you use it in your project, please credit us”

To use it in UVR 5, change “the name of the model itself to the name of the YAML file.

This model only works the best when at 2 overlap, since anything higher than that it'll stop isolating parts of the song entirely.

Or else, you can also check out setting “inference.dim_t” parameter at the bottom of the yaml file to 801. “Leaving dim_t at 256 (2.5seconds) makes the model only usable with overlap=2 (2 seconds) with current beta code. Higher value will result in missing/non-processed parts.” - jarredou

“The Roformer model does a better job at retaining the instruments and vocals as well as some sound effects and synths better than the MDX-NET decrowd, but at the cost of crowd bleed. While the MDX-NET decrowd model does a better job at removing most of the crowd at the cost of instrumental bleed into the crowd stem.

Sometimes [the old model] mistakes the fuzzy sounds of guitar as crowd noise
Also isolates some kicks in songs” - Kashi

“For really difficult live songs (where the crowd is overwhelmingly loud to the point where you can't hear the band properly) sometimes filtering vocals with mel roformer on xminus THEN running the vocals stem through the mdx decrowd model sometimes helps” - isling

- (x-minus) “The new Wind / Saxophone model has been added! It completely replaces the old UVR model [on the site]. Thanks to viperx for model training.”

“Really great model! Big step since the last VR winds model.” It works better for brass instruments than wind.

- (x-minus) “BS-RoFormer Bass model added! This is a model by viperx.” Aufr33
“much better at treble-heavy bass tones than demucs”

It’s different from the latest MVSEP bass model. Viperx’ model might be cleaner, but pick up less bass at times. Both are improvement over demucs_ft ~drypaintdealerundr
It’s best in no piano stem.

- (x-minus) “Piano beta model added! Thanks to viperx for model training.”

- (x-minus) “Now all models except BVE are available for free, even without registration!

The only restrictions:

Only mp3 downloads are available

No ensemble

10 minutes of audio per day (past 24 hours). This is enough for testing 2-3 songs.

Max song duration is 8 minutes

It’s not available through Tor, and it’s not available in some countries.” Aufr33

Q: Why BVE models are excluded? [from free option]

A: “Because in the free version, wav files are deleted immediately after processing is complete. This makes it impossible to download some stems. In addition, this model necessarily uses MDX for preprocessing, which is very compute-intensive.”

- (mvsep) Bass model is online. The metrics:

Single models:

HTDemucs4 bass SDR: 12.5295

BSRoformer bass SDR: 12.4964

MelRoformer bass SDR: 11.86

MDX23C bass SDR: 11.20

Models on site:

HTDemucs4 + BSRoformer Ensemble (It’s available on site as MVSep Bass (bass, other)):

13.25

Ensemble 4 stems and All-In (from site): 13.34

For comparison:

Ripple lossless (bass): 13.49

Sami-ByteDance v1.0: 13.82

- GSEP announced works on a new model

- Mel-RoFormer Karaoke model added on [x-minus.pro](#)

“one of the cleanest lead vocals result[s]”

“I noticed that the new karaoke model considers vocals as lead vocals, even if they are quite wide. In other words, it has a much larger tolerance for vocal width than other karaoke models. This means that backing vocals that sound almost centered can be removed along with the lead vocal. If I apply a stereo expander, the model produces more adequate results. So when I add the Lead vocal panning setting, the "center" will actually work as "stereo -20%" (for example).”

“Q: wouldn't this mean that there will be more backing vocal bleed in the lead vocal stem too?

A: The model behaves differently. In some cases, it completely isolates vocals, in other cases it gets confused and vocals appear in both stems at once, in other cases it doesn't isolate at all.”

Q: What are the differences between mel-roformer karaoke and the last model?

A: “If the vocals don't contain harmonies, this model (Mel) is better. In other cases, it is better to use the MDX+UVR Chain ensemble for now.”

Although your mileage will still vary on a song, e.g. “For most of the songs I tried it worked very well. Example: “From Souvenirs to Souvenirs” by Demis Roussos. It's the only model as of now which can separate the lead from the back vocals correctly” dca100fb8

- Izotope RX11 officially released. Seeing by the unencrypted onnx model file names, it uses demucs_ft for stem separation now, but maybe it's not the same model as the public one, as all stems “null back to the input stereo which is something standard demucsht doesn't appear to do (...) mdx seems to always null luckily.” The feature still doesn't use CUDA/Nvidia GPU for processing (and there's no such option anywhere). It's still an improvement over RX8-10 as they used in-house Spleeter 22kHz models before.

[Comparison](#)

<https://www.youtube.com/watch?v=MhUEmvneerc>

New [features](#) (e.g. clean up dialogue in real time).

- Logic Pro 11 now incorporates a [Stem Splitter](#). Results vary from good to bad (and worse than known solutions) depending on a song.

- Mel-RoFormer De-crowd model added on x-minus.pro.
Results are more accurate than in the old MDX model.

- [GSEP](#) has been updated.

Free option for all stems has been removed. There's only a 20 minutes free trial. WAV is only for paid users.

Vocals and all other stems (including instrumentals/others) are paid, and length for each stem is taken from your account separately for each model.

No credit is not required for the trial.

For free, only mp3 output and 10 minutes input limit.

For paid users there's a 20 minutes limit, and mp3/wav output, plus paid users have faster queue, shareable links, and long term results storage.

Pricing

7\$/60 minutes

16\$/240 minutes

50\$/1200 minutes

Seems like there weren't many changes in the model (if there weren't even more vocal residues introduced since then). People still have similar complaints to it. [Comparison](#) video.

There was an average of 0.13 SDR increase for mp3 output and first 19 songs from multisong dataset evaluation, but judging by no audible difference for most people, they could simply change some parameters for inference.

The old files from previous separations on your account didn't get deleted so far.

- (x-minus) max_mag of (?)Roformer and Demucs (drums only) added

"now the synths and everything else feels muddy
noticed the drums in some places (mainly louder-ish bits) sound a bit weird
mostly lower end like bass drum instead of hi hats
great improvement overall" isling

- Doubledouble might have some occasional hiccups on downloading. If you encounter very slow download, don't attempt retrying the same download, but generate a new download query. Do it even three times in a row if necessary or wait half an hour and retry. Also, you can check the option to upload your result on external hosting.

- (x-minus) "Added max_mag ensemble for Mel-RoFormer model! It combines Mel and BS results, making the instrumentals even less muddy, while better preserving saxophone and other instruments."

- New Mel-Roformer model trained by Kimberley Jensen on Aufr33 dataset dropped exclusively on [x-minus](#).

"This model will now be used by default and in ensemble with MDX23C (avg)."
It's less muddy than viperx model, but can have more vocal residues e.g. in silent parts of instrumentals, and can be more problematic with wind instruments putting them in vocals, plus it might leave more instrumental residues in vocals.

“godsend for voice modulated in synth/electronic songs”

SDR is higher than viperx model (UVR/MVSEP) but lower than fine-tuned 04.24 model on MVSEP.

- New UVR patch has been released. It fixes using OpenCL on AMD and Intel GPUs (just make sure you have GPU processing turned on in the main window and (perhaps only in some cases) OpenCL turned on in the settings).

Plus, it fixes errors when the notification chimey in options is turned on.

https://github.com/TRvlvr/model_repo/releases/download/uvr_update_patches/UVR_Patch_4_14_24_18_7_BETA_full_Roformer.exe (be aware that you can lose your current UVR settings after the update)

To use BS-Roformer models, go to download center and download them in MDX-Net menu (probably temp solution).

For 4GB VRAM and at least AMD/Intel GPUs, you can try out segments 32, overlap 2 and dim_t 201 with num_o 2 (dim_t is at the bottom of e.g. model_bs_roformer_ep_368_sdr_12.9628.yaml) to avoid crashes.

You might want to check a new recommended ensemble:

1296+1297+MDX23C HQ

Instead of 1297 and for faster processing and similar result, make a manual ensemble with a copy of 1296 result instead. It might work in similar fashion like weighting in 2.4 Colab and model ensemble on MVSEP ([source](#)).

- VIP code allowing access to extra models in UVR currently doesn't work using [Roformer beta patch](#) older than #10, and MDX23C Inst Voc HQ 2 models disappeared from download center and GH. You can try to download VIP model files manually from this link and place them in Ultimate Vocal Remover\models\MDX_Net_Models directory:

https://github.com/deton24/Colab-for-new-MDX_UVR_models/releases/download/v1.0.0/UVR-MDX-NET_Main_406.onnx

https://github.com/deton24/Colab-for-new-MDX_UVR_models/releases/download/v1.0.0/UVR-MDX-NET_Main_427.onnx

https://github.com/deton24/Colab-for-new-MDX_UVR_models/releases/download/v1.0.0/MDX23C-8KFFT-InstVoc_HQ_2.ckpt

Of course, it's not all. E.g. 390 340 models and old beta MDX-Net v2 fullband inst models epochs are not reuploaded. This situation might cause errors on an attempt of using Inst Voc HQ 2 in AI Hub fork of Karafan.

Decrypted VIP repo leads to links which are offline, and also it doesn't contain all models. Possibly the only way to access all the VIP models in beta UVR, is to roll back to stable 5.6 version from UVR official repo, and after downloading all desired VIP models, update to the latest patch.

- According to their forum leak, iZotope RX11 might be released between May and July, and contain some “pretty big changes”, among others, a novel arch for separation is rumored, and a lot of options reworked. (cali_tay98)

Official announcement is out:

<https://www.izotope.com/en/learn/rx-11-coming-soon.html>

(overhauled repair assistant, real time dialogue isolation for better separation of noise and reverb from voice recording)

- GSEP announced an update on May 9th with a WAV download option and redesigned UI. The site will be unavailable on 8th May.

Noraebang (karaoke) service “due to low usage” will be shutdown, and your separated files deleted (you can make a backup of your files before).

Paid plan will be offered with faster processing times and “additional features”. No model changes are announced so far. The update schedule might change.

- MDX23-Colab Fork [v2.4](#) is out. Changes:

“BS-Roformer models from viperx added, MDX-InstHQ4 model added as optional, FLAC output, control input volume gain, filter vocals below 50Hz option, better chunking algo (no clicks), some code cleaning” - jarredou

- (x-minus) “Added mixing of MDX23C and BS-RoFormer results (avg/bs-roformer option). So far, it works only for MDX23C.” Aufr33

- “Output has released a free AI based generator that create multitrack stem packs

<https://coproducer.output.com/pack-generator>”

12-seconds long audio, fullband, 8 stems (drums in one stem, electric and rhythm guitar, hammond organ, trumpet, vocals) with 8 variations

“this looks more like it's mixing different real instruments, rather than actually making up songs (like a diffusion based generator)” ~jarredou/beacruly

- Ensemble on MVSEP updated

- The site is up and running after some outage

- ZFTurbo released fine-tuned viperx model (“ver. 2024.04”) on MVSEP (further trained from checkpoint on a different dataset). Ensembles will be updated tomorrow. Clicking issue has been fixed.

SDR vocals: 11.24, instrumental: 17.55 (from 17.17 in the base model)

Depends on a song if it's better. Some vocals can be worse vs the previous model.

- Test out ensemble 1296 + 1143 (BS-Roformer in beta UVR) + Inst HQ4 (dopfunk)

Ensembles with BS-Roformer models might not work for everyone, use manual ensemble if needed.

- Viperx model added also to beta Colab by jarredou. It gives only vocals, so perform inversion on your own to get instrumental

https://colab.research.google.com/drive/1pd5Eonbre-khKK_gn5kQPfTB1T1a-27p?usp=sharing

Update: now BS-Roformer is also added in the newer v.2.4 [Colab](#)

- Viperx' BS-RoFormer models have been implemented by Anjok to UVR

- BS/Mel-Roformer UVR beta patch

For GPU acceleration, UVR currently supports:
CUDA (NVIDIA GPUs)/DirectML (AMD and Intel GPUs; previously misnamed as OpenCL)/MPS (Mac M1 [ARM]/x86-64),
and even old CPUs (AMD A6-9225 Dual-Core or Intel Core 2 Quad [models with SSE4.1 tested]). DirectML is not supported for Apollo, Bandit (also incompatible with MPS), SCNet and probably Demucs 2 archs - CPU will be used automatically.

Model architectures supported:

MDX-Net, MDX23C (archs by kuelab), VR (voice-remover by tsurumeso, v. 4, 5 [UVR fork], 5.1), Demucs (arch by Meta; v. 1-4, only models trained on OG code, not MSST ones), BS-Roformer, Mel-Roformer (arch by Bytedance & impl. by lucidrains; issues on Linux explained later), SCNet, Apollo (in Tools; for upscaling, no DirectML acceleration), Bandit (SFX, no DirectML).

It has also a feature of ensemble MDX models with different archs like MDX23C/v3 or Roformers, VR.

Demudder added in newer patches (currently not on Linux).

Models for Roformers/SCNet/Bandit arch are located altogether in the MDX-Net menu.

Apollo upscaler (Apollo) is located in Tools.

Don't forget to enable GPU Conversion - if it works, it speeds up separation hugely.

- Min. 4GB VRAM GPUs tested (with chunk_size = 112455 in model yaml for Roformers; more below), On AMD, 16GB VRAM recommended (so no config modifications are required).
- Min. NVIDIA Maxwell/900 series GPUs/compute compatibility 5 is the minimum requirement for UVR to work (at least NVIDIA GTX 650 and GT 700 series and older are unsupported returning: "AssertionError: "" Traceback Error:" or "CUDNN_STATUS_NOT_INITIALIZED"), although DirectML should be theoretically supported by all DX12 GPUs.
- For AMD, at least RX 4GB models tested (not sure about R9 200 4GB GPUs - either if on newer modded Radeon-ID drivers and/or with downgraded DirectML.dll attached with drivers, copied to UVR\torch_directml folder, but seems like someone had occasional memory issues on HD 7870 2GB, but GPU Conversion still worked). "AssertionError: "" Traceback Error:" also exist when your AMD driver/Windows is outdated (then use e.g. [1.9.1.0](#) library).
- Intel was confirmed to work with ARC GPUs, and Xe integrated graphics (e.g. Tiger Lake 2021) with at least MDX-Net HQ (v2) models.
- If your separation on DirectML stuck, decrease [chunk_size](#).

- RTX 5000 series support on Windows was added in a separate UVR [patch](#) (or you can use OpenCL (DirectML) in options instead [slower]). The patch is not compatible with Intel/AMD GPUs, and potentially also older NVIDIA GPUs, giving the following error:
AttributeError: module 'torch_directml' has no attribute 'is_available'

In patch #12 a new “Inference mode” option in Advanced MDX-Net>Multi Network was implemented (disabled by default). In the current state, it fixes silence in separation on GTX 10XX and maybe older, but might make separations longer for other compatible GPUs. So if you have slower separations after updating UVR, check if GPU Conversion is still enabled (it’s rather disabled by default on new installations) and you can try to turn on Inference Mode if you have RTX GPU, or potentially GTX 16XX.

Download

(*if your download speed gets slow, use e.g. “Free Download Manager” on Windows or any other increasing connection count*).

- for Linux

Sadly, the currently released branch working on Linux is old and doesn’t support all models (at least without some workarounds below) and you can’t use WINE and keep GPU acceleration.

Some installation [directions](#) (from our [Discord](#)) from Roformer patch #1/2 period (or also [here](#)), plus comment out segmentation-models-pytorch 0.3.3 in requirements.txt [here](#) (line 136) (for Nvidia/CPU).

Current code repository for Roformers with DirectML support is located here (although at certain periods it might lack current patches):

(might work for non-Nvidia GPUs):

https://github.com/Anjok07/ultimatevocalremovergui/tree/v5.6.0_roformer_add%2Bdirectml

Judging by the date of files from 9 December 2024 in the repo at the moment, they seem to derive from outdated 12_8_24_23_30_BETA beta #9 patch (some newer models will fail with that codebase, plus it’s before chunk_size implementation, so it rather still uses dim_t).

Some other potentially useful information:

<https://github.com/Anjok07/ultimatevocalremovergui/issues/1674>

Or “you just need to use export PYGLET_SHADOW_WINDOW=0 and it’ll work”

If you get errors anyway, like [here](#):

“Getting requirements to build wheel did not run successfully. ... ModuleNotFoundError: No module named ‘imp’”

edit requirements.txt:

pyrubberband==0.3.0

PyYAML==6.0

scipy==1.9.3

playsound

numpy==1.23.5

Workarounding issues with Python 3.12:

<https://github.com/Anjok07/ultimatevocalremovergui/issues/1789>

Becruily karaoke model error fix:

"Those of you on linux running the current roformer_add+directml branch that cant get becruily's karaoke model working due to the same error: it seems editing line 790 in separate.py setting the keyword argument strict to False when calling load_state_dict seems to make the karaoke model load and infer properly, so I think it will work
model.load_state_dict(checkpoint, strict=False)

I don't know if this is a robust workaround, but I haven't observed anything behaving differently than it should yet, so if you want to give it a shot I think it will work

TL;DR change line 790 in separate.py to the codeblock and then run again and karaoke model should work" stephanie

ROCM instructions for AMD (also for Windows, but currently only using WSL)

- for better separation speed than DirectML:

<https://github.com/Anjok07/ultimatevocalremovergui/issues/1822#issuecomment-282436374>
7

You won't get DirectML acceleration to work using just WINE ([error](#)).

Fix for" "ModuleNotFoundError: No module named 'audioread'"

<https://github.com/Anjok07/ultimatevocalremovergui/issues/1797#issuecomment-331519105>
5

- Fixing issues in Matchering on Linux:

"Succeeded to run after modifying UVR.py :

```
match.process(  
    target=target,  
    reference=reference,  
    results=[match.save_audiofile(save_path, wav_set=self.wav_type_set)],  
)  
to  
  
match.process(  
    target=target,  
    reference=reference,  
    results=[match.pcm16(save_path)]  
)  
"
```

Below you'll find a download of ready packages

(they contain isolated Python environment - so you don't have to mess with your local Python installation - you don't even have to have Python installed on your computer)

- Fixing matching errors

<https://github.com/Anjok07/ultimatevocalremovergui/issues/2018#issue-3564850746>

- for macOS

- UVR Roformer beta patch #13.1

(beta_0115_MacOS_arm64_hf)

which applies a hotfix to address a few graphics issues. Be aware that it doesn't incorporate demudder from Windows patch #14:

- Mac M1 (arm64) users - [Link](#)

- Mac Intel (x86_64) users - [Link](#) (it went offline for some reason; older version below)

Note: Some people get error about failed malicious software check. Then check patch #9 below or read "MacOS Users: Having Trouble Opening UVR?" section in the [repo](#). Plus: "Functionality for systems running macOS Catalina or lower is not guaranteed".

Also: "What ended up working for me was to make sure that UVR lived in my root level Applications folder. I normally have an 'Installs' folder inside of Applications to help me keep track of things I have downloaded and installed. Nothing would load from the Installs folder, but worked fine from the root Applications folder."

Older Mac versions:

UVR beta Roformer patch #13

VR_Patch_1_15_25_22_30_BETA:

- Mac M1 (arm64) users - [Link](#)

- Mac Intel (x86_64) users - [Link](#)

UVR Roformer beta patch #9:

UVR_Patch_12_8_24_23_30_BETA:

Mac M1 (arm64) users - [Link](#)

Mac Intel (x86_64) users - [Link](#)

(Patch #9 fixes mainly Apollo arch issues)

Apollo arch was made compatible with MacOS MPS (metal) but it might be unstable and very RAM intensive - use chunk size over 7 to prevent errors.

Apollo is now compatible with all Lew models (fixed incompatibility with any other than previously available in Download Center). Fixed Matchering (presumably regression).

Changelog for Mac since patch #2 (older patches later below):

"Roformer checkbox now visible for unrecognized Roformer models" so now you can use custom Roformer models on MacOS Roformer patch without copying/modifying configuration files from Windows version or other users. Plus it includes all the previous fixes in the previously released patch (overlap code fixed, so no stem misalignment should occur

For help and discussion, visit our Audio Separation Discord: <https://discord.gg/ZPtAU5R6rP> | Download [UVR](#) or [MSST-GUI](#)

For inst/voc separation in cloud, try out free Colabs: [BS/Mel-Roformer](#) | [MDX23](#) (2-4 stems) | [MDX-Net](#) | [VR](#) | [Demucs 4](#) (2-6)

on certain overlap settings - higher overlap now means longer separation - so it's the opposite now)

- for *Windows*

- (optional) UVR Roformer beta patch #15 (or more precisely 14b) fixing issues with not working CUDA on RTX 5000 series GPUs. It might not be compatible with older GPUs.

- Full Install: [Download](#)

- UVR Roformer beta patch #13 fixing issue with no sound on some Roformer models (like avvew's de-reverb) on GTX 10XX or older:

UVR_Patch_1_15_25_22_30_BETA:

- Full Install: [Link](#)

- Patch Install (use if you still have non-beta UVR installed, e.g. 5.6, not 5.6.x): [Link](#)

- Small Patch Install (have any Roformer patch previously installed for this to work): [Link](#)

The issue was some older GPU's are not compatible with Torches "Inference Mode," (which is apparently faster) so it's now using "No Grad" mode instead. Users can switch back to using "Inference Mode" via the advanced multi-network options. [More](#)

- UVR Roformer beta small patch #14 - the long anticipated **demudder** added:

UVR_Patch_1_21_25_2_28_BETA: [Link](#)

It's a small patch (you must have a [Roformer Installation](#) [e.g. full #13 above] previously installed for this to work).

Also, minor bugs fixed, calculate compensation for MDX-Net v1 models added.

The MacOS version will be released later.

Demudder troubleshooting:

- Be aware that at least Phase Rotate doesn't work on AMD and 4GB VRAM GPUs on even 88200 chunk size (prev. dim_t 201 - 2 seconds) and 800MB Roformers like Becruily's, while 112455 (2,55s, prev. dim_t = 256) works fine for normal separation.

- In case of file not found error on attempt of using demudder, reinstall UVR.

- In case of Format not recognised error for demudder, keep Match freq cut-off enabled in MDX settings.

- "For Roformer models, it must detect a stem called "Instrumental" so for some models like Mel-Kim, you need to open model's corresponding yaml, and change "other" to "instrumental"."

"With the new config editor feature you could probably edit the configs of models to have the vocal stem labelled as the Instrumental stem so the demudder demuds the vocal stem, it definitely still makes a difference

I accidentally did this when installing another model, but it seems to actually have an effect on vocal stems too" stephanie

Demudder usage:

Go to options:>Advanced MDX-Net options>Enable Demudder

Then you can pick what Demudder variant you want and it will be used in every separation using MDX-Net models.

Demudder consists of three methods to choose from:

- Phase Rotate
- Phase Remix (Similar to X-Minus) - “the fullest sounding, but can leave a lot of artifacts with certain models. I only recommend that method for the muddiest models. Otherwise, Combined Methods is the best” “I don't recommend using phase remix on the Instrumental v1e model. I recommend combined methods or phase rotate for models produce fuller instrumentals.” Anjok
- Combine Methods (weighted mix of the final instrumentals generated by the above). More in the [full changelog](#).
- You can also use phase remix in [SESA](#) Colab

Demudder is “meant to solely target instrumentals. The vocals should stay exactly as before.”

“It works best on tracks that are spectrally dense (ex. Metal, Rock, Alternative, EDM, etc.) I don't recommend it for acoustic or light tracks.

I don't recommend using it with models that emphasize fuller instrumentals (like Unwa's v1e model).” Anjok

“I've noticed with the few amounts of tracks I've tried, demudding can sometimes accentuate instances of bleeding or otherwise entirely missed vocal-like sounds”. More in the [full changelog](#).

“I put the demuddled instrumental in the bleed suppressor, and it sounds really good, almost noise free. I either do a bleed suppressor or a V1/bleed suppressor ensemble” gilliaan

“I found that Phase Remix also works well on pop and other genres of music, but it only works using the vocal models (Phase Remix and VOICE-MelBand-Roformer Kim FT (from Unwa) or

VOICE-MelBand-Roformer Kim FT 2 (by Unwa).” Fabio

“I do plan on adding options to tweak the phase rotation.

I also plan on adding another combination method that may work better on certain tracks.”

Anjok

If you set 64-bit float output in Options>Additional settings, the results might be slightly less muddy, but also in very big size.

OG Discord [channel](#) to follow for updates

Older patches

(old) Potential fixing of RTX 5000 series CUDA acceleration for #14 patch - now unnecessary since dedicated patch was released above
(although there's no full success with the below, so also refer [here](#))

Some people reported that following the steps below might still result in slow separations:
Probably you'll be able to install required CUDA 12.8 and nightly PyTorch to fix the compatibility issue when following steps for [manual installation](#) (unfold it), so UVR won't use its own Python environment. In addition to the above link, use [this](#) repo with newer code, although for now, not the newest code is attached from the patches below... and all if you fix "Getting requirements to build wheel ... error" afterwards. Then you'll have a bug causing no GPU Conversion option functional - then you need to use:

python.exe -m pip install --upgrade torch --extra-index-url

<https://download.pytorch.org/whl/cu118>

instead of cu117 as in the instruction above (although for RTX 5090 it probably won't work, and you can use this <https://download.pytorch.org/whl/nightly/cu128> instead).

- Similar issue might occur when you don't install "onnxruntime-gpu" when the current is "onnxruntime" library (which does not support GPU)

- For Demucs UnpicklingError issue using manual installation:

modify the demucs/stats.py:

package = torch.load(path, 'cpu', weights_only=False)

Sometimes it still happens for e.g. Begruij inst model anyway. It can be the indicator that the model file is corrupted and has wrong CRC (most likely wrongly downloaded) - redownload the model.

On the old 5.6.0 version, there's OpenCL in options instead of DirectML in newer patches (although it's the latter).

Alternatively, you can use [MSST-GUI](#).

- Anjok released a new UVR beta Roformer patch #11 (Windows only for now)

It fixes 4 bugs: with VR post-processing threshold, Segment default in multi-arch menu, CMD will no longer pop-in during operations, and error in phase swapper.

[More](#) details/potential updates.

Standalone (for non-existent UVR installation)

[UVR 1 13 0 23 46 BETA full.exe](#)

For 5.6 stable (so for non-beta Roformer installation)

[UVR Patch 1 13 0 23 46 BETA rofo.exe](#)

Small (for already existing Roformer beta patch installation)

[UVR Patch 1 13 25 0 23 46 rofo small patch.exe](#)

- Patch #12 which is a hotfix for the [4 stem](#) BS-Roformer model by ZFTurbo (trained on MUSDB)

[UVR Patch 1 13 0 23 46 BETA rofo fixed.exe](#) (Windows only)

Users undergo some issues (no sound) with Mel-Roformer de-reverb by anvuew (a.k.a. v2/19.1729 SDR) since the latest UVR beta #11 or #12 update. Patch #10 works.

The issue seems to occur only on GTX 10XX series, and maybe older.

You should be able to use more than one UVR installation at the same time when one's been copied before updating (potentially patch #10 will still work) or use MSST repo and/or its GUIs.

UVR Roformer beta patch #9:

UVR_Patch_12_8_24_23_30_BETA

Windows: [Full](#) | [Patch](#) (Use if you still have non-beta UVR installed)

[Small Patch](#) (for this you must have a Roformer patch previously installed for this to work)

UVR Roformer beta patch #10

UVR_Patch_1_9_25_23_46_BETA_rofo_small_patch - [Link](#)

For now, only small patch for already existing beta Roformer installation above is available, and only for Windows.

If you have Python DLL error on startup, reinstall the last beta update using the full package instead, then the small installer from the newer patch.

Since beta version #10, **UVR doesn't rely on 'inference.dim_t' value for Roformers anymore** (if you were using edited "dim_t" value in yaml configuration files).

You have to edit audio.chunk_size instead if need it (e.g. for 4-12GB VRAM on AMD/Intel).

It's located "In model yaml config file, at top of it, chunk_size is first parameter (...) you can edit model config files directly inside UVR now." or in the new config editor in newer versions.

"Conversion between dim_t and chunk_size

dim_t = 801 is chunk_size = 352800 (8.00s)

dim_t = 1101 is chunk_size = 485100 (11.00s)

dim_t = 256 is chunk_size = 112455 (2,55s)

dim_t = 1333 is chunk_size = 587412 (13,32s)

[more values later below]

The formula is: $\text{chunk_size} = (\text{dim_t} - 1) * \text{hop_length}$ " - jarredou

Generally, to have the best SDR, use chunks not lower than 11s in the yaml for inference, which is usually training chunks value (rarely higher). Although, at times people get better results with 2,55s chunks, although some models behave worse than others with such small values.

"most of the time using higher chunk_size than the one used during training gives a bit better SDR score, until a peak value, and then quality degrades.

For Roformers trained with 8sec chunk_size, 11 sec is giving best SDR (then it degrades with higher chunk size)

For MDX23C, when trained with ~6sec chunks, iirc, peak SDR value was around 24 sec chunks (I think it was same for vit_large, you could make chunks 4 times longer)

How much chunk_size can be extended during inference seems to be arch dependant." - jarredou

Changelog #10:

Added SCNet and Bandit archs with models in Download Center, fixed compatibility with some newer Roformer models (prob. the Phantom center and 400MB small Unwa models, not sure yet), new Model Installer option added, model configuration menu enhanced, allowing aliases to selected models, added compatibility for Roformer/MDX23C Karaoke models with the vocal splitter, VIP code issue is gone, issues with secondary models options and minor bugs and interface annoyances are addressed, "improved the "Change Model Settings" menu. Now, any existing settings associated with a selected model are automatically populated, making it easier for users to review and adjust settings (previously, these settings were not visible even if applied).".

"Unfortunately, SCnet is not compatible with DirectML, so AMD GPU users will have to use the CPU for those models.

Bandit models are not compatible with MPS or DirectML. For those with AMD GPU's and Apple Silicon, those will be CPU only.

The good news is those models aren't all that slow on CPU." - Anjok

Changelog #9:

Apollo fixes: "Chunk sizes can now be set to lower values (between 1-6)

Overlap can be turned off (set to 0)"

Fix both for Apollo and Roformers: now 5 seconds or shorter input files no longer cause errors.

OpenCL was wrongly referenced in the UVR. It was actually DirectML all the way, and Anjok changed all the OpenCL names in the app into DirectML.

Changelog for all platforms:

Patch #3 fixed the issue with stem misalignment when using incorrect overlap setting for Roformers. Now it uses ZFTurbo code (also for MDX23C), meaning that **now increasing overlap for Roformers will result in increasing separation times** and potentially better SDR [the opposite of what it used to be in the previous beta Roformer patches #1 and #2]. Also, it "Fixed manual download link issues in the Download Center. Roformer models can now be downloaded without issue."). Also, new Roformer models were added to Download Center, so you don't have to download them manually.

- UVR Roformer beta patch #8 for Win: [full](#) | [patch](#) | Mac: [M1](#) | [x86-64](#):

UVR_Patch_12_3_24_1_18_BETA

Apollo arch was made compatible with OpenCL too, but it might be unstable and very RAM intensive - use chunk size over 7 to prevent errors (currently it's not certain that all models will work with less than 12GB of VRAM). At least in newer patches, it can straight up say that Apollo is not compatible with DirectML, and fall back to CPU mode.

Apollo is now compatible with all Lew models (fixed incompatibility with any other than previously available in Download Center). Fixed (presumably regression with) Matchering.

UVR Roformer beta patch #7 ([full](#) | [patch](#)) Win

UVR_Patch_12_2_24_2_20_BETA.

It introduces support for Apollo arch. The OG mp3 enhancer and Lew v1 vocal enhancer were added to Download Center. The arch is located in Audio Tools. Sadly, this arch cannot be GPU accelerated with OpenCL so AMD and Intel cards (you're forced to use CPU which might be long).

Also, "Phase Swapper" a.k.a. Phase fixer for Unwa inst models was added to Audio Tools.

Roformer beta patch #6: [M1](#) | [x86-64](#)

UVR Roformer beta patch #6: Win

UVR_Patch_11_25_24_1_48_BETA ([standalone](#) or [patch](#) - you can install it on stable 5.6 version already installed)

Fixes issues with viperx' models.

And with it, a long anticipated MDX-Net HQ_5 model was released (available for older versions in Download Center too. [Changelog](#)

UVR_Patch_UVR_11_17_24_21_4_BETA_patch_roformer (Beta [patch](#) #5 for UVR, Windows only):

"- Fixed OpenCL compatibility issue with Roformer & MDX23C models.

- Fixed stem swap issue with Roformer instrumental models in Ensemble Mode."

That patch is probably not standalone like patch #3, so have a previous UVR installation.

Roformer patch #4 for MacOS: [M1](#) | [x86-64](#)

UVR_Patch_11_17_24_21_4_BETA (Windows: [full](#) | [patch](#) | [changelog](#)) beta patch #4

[UVR_Patch_11_14_24_20_21_BETA_patch_roformer](#) (beta patch #3 "requires an existing UVR installation" so either the previous beta Roformer patch above or stable [5.6 patch](#).

[Full changelog](#).

[UVR_Patch_4_14_24_18_7_BETA_full_Roformer](#) | [mirror](#) (standalone Roformer beta patch #2, fixed OpenCL separation for AMD/Intel GPUs)

[UVR_Patch_3_29_24_5_11_BETA_full_roformer.exe](#) (older Roformer patch #1)

With the following issue fixed in the newer patch #2 above -

if you have playsound.py errors, disable notification chimneys in settings>additional settings, using OpenCL GPU acceleration (AMD) for BS-Roformer doesn't work (or at least not for everyone)

Older Roformer patch #1/2 for *MacOS* (ARM only) got deleted from the Discord server

(Roformer models are also added on MVSEP and x-minus.pro/uvronline.app and [MSST-GUI](#), and [inference](#) Colab and MDX23 v.2.4/2.5 [Colab](#))

Instructions for UVR Roformer patches and installing custom models

- Your current settings might be lost after patching your current UVR installation
- Applying newer UVR Roformer versions over some older UVR versions might cause errors on startup - then perform clean installation to fix the issue. Just make sure that after uninstalling UVR, nothing is inside the old UVR folder
- To perform clean installation of the latest version, for now you need:

Windows:

Roformer full patch #13:

UVR_Patch_1_15_25_22_30_BETA: [Link](#)

And then install small patch #14:

UVR_Patch_1_21_25_2_28_BETA: [Link](#)

RTX 5000 full patch #14:

UVR_Patch_4_24_25_20_11_BETA: [Link](#)

(iirc uses newer PyTorch and CUDA)

The above is enough for complete installation. Installing the 2023 5.6.0 version before is unnecessary.

MacOS:

- Roformer full patch #13.1 (standalone)

(beta_0115_MacOS_arm64_hf)

which applies a hotfix to address a few graphics issues:

- Mac M1 (arm64) users - [Link](#)
- Mac Intel (x86_64) users - [Link](#) (it went offline for some reason; older version #13 below [Link](#))

Linux

Old patch #9 only for now (no demudder, some fixable issues with certain models):

https://github.com/Anjok07/ultimatevocalremovergui/tree/v5.6.0_roformer_add%2Bdirectml

(for older patches and more info and troubleshooting refer [here](#))

- Some newer Mel/BS models really require the newest Roformer patches (otherwise you'll get MLP error).

- In the Download Center you'll find some BS/Mel models, but not all. Refer to the full list of models [here](#).

Installing custom Roformer models in UVR

(those unavailable in Download More Models a.k.a. Download Center)

- Since patch #10 new “Install Model” option was added.

Click RBM on the models’ list to access the option and follow the instructions on your screen.

- Models for Mel/BS Roformer, SCNet and Bandit Plus/v2 are located in the MDX-Net menu.

- For most models, you need both ckpt and yaml file for the model to work (if it's not the same with any config you already downloaded before - some models share the same config file - e.g. one folder with models in the repository might have only one yaml)

If you opened yaml file to download, but its content opened instead of its downloading started, press CTRL+S or go to options of your browser and find the option called “Save As”. Now you'll have a txt extension, but we need it to be yaml (otherwise UVR won't detect it), so choose All files in Extension, and edit it there manually to yaml (or after download).

- If you don't see ckpt on the extensions list, perform clean UVR installation from the patches above

- In “Set Model Type”, most Roformers will use Roformer non-v2 (and most are Mel).

For now, you should pick v2 Roformer type probably only for unwa 400MB experimental model (if you have lots of *layers* errors using Roformers, it means you picked v2 config unnecessarily).

- Manual model installation for e.g. Demucs which doesn't have “Install model” option (or for patch before #10) - on example of MDX/Roformers/SCNet/Bandit.

To install models from external sources (those unavailable in Download Center) you can copy the model file to models\MDX_Net_Models and .yaml config to models\model_data\mdx_c_configs, then after choosing the model in the app, press yes to recognize the model, wait a while. In older beta check also “roformer model” option when asked for configuration file and confirm (you cannot press confirm or check the option on the oldest Mac Roformer patch, the issue is explained below, and fixed in newer versions). (or [step by step](#))

Misc

- You might want to decrease default chunk_size in Edit Model Param (or yaml) for AMD/Intel GPUs with VRAM lower than 16GB if you have memory errors with GPU Conversion enabled or your separation is stuck on e.g. 5% (read more in [Common issues](#) later below)

- How would I assign a yaml config to an Apollo model on the new UVR [patch]?

“1. Open the Apollo models folder
2. Drop the model into the folder
3. From the Apollo models folder, drop the yaml into the model_configs directory
4. From the GUI, choose the model you just added and if the model is not recognized, a pop-up window will appear, and you'll have the option to choose the yaml to associate with the model.” - Anjok

- “batch_size = 2 (not less or more)” - “1 can lead to some clicks in output, while with batch_size>=2, there are no clicks. Clicks are obvious in low freq of log spectrogram” edit. iirc clicks with batch_size=1 could have been fixed (at least were with MSST from which the inference code was implemented in newer UVR patches, but iirc it wasn't used, and later the clicks were fixed in MSST).

- Segment size in the UVR UI does nothing for these Roformer models due to Advanced arch options being set by default to Segment Default which makes it being read from the yaml file. While "Segment_Default" in Advanced MDX-NET23 settings is checked, it will use the dim_t value from the bottom of the config. Simply dim_t is the segments. Although now chunk_size is used in newer patches instead.

- “The overlap value in yaml files is never used by UVR, only the value in GUI is used.” “UVR uses inference.dim_t from config as segment_size, but the inference.num_overlap is not used by UVR, it's always using the value in the GUI (while ZFTurbo original script is using audio.chunk_size and inference.num_overlap but not inference.dim_t . That's a mess) ” jarredou

Overlap comparisons for Roformers

4 is a balanced value in terms of speed/SDR according to [measurements](#) (since the beta patch #3 or later used above, overlap 16 is now the slowest in UVR (not overlap 2 is the slowest anymore when it was set the opposite) and overlap 4 has a bigger SDR than overlap 2 now).

Some people still prefer using overlap 8, while for others it's already an overkill. There's very little SDR improvement for overlap 32, and for 50 there's even a decrease to the level of overlap 4, and 999 was giving inferior results to overlap 16.

Compared to overlap 2, for 8 "I noticed a bit more consistency on 8 compared to 2 (less cut parts in the spectrogram)." Instrumentals with overlap higher than 2 can get gradually muddier.

The info is based on evaluations conducted on multisong dataset on MVSEP. Search for e.g. overlap 32 and overlap 16 below, and you will see the results to compare:

https://mvsep.com/quality_checker/multisong_leaderboard?algo_name_filter=kim

"overlap=1 means that the chunk will not overlap at all, so no crossfades are possible between them to alleviate the click at edges." fixed in MSST.

The setting in GUI overrides the one in yaml's setting.

chunk_size

"Most of the time using higher chunk_size than the one used during training gives a bit better SDR score, until a peak value, and then quality degrades.

For Roformers trained with 8 sec chunk_size [can be found in the yaml], 11 sec is giving best SDR (then it degrades with higher chunk size)" - jarredou

All the notable chunk_sizes are described later below.

Sometimes chunk_size can influence the ability of picking up e.g. some screams in the model ("kim's can do it if you mod the chunk size").

Lower chunks might sound a bit less muddy.

Unless they're set too high and VRAM is exceeded, while separation still doesn't fail on AMD/Intel, various values should provide similar separation times, esp. for NVIDIA users.

batch_size (not used in UVR, but in MSST)

Leave it default, but for faster inference, e.g. Gabox used 6. Using above 2 might increase VRAM usage. 1 is forced in the inference Colab (it has the clicking issue with that setting fixed in newer MSST code).

"i.e. instead of running a single song at batch_size = 1 you can run 2 at the same time (batch_size = 2)" so "you can use more than one song to process"

Most common issues

- Some models might occasionally disappear from your list (most likely sideloaded outside Download Center) and change their name (probably once they got added to Download Center - e.g. Melband Roformer karaoke ckpt)

- Since patch #3, Roformers' separation times with even smaller overlap are longer than before. Also, remember about inference mode added in one of the later patches. It's disabled by default as it causes silent separation issues on GTX 10XX GPUs and probably older. Enabling it on newer GPUs might be beneficial for performance
- If UVR freezes your PC occasionally during separation, you can change priority of UVR to Idle in Task manager, and the issue is gone sooner or later.

You can force it to remember every time you run UVR with Process Lasso (it autostarts with the OS, so you don't have to see the splash screen for a free user every time you want to use it).

- [Read](#) for RTX 5000 series issue, or use OpenCL (DirectML) in options (slower)

Ensembling impossible - model not visible in vocal splitter in UVR

- If user-imported Roformers aren't recognized in "instrumental/vocals" in ensemble or in vocal splitter, but are in "multi-stem ensemble":
 "The .yaml associated with the model usually needs to be updated to match UVR's stem naming conventions. For example, if your config shows the instruments as "other" and "vocals", it will need to be updated to "Instrumental" and "Vocals" (case-sensitive)" - Anjok
 E.g. for Karaoke models, you need to change "Karaoke" to "Vocals".

But you can also use Tools>Manual Ensemble instead for ready separations.

Ensemble vs multi-stem ensemble explained (by stephanie)

"The multi-stem ensemble mode is designed to ensemble every stem in all the models that were selected. I'll explain how it's related:

Let's say you have two vocal/instrumental models selected, and two 4-stem models selected (vocals, drums, bass, other), then it will process the song through all of the models. As a result, the final ensembled output will be 5 stems: vocals, instrumental, drums, bass, and other.

The drums, bass, and other stems will just be the ensemble of the 2 models in that selection that shared those outputs. However, all models in that particular selection shared a vocal output so it will be the ensemble of each model's vocal output

I'm not sure if that's very clear, but that's how it works and why the way it handles each stem is different from the other stem pair modes"

- Roformers might be sometimes slow/stuck/give memory allocation error during separation on AMD/Intel GPUs with VRAM lower than 16GB, if you don't lower default "chunk_size" during importing the model, or in the corresponding yaml in:
models\MDX_Net_Models\model_data\mdx_c_configs
or in Choose Model>Edit Models Config>Change Parameters>Edit Model Param.
Start with min. chunk_size = 112455 (dim_t 256 equivalent) and increase it depending on GPU or model size till you start getting errors to get the best possible SDR.

For older beta 1-9 and 4GB VRAM GPUs, lower dim_t at the bottom of the yaml (not at the top) to e.g. 256 or 201, sometimes 301 - some Roformers will require lower dim_t/chunk_size - the higher, the better till 1101, or training chunks value in the config. So since patch #10 "new beta version[s] doesn't rely on 'inference.dim_t' value anymore (if you were using edited "dim_t" value).

Now you have to edit audio.chunk_size now "In model yaml config file, at top of it, chunk_size is first parameter (...) you can edit model config files directly inside UVR now.

Memory issues - chunk_size table

dim_t to chunk_size conversion for Roformers and hop_length = 441 in the model's yaml

- useful if you see insufficient memory error

The formula is: $\text{chunk_size} = (\text{dim_t} - 1) * \text{hop_length}$

dim_t = 1700 is chunk_size = 749259 (17s) - used by inst resurrection

dim_t = 1333 is chunk_size = 587412 (13,32s)

dim_t = 1201 is chunk_size = 529200 (12s) - used by some newer models

dim_t = 1101 is chunk_size = 485100 (11.00s) - that dim_t value was giving the highest SDR for models trained with 8s chunks, at least in times of models released in beta Roformer beta patch #2 period, it's default for e.g. duality models

dim_t = 801 is chunk_size = 352800 (8.00s) - default for most models, max working on Intel/AMD 8GB GPUs on 900MB models

dim_t = 556 is chunk_size = 244755 (5,5s)

dim_t = 501 is chunk_size = 220500 (5s) - also, as below, but separation time slightly increased with at least a few browser tabs opened, higher crashes no matter what

dim_t = 456 is chunk_size = 200655 (4,5s) - works with Resurrection inst on 4GB AMD GPU

dim_t = 401 is chunk_size = 176400 (4s)

dim_t = 356 is chunk_size = 156555 (3,55s) - max supported dim_t for some becruily inst and AMD 4GB VRAM (when in previous beta, dim_t needed to correspond with overlap to avoid stem misalignment, so probably halves caused misalignment before),

it can be more muddy in certain parts of songs vs 256, the highest working value with becruily instrumental on AMD 4GB VRAM GPU

dim_t = 301 is chunk_size = 132300 (3s)

dim_t = 256 is chunk_size = 112455 (2,55s) - max working with e.g. becruily and smaller unwa's 400MB exp. models on 4GB AMD GPU

`dim_t = 201` is `chunk_size = 88200 (2s)` - required value for some more resource-hungry/bigger Roformers on AMD 4GB VRAM (some models only worked with that `dim_t` with earlier beta patches and here's it's probably the same or with 256 equivalent), but is still not low enough for most Roformers when demudder is used, and give "Could not allocate tensor" while single model separation previously worked with even bigger chunk setting. You shouldn't go lower with that parameter, as even a 2 seconds chunk might sometimes give audio skips every two seconds, at least on UVR Roformer patch #2. 2 or 2,5 seconds will be rather bare minimum." - jarredou (DTN edit)

All inst/voc Roformers seem to use `hop_length: 441` (but ensure in yaml), so you always multiply that hop value by the desired `dim_t - 1` to get correct `chunk_size` (e.g. corresponding with old `dim_t` values you were using in older beta patches)

- We have some reports about user custom ensemble presets from older versions no longer working (since 11/17/24 patch).
- Sadly, you need to get rid of them (don't restore their files manually) or the ensemble will not work and model choice will be greyed out. You need to start from scratch

Errors troubleshooting

"got an unexpected keyword argument 'linear_transformer_depth' "

In case of the error with any external Roformer model:

- delete "`linear_transformer_depth: 0`" line from the YAML file
- To fix issues with BS variant of e.g. anvuew's de-reverb model in UVR, additionally to the above, also change the following in the yaml file:
`stft_hop_length: 512` to `stft_hop_length: 441` so it matches the `hop_length` above" (thx lew).

If that line is not present in your model's yaml config file, go to the settings, then choose MDX In the Advanced menu, and click the "Clear auto-set cache" button.

Then go back to the main settings, click "Reset all settings to default" and restart the app (thx santilli_).

These issues don't happen in the ZFTurbo's CML inference code of:

<https://github.com/ZFTurbo/Music-Source-Separation-Training/>

'use_amp' "Key error"

using the GH repo above, and referencing (separating) some models:

- "add:
`use_amp: true`
in the training part of [models' yaml] config file (it's [missing](#))"

“”norm”” attributeError using e.g. unwa beta 5e model in UVR

- a) Ensure you installed [UVR Roformer patch](#) (5.6.1), and you're not using the old 5.6 version (but 5.6.1 is reported once you open the app)
- b) You could pick wrong model architecture in Install model option (so not Roformer, and generally v1), or haven't turned on “Roformer model” option during importing the model into UVR (option present in the older beta versions)
- c) Or “Edit the yaml file [of the model] from this -

training:

instruments:

- vocals

- other

target_instrument: vocals

use_amp: True

to this -

training:

instruments:

- Vocals

- Instrumental

target_instrument: Vocals

use_amp: True” - Anjok

[More](#) troubleshooting on the “norm” issue

E.g. BS-Roformer_LargeV1 is stuck on 5%

- Decrease chunk_size to 112455 (new patches)

(pre-10# patches) “Go to MDX settings, MX23C specific, turn off default segment size and use segment size 256, it's probably filling up your VRAM”

The setting resets itself. You should be able to set it permanently in the yaml configuration file of the model at the bottom (dim_t parameter).

It might be required for AMD/Intel 4GB VRAM GPUs (or potentially even 201, although it was using Rofo patch #2).

Q: “is there a way to reset which yaml file to use? I chose the incorrect yaml file for a particular ckpt file, and now I cannot change it”

A: Go to models list>Edit Model Config>Change parameters and choose the yaml from the list.

Or go to Ultimate Vocal Remover\models\MDX_Net_Models\model_data and if it was done just now, the last modified yaml in this folder will be the one corresponding to your model.

You can just open that json, and edit it to write the config name located in mdx_c_configs you want to use with that model. You should find proper hashed yaml by the saved model of your choice, but if you really feel lost, you can delete all hashed yaml's, so you'll need to go through the process of choosing configs for all custom MDX and Roformer models, but remember to not delete "model_data.json" and "model_name_mapper.json" as they cover models added to Download Center or written to be recognized automatically by UVR, so it's rather not a place you look for. Also, you can decode hashed json names corresponding to specific models [here](#).

Layers error - for issues with Unwa 400MB model

-First make sure you're running the latest patch. If you're on the latest patch, It might be trying to associate with an incompatible YAML", but resetting the parameters below might be enough.

Go into the mdx_c_configs folder

Find and delete BS_Inst_EXP_VRL.yaml

Go back into the "Download Center"

Select "MDX-Net" and give it a moment.

Close the Download Center and try again.

If that doesn't work, you might have a previous json model file that's interfering:

Select the model in the MDX-Net model menu

Then select "Edit Model Config"

From the popup, click "Reset Parameters"" Anjok

Layers errors - general

a) You didn't install the newest patch and still use e.g. beta 2 with some newer model

b) You could check Roformer v2 instead of v1 during installing custom model

TypeError: (...) freqs_per_bands

You probably set Mel-Roformer model type instead of BS-Roformer when it was necessary.

Go to MDX-Net, pick the model>Edit model config>Choose parameters.

There you should change Model type.

More troubleshooting

- If you have:

RuntimeError: ""

Traceback Error: "

without any text in these lines on AMD GPU on every attempt of using GPU Conversion in UVR for all archs and models (you probably use outdated GPU drivers and/or Windows), go to Ultimate Vocal Remover\torch_directml and replace DirectML.dll from C:\Windows\System32\AMD\ANR (make backup before). Experimentally, you can use this

older [1.9.1.0](#) version of the library. Restart UVR after replacing the file!

If you use an incompatible library version, you'll encounter the "Unhandled exception" startup issue.

Be aware that the linked older version of the library might cause additional noise for MDX-Net v2 models like HQ_X (the issue is gone when you turn off GPU Conversion).

- All MDX-Net v2 models (maybe beside 4 stem variants), have so called MDX noise, which can be cancelled by using Options>Advanced MDX-Net Settings>Denoise Output>Standard (or Model)

- At least beta #2 Roformer update caused some stability and performance issues with other archs than Roformers for some people when specific parameters started to take more time than before.

Roll back to stable 5.6 (non 5.6.1) in these cases if necessary. Possibly make a copy of the old installation. Your configuration files might be lost. You can use both installations at the same time (or at least when one, e.g. Roformer patch is installed or symlinked in the default location).

- (I think I covered that issue above more thoroughly)

Roformer models in at least patch #2 work only in "Multi-Stem" mode in UVR. Using them in Ensemble causes layers errors (you can use manual Ensemble instead).

lirc, it's caused by yaml config where instead of Instrumental + Vocals (with V as capital letter) there's written other + vocals, and you need to change it. lirc it doesn't happen on models downloaded from Download Center as Anjok was fixing the issue, but the problem might still exist in yaml of some custom models outside the center

- If you have sudden issues with not being able to separate, try to reinstall the app, and/or possibly make sure you didn't turn on some power saving option in your laptop. Plus, you can simply try to reopen UVR (few fail tries on incompatible DirectML.dll with your GPU driver/OS will hang UVR on "Loading Model" till you close UVR manually from Task Manager).

You'll find more UVR troubleshooting in [this](#) section

Problems fixed in newer patches

- (deprecated since patch #10 - now convert dim_t it to chunk_size) dim_t = 1101 seems to be a sweet spot in terms of speed/SDR according to [measurements](#) (although on 1 minute files); use 1120 if UVR refuses to accept 1101 in GUI (or edit yaml file)

- (deprecated since patch #10) Some Roformer configs have wrong dim_t at the bottom of the yaml by default (e.g. 256), change it at the bottom of the yaml config for better SDR (not the one at the top), e.g. to 1101 (more explanations on it later).

- (fixed in patch #10) VIP code in Roformer beta patch #2-9 (and probably #1) doesn't work - Download all the VIP models you need before patching older 5.6 to beta Roformer or use two installations of the UVR if you can't use patch #10 with the fix.
- (fixed in patch #6) People experience All stems error with viperx' 12xx models in newer versions of UVR Beta Roformer patch (patch #2 was the last confirmed to work with these older models)
- (fixed in patch #10) mlp_expansion_factor: 1 or (when mlp line is deleted from yaml) mismatch for MelBand Roformer error
You probably use older Roformer patch incompatible with newer models (e.g. #2)
It also appears when you wrongly set v2 model type.

Fixed in the beta patch #3 and #4 for all platforms

- Don't set overlap higher than 11 for 1101 dim_t (**at the bottom** of yaml file in the "inference" section, not above) and overlap 8 for 801 - these two are the fastest settings before stem misalignment issues occur. Otherwise, it can lead occasionally to some effects or synths missing from the instrumental stem (although some rules can be broken here with various settings). Also, the problems with clicks are alleviated with these good settings.
 - In beta #2 patch, best measured SDR for both Mel and BS-Roformers is when dim_t = 1101 in the inference section of yaml config and when overlap is set to 2 in GUI (although 1 wasn't tested, and is actually lower). But the last beta patches, all bigger overlap values are slower, so SDR might be higher with higher values.
Be aware that it will increase separation time. Maximum allowed value before error is 1801, but 1501 or 1601 depending on a model will be the max reasonable for experiments before some unwanted downsides of too high or too low dim_t appear (disappearing of some stem elements). In some specific cases, 1333 (or potentially 1301) was giving better results than 1101 or 1501, but it depended on song length - usually it happened on short fragments.
 - Instruction for overlap and dim_t above applies to other Roformer models as well, and not only those in Download Center. With the instructions, you can achieve faster separation times, as you're not forced to use the most time-consuming overlap 2 in older patches to avoid stem misalignment issues
-

Model characteristics

(the list might be getting outdated, read models list at the [top](#))

Note: E.g. unwa's duality models v1/2 and inst v1/2/v1e are now added to UVR Beta

Roformer Download Center (so you don't have to mess with models and configs manually)

- 1053 model separates drums and bass in one stem, and it's very good at it
(although now it might be better to use Mel-Roformer drums on [x-minus.pro/uvronline](#))
“Target is drums and bass, and “other” is the rest. Despite that, it says vocals”
- Unwa released a new Inst v1e [model](#) | [Colab](#) | [MSST-GUI](#) (“The model [yaml] configuration is the same as v1”)
“The “e” stands for emphasis, indicating that this is a model that emphasizes fullness.”
- unwa inst v2 - it gets muddier than v1 at times, but it has less of noise
- unwa inst v1 - focused on instrumental stem:
[model](#) | [Colab](#) | [MSST-GUI](#) | [phase fixer](#)
“much less muddy (...) but carries the exact same UVR noise from the [MDX-Net v2] models”
But it's a different type of noise, so aufr33 denoiser won't work on it.
“you can “remove” [the] noise with uvr denoise aggr -10 or 0” although with -10 it will make it sound more muddy like Kim model and synths and bass are sometimes removed with the denoiser (~bebruily). Mel-Roformer denoise might be better for it.
bebruily released a Python [script](#) fixing the noise issue (execute “pip install librosa” in case of module not found error) - it sounds similar to the method used for premium user on x-minus.
- unwa beta 4 Mel-Roformer (fine tune of Kim's voc/inst model):
<https://huggingface.co/pcunwa/Mel-Band-Roformer-big/tree/main> | [Colab](#)
Be aware that the yaml config has changed, and you need to download the new beta4 yaml.
“Metrics on my test dataset have improved over beta3, but are probably not accurate due to the small test dataset. (...) The high frequencies of vocals are now extracted more aggressively. However, leakage may have increased.” - unwa
“one of the best at isolating most vocals with very little vocal bleed and still doesn't sound muddy” “gives fuller vocals”. Can be a better choice on its own than some ensembles.
- unwa duality model - focused on both stems, and instrumental is similarly muddy like in beta 4
- Kim Mel-Band Roformer vocal model
It's less muddy than 1296/1297.
([original repo](#) - CML faster on CUDA than in UVR | [model](#) | [config](#) - place the model file to models\MDX_Net_Models and .yaml config to model_data\mdx_c_configs subfolder and “when it will ask you for the unrecognised model when you run it for the first time, you'll get some box that you'll need to tick “roformer model” and choose it's yaml” (Mac issue explained in the section above).
(simple [Colab/CML inference/x-minus/MVSEP/jarredou Colab](#) too now)
- unwa BS-Roformer finetuned a.k.a. large (further trained viperx 1297 model) [download](#)

More muddy than Kim above, a bit less of vocal residues, a bit more artificial sound.

- Mel-RoFormer Karaoke / Lead vocal isolation model files released by Aufr33 and viperx ([download](#))

Older models in Download Center

- older viperx' 1297 model tend to be a bit better for instrumentals, and 1296 for vocals (both more muddy than Kim and Unwa models, but "still pretty good for voice cleaning" and dealing with noise) - BS-Large model by Unwa is a fine-tune of that model.

- 1143 model is the first Mel-Roformer trained by viperx before Kim introduced changes to the config, which fixed the problem of lower SDR vs models trained on BS-Roformer. Use Kim Mel-Roformer instead

Both models struggle with saxophone and e.g. some Arabic guitars. It can still depend on a song whether these are better than even the second oldest Roformer than on MVSEP (from before viperx model got fine-tuned version). They tend to have more problems with recognizing instruments. Other than that, they're very good for vocals (although Mel-Roformer by Kim on x-minus tends to be better).

Muddy instrumentals when not ensembled with other archs.

Be aware that names of these models on UVR refer to SDR measurements of vocals conducted on private viperx dataset, not even older Synthetic dataset, instead of on multisong dataset on MVSEP, hence the numbers are higher than in the multisong chart on MVSEP.

Infos and fixes for older patch #1/2 (with matching overlap (reversed) and dim_t necessity)

- To avoid separation errors for 4GB VRAM and AMD/Intel GPUs using Roformers, set segments 32, overlap 2 and dim_t 201 with num_overlap 2 both at the bottom of yaml config in \models\MDX_Net_Models\model_data\mdx_c_configs (dim_t 301 and overlap 3 also works, although not on all models [e.g. not for beta 3, but inst v1] and seems to be less muddy and fewer clicks appear).
dim_t 201 is not optimal setting and might lead to more occasional quiet residues, clicks or sudden volume changes (like chunk was changing every 2 seconds), although there's no stem misalignment issue with these settings (they work both for Mel and BS Roformers).
dim_t 301 with lighter models seems to be a bare minimum to avoid the majority of audible artefacts (after patch #3 dim_t 256 is allowed - "make sure you check the "Segment Default" in MDXNET23 Only Options for it to take effect").

Using the settings above on patch #2, with GPU acceleration it will take 39m 28s for 3:28 song using 1296 model on RX 470 4GB and 18 minutes for Kim Mel-Roformer and 3:01 song.

Using HQ_4 is much faster than realtime using default settings, but even longer than accelerated Roformer, when on CPU only using old Core 2 Quad @3.6 DDR2 800MHz.

On Mac M1 using the patch above, it takes 9 minutes to process a 3-minute song using BS-Roformer (dim_t 1101, batch size 2, overlap 8) with “constant throttling”. [Click](#)

And below 4 minutes for Kim Mel-Roformer (overlap 1, dim 801). [Click](#)

- Settings working for 6GB AMD GPUs: dim_t 601 or 701 at the bottom of the yaml file and overlap 6 or 7 in GUI.
- Like I mentioned, overlap 8 can be good enough too when dim_t=801 is set (the fastest setting before SDR getting drastically reduced), at least in other cases you shouldn't exceed 6, while 2 should provide the best quality in most cases.
- 1602 (or rather 1601) dim_t might lead to less wateriness, but turns out in cost of a bit more of vocal residues.
- “In theory, max overlap value [for Roformer separations without mentioned issues in UVR] can be known with formula:

$$(\text{dim_t} - 1) / 100 = \text{Max_overlap_value}$$

if dim_t = 801:
 $(801 - 1) / 100 = 8$

if dim_t = 1101:
 $(1101 - 1) / 100 = 10$ [jarredou wrote 10 here, but it's actually 11]

Above that max value, some parts of the input will not be processed.

The lower the overlap value is, the more overlap is used, so better SDR.

- Some Rofos models still have wrong config by default, with dim_t=256, so max overlap value for that is 2. That's why I've advised to stick to overlap=2” - jarredou
So in times before dim_t was known how to be correctly set, so now overlaps can be even set to 8 now when dim_t=801 is set].”

The same thing applies for both BS and Mel Roformers in UVR.

“audio.dim_t value is not used with roformers in ZFTurbo script, it uses audio.chunk_size and then it's parameters in the model part of config.”

- Using older Roformer beta patches for **Mac M1** doesn't allow you to choose the Roformer parameter to check for custom Roformer models and only config name can be chosen, but no confirm button is available. So the error "File "libv5/tfctdfv3.py", line 152, in __init__" appears.

> Place the corresponding json file with your model from [this](#) repo into: models\MDX_Net_Models\model_data beforehand, to fix the issue.

In some cases, you may still get the same error anyway and to get rid of it, you need to edit manually model_data.json adding desired model line at the end like your custom model was downloaded from download center. On example of unwa's beta 3:

```
},
"d43f93520976f1dab1e7e20f3c540825":{
  "config_yaml": "config_melbandroformer_big.yaml",
  "is_roformer": true
}
```

Additionally, you need the model at the end of model_data_mapper.json:

```
"model melband_roformer_big_beta3.ckpt": "config_melbandroformer_big"
}
```

Now copy the hash-named json file (d43f93520976f1dab1e7e20f3c540825.json for beta 3) to model_data folder.

All the three modified files for beta 3 and other models [here](#).

If you have problems generating hash on first launch of the model and your model is not uploaded in the repo above or json is not generated then use Windows installation in VM, or ask some PC user for the config. Potentially reading [Hash decoding](#) can be helpful.

But maybe your hashed config name will be generated correctly already after you imported the model into UVR (although no confirmation button might prevent it), and now it will be enough to just place the following line like in the jsons presented above: "is_roformer": true" (so after " , in the yaml line above).

- More in-depth - Settings per model SDR vs Time elapsed -||- (incl. dim_t and overlap evaluation for Roformers) - [click](#) or [here](#) | [conclusion](#) - made before patch #3

Older news follow

- The viperx model was also added on MVSEP

- New ensembles with higher SDR were added on MVSEP

- BS-Roformer model trained by viperx was added on x-minus (it's different from the v2 model on MVSEP, and has higher SDR, it's the "1.0" one). If it's better vs V2 might depend on a song.

It struggles with saxophone and e.g. some Arabic guitars.

- (x-minus - aufr33) "I have just completed training a new UVR De-noise model. Unlike the previous version, it is less aggressive and does not remove SFX.

It was trained on a modified dataset. I reduced the noise level and made it more uniform, removed footsteps, crowd, cars and so on from the noise stems. On the contrary, the crowd is now a useful / dry signal. (...) The new model is designed mainly to remove hiss, such as preamp noise."

For vocals that have pops or clipping crackles or other audio irregularities, use the old denoise model.

- Dango.ai updated their model, also giving some kind of demudder to the instrumentals, enhancing their results. Results might be better than MDX23C and BS-Roformer v2. Still, it's pretty pricey (8\$ for 10 separations). 5x 30 seconds fragments per IP can be obtained for free, and usually it doesn't reset. "It's \$8 for 10 tracks x 6 minutes, all aggressiveness modes included (but vocal and inst models are separate). The entire multisong dataset for proper SDR check would cost around \$133." becruily

- Be aware that queues on <https://doubledouble.top/> are much shorter for Deezer than Qobuz links. If there's no 24 bit versions for your music, use Deezer instead.
[outdated; currently there's no longer any MQA files on Tidal] Also, avoid Tidal and 16 bit FLACs from "Max" quality, which is slightly lossy MQA. Use 24 bit MQA from Tidal only when there's no 24 bit on Qobuz. Most older albums under 2020 are 16 bit MQA instead of 24 bit MQA on Tidal, and are lossy compared to Deezer and Qobuz which doesn't use MQA (so doubledouble doesn't convert MQA to FLAC like on Tidal). MQA is only "slightly" lossy, because it affects frequencies mainly from 18kHz and up, and not greatly.

- Members of neighboring AI Hub server made a fork of KaraFan Colab updated with the new HQ_4 and InstVoc HQ2 models. It has slow separation fix applied. [Click](#)

- HQ_4 and Crowd models added to HV Colab temp [fork](#) before merge with main GH repo

- (MVSEP) "We have added longer filenames disabling option to mvsep, you can access it from Profile page

20240312034817-b3f2ef51cb-ballin_bs_roformer_v2_vocals_[mvsep.com].wav ->
ballin_bs_roformer_v2_vocals.wav

Due to browser caching, you might want to hard refresh the page if you have downloaded onc"

- The ensembles for 2 and 5 stems on MVSEP have been updated with bigger SDR bag of models containing now new BS-Roformer v2 (with MDX23C, VitLarge23, and for multistem, the old demucsht_ft, deumcs_ht, demucs_6s and demucs_mmi models)
- All the Discord direct links leading to images in this document have expired. I already reuploaded some more important stuff. Please ping me on Discord if you need access to some specific image. Provide page and expired link.
- <https://free-mp3-download.net> has been shut down. Check out alternatives [here](#). New Apple Music ALAC/Atmos downloader added, but its installation is a bit twisted and subscription is required. Murglar added.
- MDX-Net HQ_4 model (SDR 15.86) released for UVR 5 GUI! Go to Models list>Download center>MDX-Net and pick HQ_4 for download. It is an improved and faster than HQ_3, trained for epoch 1149 (only in rare cases there's more vocal bleeding, more often instrumental bleeding in vocals, but the model is made with instrumentals in mind). Along with it, also UVR-MDX-NET Crowd HQ 1 has been added in download center.
- HQ_4 model added to the Colab:
https://colab.research.google.com/github/kae0-0/Colab-for-MDX_B/blob/main/MDX_Colab.ipynb
- New BS-Roformer v2 model released on MVSEP. It's more aggressive model than above.
- [Fixed](#) KaraFan Colab with the fix for slow non-MDX23 models. You'll no longer stack on voc_ft using any other preset than 1, but be aware that it will take 8 minutes more to initialize. (same fix as suggested before, but w/o console, as it wasn't defined, and faster ort nightly fix doesn't work here).

Turns out, there has been an official non-nightly package released, and it works with KaraFan correctly (no need to wait 8 minutes any longer):
!python -m pip -q install onnxruntime-gpu --extra-index-url
https://aiinfra.pkgs.visualstudio.com/PublicPackages/_packaging/onnxruntime-cuda-12/pypi/simple/

- (x-minus.pro) "Since Boosty is temporarily not accepting PayPal and generally working sucks, I made the decision to go back to Patreon. Please be aware that automatic charges will resume on March 22, 2024. If you have Boosty working correctly and do not intend to use Patreon, please cancel your Patreon subscription to avoid being charged. If you wish to switch from Boosty to Patreon, please wait for further instructions in March." Aufr33
- If you suffer from bleeding in other stem of 4 stems Ripple, beside decreasing volume by e.g. 3/4dB also "when u throw the 'other stem' back into ripple 4 track split a second time, it

works pretty well [to cancel the bleeding]" if it's still not enough, put other stem through Bandlab Splitter.

- If you suffer from vocal residues using Ensemble 4 models on MVSEP.com, decrease volume of input file by -8dB "now it's silent. No more residue" usually 3 or 4dB was doing the trick for Ripple, but here it's different. Might depend on a song too.

- Image Line "released an update for FL Studio, and they improved the stem separation and it's better, but it has quite a bit of bleeding still, but it also seems they may have improved the vocal clarity"

- (probably fixed in new HV MDX) Our newly fixed VR and newer HV MDX Colabs started to have issues with very slow initialization for some people (even 18 minutes/+ instead of normally 3). It's probably due to very slow download of some dependencies. Possible solutions: use other Google account, use VPN, make another Google account (maybe using Polish VPN). Let us know if it happens only for some specific dependency or all of them. You can try to uncomment the ORT nightly line in mounting cell (add # before), as it triggers more dependencies to be installed, which can be slow in that case. The downside is - there won't be GPU acceleration, and one song will be processed in 6-8 minutes instead of ~20 seconds.

- New paid drum separation service:

<https://remuse.online/remusekit>

It uses free [drumsep](#) model (same model hash: 9C18131DA7368E3A76EF4A632CD11551)

- MDX Colab seem to not work due to Numpy issues. I already fixed them in Similarity Colab, and hopefully reimplement the fixes elsewhere soon. [VR Colab](#) fixed too. Tech details about introduced changes described below [Similary Extractor](#) section.

- [Music AI](#) surfaced. Paid - \$25 per month or pay as you go ([pricing chart](#)). No free trial. Good [selection](#) of models and interesting [module stacking](#) feature. To upload files instead of using URLs "you make the workflow, and you start a job from the main page using that custom workflow" [~ D I O ~].

Allegedly it's made by Moises team, but the results seem to be better than those on Moises. "Bass was a fair bit better than Demucs HT, Drums about the same. Guitars were very good though. Vocal was almost the same as my cleaned up work. (...) I'd say a little clearer than mvsep 4 ensemble. It seems to get the instrument bleed out quite well, (...) An engineer I've worked with demixed to almost the same results, it took me a few hours and achieve it [in] 39 seconds" Sam Hocking

- "I just got an email from Myxt saying they're going to limit stem creation to 1 track per month. For creator plan users (the \$8 a month one) and 2 per month for the highest plan. So I may assume with that logic, they're gonna take it away for free users?"

- (probably fixed) For all jarredou's MDX23 v. 2.3 Colab fork users:

"Components of VitLarge arch are hosted on Huggingface... when their maintenance will be finished it will work again. I can't do anything about it in the meantime."

2.2 and 2.1 and MVSEP.com 4-8 models ensemble (premium users) should work fine.

- Ripple now has fade in and clicking issues fixed. Also, there's less bleeding in the other stem (but Bas Curtiz' trick for -3dB/-4dB input volume decreasing can be still necessary). "Ripple's lossless outputs are weird, some stems like the drums are semi full band (kicks go full band, snares not etc) and the "other" stem looks like fake full band". These fixes are applied also for old versions of the app.

Also, the lossless option fixes to some extend the offset issue so it's more similar to input now, but not identical (lossless option might require updating). Also no more abrupt endings

Ripple = better than CapCut as of now (and fullband).

plus Ripple fixed the click/artifacts using cross-fade technique between the chunks.

- ViperX currently doesn't plan to release his BS-Roformer model

- New "uvr de-crowd (beta)" model added on x-minus. Seems to provide better results than the MVSEP model. Also, an MDX arch model version is planned for training.

"At minimum aggressiveness value, a second model is now used, which removes less crowd but preserves other sounds/instruments better."

- Ripple seems to have a lossless export option now. "First make sure the app is updated then click the folder then click the magnet icon then export and change it to lossless"

- Seems like CapCut now has added separation inside Android Capcut app in unlocked Pro version

<https://play.google.com/store/apps/details?id=com.lemon.lvoverseas> (made by ByteDance)
Seems like there is no other Pro variant for this app.

At least unlocked version on apklite.me have a link to regular version, so it doesn't seem to be Pro app behind any regional block. But -

"Indian users - Use VPN for Pro" as they say, so similar situation like we had on PC [Capcut](#) before. Can't guarantee that unlocked version on apklite.me is clean. I've never downloaded anything from there.

- Mega, GDrive and direct link support for input files added on MVSep. If you want to apply MVSep algorithm to result of other algorithm, you can use "Direct link" upload and point https link on separated audio-file on MVSep.

- If you have an issue with Demucs module not found in e.g. MDX23 v.2.3 Colab (now fixed there and also in [VR Colab](#)), here's a solution:

"In the installation code, I added `!pip install samplerate==0.1.0` right before the `!pip install -r requirements.txt &> /dev/null` and I managed to get all the dependencies from the requirements.txt installed properly." (derichtech15)

- If you repost your images or files from Discord elsewhere while cutting link after "ex=" for all new posted files, it will make your files expire pretty soon (17.02.24). If you leave the full link with "ex=" and so on, it won't expire so fast, but who knows if not later.

So far, all the old Discord images shared elsewhere with "ex=" cut, work (also in incognito without Discord logged in), but it's not certain that it will be that way forever.

Discord announced in the end of 2023, that they'll update their mechanisms of sharing links, so they'll expire after some time when they're shared, to avoid some security vulnerabilities allowing scams. Or they just want to offload the servers.

- [OpenVINO™](#) AI Plugins for Audacity [3.4.2](#) 64-bit introduced.

4 stems separation, noise suppression, Music Style Remix - uses Stable Diffusion to alter a mono or stereo track using a text prompt, Music Generation - uses Stable Diffusion to generate snippets of music from a text prompt, Whisper Transcription - uses whisper.cpp to generate a label track containing the transcription or translation for a given selection of spoken audio or vocals.

Not bad results. They use Demucs.

- For people with low VRAM GPUs (e.g. 4GB or less), you can test out [Replay](#) app, which provides voc_ft model and tends to crash less than UVR. Sadly, the choice of models is much smaller, but it has some de-reverb solution. [Screenshot](#)

- Latest MVsep changes:

- 1) All ensembles now have option to output intermediate waveforms from independent algorithms + additional max_mag, min_mag.
- 2) Ensemble All-In now includes DrumSep results extracted from Drum stem.

- resemble-enhance ([GH](#)) model added on x-minus in denoise mode. It can work better than the latest denoise model on x-minus. It is intended only for vocals. For music use UVR De-noise model on x-minus.

- (fixed in kae, 2.1, 2.2 [and KaraFan irc] Colabs) All Colabs using MDX-Net models are currently very slow. GPU acceleration is broken and separations now only work on CPU with onnxruntime warnings.

To work around the issue, go to Tools>Command palette>Use fallback runtime version (while it's still available).

Downgrading CUDA to 11.8 version fixes the issue too, but it takes 9 minutes in order to install that dependency, so it's faster to use fallback runtime till it's still available. After that period, just execute this line after initialisation cell:

```
console('apt-get install cuda-11-8') and GPU acceleration will start to work as usual.  
>"Better fix [than CUDA 11.8] until final version is released, using that onnxruntime-gpu  
nightly build for cuda12:  
!python -m pip install ort-nightly-gpu  
--index-url=https://aiinfra.pkgs.visualstudio.com/PublicPackages/_packaging/ort-cuda-12-nig  
htly/pypi/simple/
```

(no need to install cuda 11.8)" jarredou

In case of credential issues you can try out this package instead:

```
!python -m pip -q install onnxruntime-gpu --extra-index-url  
https://aiinfra.pkgs.visualstudio.com/PublicPackages/_packaging/onnxruntime-cuda-12/pypi/  
simple/
```

- LarsNet model was added on MVSep. It's used to separate drums tracks into 5 stems: kick, snare, cymbals, toms, hihat. Source: <https://github.com/polimi-ispl/larsnet>

It's worse than Drumsep as it uses Spleeter-like architecture, but "at least they have an extra output, so they separate hihats and cymbals.". [Colab](#)

"Baseline models don't seem better quality than drumsep, but the provided checkpoints are trained with only 22 epochs, it doesn't seem much. (and STEMGMD dataset was limited by the only 10 drumkits), so it could probably be better with better dataset & training"

" it separates the toms so much better [than Drumsep]"

Similar situation as with Drumsep - you should provide drums separated from e.g. Demucs model.

- Captain FLAM from [KaraFan](#) asks for some help due to some recent repercussions.
You can support him on https://ko-fi.com/captain_flam

- To preserve instruments which are counted as vocals by other MDXv2 models in KaraFan, use [these](#) preset 5 modified settings (dca100fb8).

- Added more remarks from testing these settings against sax preset and others.

- drumsep added on MVSEP!
(separation of drums from e.g. Demucs 4 stem or "Ensemble 8 models"/+)

- New Bandit Plus model added on MVSEP
"I trained Bandit for vocals. But it's too far away from MDX23C" -ZFTurbo
"I loved this bandit plus model!! It has great potential."

- UVR De-noise model by FoxJoy added on x-minus. It's helpful for light noise, e.g. vinyl.
(de-reverb and de-echo are up already)

New MDX de-noise model is in the works and beta model was also added!

"the instruments in the background are preserved much better than the FoxJoy model"

It works for hiss, interference, crackle, rustles and soft footsteps, technical noise.

- New hifi-gan-bwe Colab fork made by jarredou:

https://colab.research.google.com/github/jarredou/hifi-gan-bwe/blob/main/HIFIGAN_BWE.ipynb

- New AI speech enhancer - <https://www.resemble.ai/introducing-resemble-enhance>

- Reason 12.5 (a DAW) was released with VST3 plugin support

- jazzpear94 "I made a [new model](#) with a modified version of my SFX and Music dataset with the addition of other/ambiant sound and speech. It's a multistem model and should even work in UVR GUI as it is MDX23C.

Note: You may want to rename the config to .yaml as UVR doesn't read .yml and I didn't notice till after sending. Renaming it fixes that, however"

"You put config in models\mdx_net_models\model_data\mdx_c_configs. Then when you use it in UVR it'll ask you for parameters, so you locate the newly placed config file."

"Keep in mind that the cinematic model focus is mainly on sfx vs instruments voice stems are supplemental. Usually I remove voices first"

- <https://github.com/karnwatcharasupat/bandit>

Better SDR for **Cinematic** Audio Source Separation (dialogue, effect, music) than Demucs 4 DNR model on MVSEP (mean SDR 10.16>11.47)

- "[Demucs+CC_Stereo_to_5.1](#)" - it's a script where you can convert Stereo 2.0 to 5.1 surround sound. Full [discussion](#) about script. They use MVSep to get steams and after use script on them.

- [Colab](#) by jazzpear96 for using ZFTurbo's MSS training script. "I will add inference later on, but for now you can only do the training process with this!"

- New djay Pro 5.0 has "very good realtime stems with low CPU" Allegedly "faster and better than Demucs, similar" although "They are not realtime, they are buffered and cached." it uses AudioShake. It can be better for instrumentals than UVR at times.

- AudiosourceRE Demix Pro new version has lead/backing vocals separation

- New **crowd** model added on MVSEP (applause, clapping, whistling, noise) (and got updated by the time 5.57 -> 6.06; added hollywood laughs, old models also available)
- VitLarge23 model on MVSEP got updated (9.78>9.90 for instrumentals)
- MelBand RoFormer (9.07 for vocals) model added on MVSEP for testing purposes
 "The model is really good at removing the hi-hat leftovers. These e.g. in the Jarredou colab sometimes when you can hear the hi-hats from the acapella. And Melband roformer can almost remove all the hi-hat leftovers from the acapella."
 "are the stems not inverted result? for me it sounds like there is insane instrument loss in the instrumental stem and vocals loss in the vocal stem, yet there is no vocal bleed in instrumental stem and vice versa" "I also think that the vocals are surprisingly clean considering the instrumentals sound quite suppressed but also clean"
- Goyo Beta plugin for dereverb stopped working on December 2nd (as it required internet connection and silent authorization on every initialization). They transitioned to paid Supertone Clear. They send BETA29 coupon over emails (with it, it's \$29).
- New MVSep-MDX23 Colab Fork v2.3 by jarredou published under new Colab link [here](#)
 Now it has Vitlarge23 model (previously used exclusively on MVSEP) instead of HQ3-Instr, also improved BigShifts and MDXv2 processing.
 Doesn't seem to be better than RipX which is better in preserving some instruments, and also removes vocals completely
- Check out new [Karaoke](#) recommendations (dca100fb8)
- Dango.ai finally received English web interface translation
- New SFX model based on Mel roformer was released by jazzpear94. [More info](#)
- User friendly [Colab](#) made by jarredou and [forked](#) by jazzpear94 with new feature. In case of some problems, use WAV file.
- Seems like Ripple got updated, "it sounds a lot better and less muddied" doesn't seem to give better results for all songs, though. Might be similar case with Capcut too.
- Hit 'n' Mix RipX DAW Pro 7 released. For GPU acceleration, min. requirement is 8GB VRAM and NVIDIA 10XX card or newer (mentioned by the official document are: 1070, 1080, 2070, 2080, 3070, 3080, 3090, 40XX, so with min. 8GB VRAM). Additionally, for GPU acceleration to work, exactly "Nvidia CUDA Toolkit v.11.0" is necessary. Occasionally, during transition from some older versions, separation quality of harmonies can increase.
 Separation time with GPU acceleration can decrease from even 40 minutes on CPU to 2 minutes on decent GPU.
- UVR BVE v2 beta has been updated on x-minus

"It now performs better on songs with 2 people singing the lead
No longer separates the second lead along with it"

-dca100fb8 found out new [settings](#) for [KaraFan](#) which give good results for some difficult songs (e.g. Juice WRLD) for both instrumental and acapella. It's now added as preset 5.

Debug mode and God mode can be disabled, as it's like that by default.

"It's like an improved version of Max Spec ensemble algorithm [from UVR]"

Processing time for 6:16 track on medium setting is 22 minutes.

- New MDX23C model added exclusively on MVSEP:

vocals SDR 10.17 -> 10.36

instrum SDR 16.48 -> 16.66

Also ensemble 4 got updated by new model (10.32>10.44 for vocals)

- For some people using mitmproxy scripts for Capcut (but not everyone), they "changed their security to reject all incoming packet which was run through mitmproxy. I saw the mitmproxy log said the certificate for TLS not allowed to connect to their site to get their API. And there are some errors on mitmproxy such as events.py or bla bla bla... and capcut always warning unstable network, then processing stop to 60% without finish."

~hendry.setiadi

"At 60% it looks like the progress isn't going up, but give it idk, 1 min tops, and it splits fine." - Bas

-ZFTurbo published his training code:

<https://github.com/ZFTurbo/Music-Source-Separation-Training>

"It gives the ability to train 5 types of models: mdx23c, htdeumcs, vitlarge23, bs_roformer and mel_band_roformer.

I also put some weights there to not start training from the beginning."

It contains checkpoint of e.g. 1648 (1017 for vocals) MDX23C model to train it further.

Be aware that the older bs_roformer implementation is very slow to train IRC.

Vitlarge23 "is running 2 times faster than MDX models, it's not the best quality available, but it's the fastest inference"

"change the batch size in config tho

I think zfturbo sets the default config suited for a single a6000 (48gb)
and chunksize"

"A small update to the backing vocals extractor [on X-Minus]

Now you can more accurately specify the panning of the lead vocal." ~Aufr33 [Screen](#)

- IntroC created a [script](#) for mitmproxy for Capcut allowing fullband output, by slowing down the track. [Video](#)

- Jazzpear created new VR SFX model. Sometimes it's better, sometimes it's worse than [Forte's](#) model. [Download](#)

For UVR 5.x GUI, use these parameters (irc same as Forte):

User input stem name: SFX

Do NOT check inverse stem!

1band sr44100 hl 1024

- Now KaraFan should work locally on 4GB GTX GPUs (e.g. laptop 1060), on presets 2 or 3, and with chunk 500K, speed can be slowest. Download on GitHub the Code > ZIP

- Bas Curtiz' new video on how to install and use Capcut for separation incl. exporting:

<https://www.youtube.com/watch?v=ppfyl91bJlw>

and saving directly as FLAC, although the core source of FLAC is still AAC in this case:

<https://www.youtube.com/watch?v=gEQFzj6-5pk>

"It's a bit of a hassle to set it up, but do realize:

- This is the only way (besides Ripple on iOS) to run ByteDance's model (best based on SDR).
- Only the Chinese version has these VIP features; now u will have it in English
- Exporting is a paid feature (normally); now u get it for free

The instructions displayed in the video are also in the YouTube description."

Capcut normalizes the input, so you cannot use Bas' trick to decrease volume by -3dB like in Ripple to workaround the issue of bleeding (unless you trick out the CapCut, possibly by adding some loud sound in the song with decreased volume, something like presented [here](#)).

- (fixed) KaraFan Colab will be fixed on 27th at morning.

- There's a workaround for people not able to split using Capcut. The app discriminate based on country (poor/rich) and paywalls Pro option.

The [video](#) demonstration for below

0. Go offline.

1. Install the Chinese version from capcut.cn

2. Use these files copied over your current Chinese installation, and don't use English patch.

3. Open CapCut, go online after closing welcome screen, happy converting!

4. Before you close the app, go offline again (or the separation option will be gone later).

Before reopening the app, go offline again, open the app, close welcome screen, go online, separate, go offline, close. If you happen to miss that step, you need to start from the beginning of the instruction.

(replacing [SettingsSDK](#) folder no longer works after transition from 4.6 to 4.7, it freezes the app)

FYI - the app doesn't separate files locally.

- Bas Curtiz found out that decreasing volume of mixtures for Ripple by -3dB eliminates problems with vocal residues in instrumentals in [Ripple. Video](#).

This is the most balanced value, which still doesn't take too many details out of the song due to volume attenuation.

Other good values purely SDR-wise are -20dB>-8dB>-30dB>-6dB>-4dB> /wo vol. decr.

The method might be potentially beneficial for other models and probably work best for the loudest tracks with brickwalled waveforms.

- Stable 5.6 OpenCL (DirectML) version of UVR 5 GUI for Windows

Supporting AMD and Intel GPUs acceleration but no Roformers yet

https://github.com/Anjok07/ultimatevocalremovergui/releases/download/v5.6/UVR_v5.6.0_setup_directml_old.exe

Mac: <https://github.com/Anjok07/ultimatevocalremovergui/releases/>

(newer [beta Roformer](#) [with "roformer" in the installer name] supports both DirectML and CUDA out of the box already; for Mac M1 [click](#)).

- For CUDA (NVIDIA GPUs) - non-OpenCL installer in the name from here:

https://github.com/Anjok07/ultimatevocalremovergui/releases/download/v5.6/UVR_v5.6.0_setup.exe

(Following based on previous OpenCL build)

8GB VRAM for 3:00/3:30 tracks using MDX23C HQ model with 12GB VRAM probably enough for 5:00 track which is more than in CUDA.

Now the issue should be mitigated, and less memory crashes should occur.

Ensembles might require more memory due to memory allocation issues not met in CUDA before. Also, VRAM is fully freed only after closing the application.

Acceleration for only Demucs 2 (and 1?) arch on AMD is not supported. All others archs should work.

- Be aware that there was also **full MPS (GPU) acceleration introduced for Mac M1** for all MDX-NET Original Models (HQ3, etc.), all MDX23C Models, all Demucs v4 models (no VR models acceleration on GPU). So don't use Windows in VM to run UVR anymore, but separate using dmg installer from [releases](#) section (ARM). GPU acceleration is 3x faster than separation took on CPU before.

- "MDX23C-InstVoc HQ 2 is out as a VIP model [for UVR 5]! It's a slightly fine-tuned version of MDX23C-InstVoc HQ. The SDR is a tiny bit lower, but I found that it leaves less vocal bleeding." ~Anjok

It's not always the case, sometimes it can be even the opposite, but as always, all can depend on specific song.

- jarredou's MDX23 2.2 Colab should allow separating faster, and also longer files now (tech details)

- All-in ensemble added for premium users of MVSEP - it has vocals, vocals lead, vocals back, drums, bass, piano, guitar, other. Basically 8 stems (and from drums stem you can further separate single percussion instruments using drumsep - up to 4 instruments, so it will give 10 stems in total).

- <https://www.capcut.cn/> (outdated section: [read](#))

Is a new Windows app which contains Ripple/SAMI-Bytedance inst/vocal model (not 4 stems like in Ripple).

"At the moment the separation is only available in Chinese version which is jianyingpro, download at capcut.cn [probably here - it's where you're redirected after you click "Alternate download link" on the main page, where download might not work at all]

Separation doesn't require sign up/login, but exporting does, and requires VIP.

Separated vocal file is encrypted and located in

C:\Users\yourusername\AppData\Local\JianyingPro\User Data\Cache\audioWave"

The unencrypted audio file in AAC format is located at \JianyingPro

Drafts\yourprojectname\Resources\audioAlg (ends with download.aac)

Drag and drop it in Audacity or convert to WAV (<https://cloudconvert.com/aac-to-wav>)

"To get the full playable audio in mp3 format a trick that you can do is drag and drop the download.aac file into capcut and then go to export and select mp3. It will output the original file without randomisation or skipping parts"

"Trying out Capcut, the quality seems the same as the Ripple app (low bitrate mp3 quality) at least the voice leftover bug is fixed, lol"

Random vocal pops from Ripple are fixed here.

Also, it still has the same clicks every 25 seconds as before in Ripple.

Some people cannot find the settings on this screen in order to separate. Maybe it's due to lack of Chinese IP, or Chinese regional settings in Windows, but logging wasn't necessary from what someone told.

- Looks like the guitar model on MVSEP can pick up piano better than the available there piano model in lots of cases (isling)

- AudioSep has been released

<https://github.com/Audio-AGI/AudioSep>

(separate anything you describe)

<https://replicate.com/cjwbw/audiosep?prediction=j7dsrvtxfm3gjax3vfzb7py>

(use short fragments as input)

https://colab.research.google.com/github/badayvedat/AudioSep/blob/main/AudioSep_Colab.ipynb (basic Colab)

<https://huggingface.co/spaces/badayvedat/AudioSep> (it's down)

"so far it's ranged from mediocre to absolutely horrible from samples I've tried"

"So far[,] it does [a] great job with crowd noise/cheering."

Didn't pick piano.

Output is mono 32kHz. Where input is 30s, the output can be 5s.

- UVR started to process slower for some people using Nvidia 532 and 535 drivers (at least Studio ones on at least W11). [More](#) about the issue. Consider rolling back to 531.79.

"Took 10 seconds to run Karaoke 2 on a full song (~5[]mins), with the latest drivers it took like 20 minutes". The problem may occur once you reboot your system.

- AMD GPU acceleration has been introduced in the official UVR repo under a new branch on GH. Beta as exe patch will be released in the following days. Currently, it supports only MDX-Net, but not MDX23C, and Demucs 4 models (not 3) and VR arch (5.0, but not 5.1). Currently, GPU memory is not clearing, so you need a lot of VRAM in order to use ensembles.

- (x-minus) "Added additional download buttons when using UVR BVE model.**

Now you can download:

- song without backing vocals
- backing vocals
- instrumental without vocals
- all vocals" Anjok

- MacOS UVR versions should be fixed now - redownload the latest 5.6 patches. GPU processing on M1 is fully functioning with MacOS min. Monterey 12.3/7 (only VR models will

crash with GPU processing). It's very fast for the latest MDX23C fullband model - 11 minutes vs 1 hour on CPU previously.

- Cyrus version of MedleyVox Colab with chunking introduced, so you don't need to perform this step manually

<https://colab.research.google.com/drive/1StFd0QVZcv3Kn4V-DXeppMk8Zcbr5u5s?usp=sharing>

"Run the 1st cell, upload song to folder infer_file, run 2nd cell, get results from folder results = profit"

"one annoying thing is that it always converts the output to mono 28k"

- Separation times since the UVR 5.6 update increased double for some people. Almost the same goes to RAM usage.

Having lots of space on your system disk or additional partition assigned for pagefile can be vital in fixing some crashes, especially for long tracks. Be aware that CPU processing tends to crash less, but it's much slower in most cases.

"I realized that with 2-3h long audio files, I was able to use Demucs, after I added another 32GB of RAM. In Total my system got 64GB and I increased the swap file to 128GB, which is located on an NVME drive.... so just in case the 64GB RAM are not enough, which I experienced with the "Winds" model, it's not crashing UVR, instead using the swap."

- Segments set to default 256 instead of 512 is $\frac{1}{3}$ faster for the new MDX23C fullband model at least for 4GB cards. But it's still very slow on such RTX 3050 mobile variant (20 minutes for 3:40 song).

- Sometimes inverting vocals with mixture using MDX23C instead of using instrumental output can give better results and vice versa.

"Differences were more significant with D1581 [than fullband], but secondary vocals stem has "a bit" higher score" ([click](#)). Generally inversion of these MDX23C models (but not spectral) was giving sometimes better results.

- MedleyVox [Colab](#) preconfigured to use with Cyrus model

Newer model epochs can be found here:

<https://huggingface.co/Cyru5/MedleyVox/tree/main>

Q: What is isnet?

A: It's basically just another model that builds on top of what I've built so far that performs better. That's the surface level explanation, at least.

- Settings for v2.2.2 Colab

https://colab.research.google.com/github/jarredou/MVSEP-MDX23-Colab_v2/blob/v2.2/MVSep-MDX23-Colab.ipynb

If you suffer from some vocal residues, try out these settings

```
BigShifts_MDX: 0
overlap_MDX: 0.65
overlap_MDXv3: 10
overlap demucs: 0.96
output_format: float
vocals_instru_only: disabled
```

Also, you can manipulate with weights.

E.g. different weight balance, with less MDXv3 and more VOC-FT.

- As an addition to [AI-killing tracks](#) section, and in response to deletion of "your poor results" channel, there was recently created a [Gsheat](#) with your problematic tracks to fill in. It is open to everyone to contribute.

- Video [tutorial](#) by Bas Curtiz how to install MedleyVox (based on Vinctekan fixed source). Cyrus trained a model. MD serves to separation of various singers from a track. It sometimes does a better job than BVE models in general.

Sadly, it has 24kHz output sample rate, but AudioSR works pretty good for upscaling the results.

https://github.com/haoheliu/versatile_audio_super_resolution

<https://replicate.com/nateraw/audio-super-resolution>

<https://colab.research.google.com/drive/1ILUj1LvrP0PyMxyKTfIDJ-o2Nrk8w7?usp=sharing>

Be aware that it may not work with full length songs - you might need to divide them into smaller 30 seconds pieces.

- "Ensemble 4/8 algorithms were updated on MVSep with new VitLarge23 model. All quality metrics were increased:

Multisong Vocals: 10.26 -> 10.32

Multisong Instrumental: 16.52 -> 16.63

Synth Vocals: 12.42 -> 12.67

Synth Instrumental 12.12 -> 12.38

MDX23 Leaderboard: 11.063 -> 11.098

I added Ensemble All-In algorithm which includes additionally piano, guitar, lead/back vocals.

Piano and guitar has better metrics comparing to standard models, because they are extracted from high quality "other" stem. Lead/back vocals also has slightly better metrics.

piano: 7.31 -> 7.69

guitar: 7.77 -> 8.95" ZFTurbo

- New vocal model added on MVSEP:

"VitLarge23" it's based on new transformers arch. SDR wise (9.78 vs 10.17) it's not better than MDX23C, but works "great" for ensemble consisting of two models with weights 2, 1.

- MVSEP-MDX23-Colab fork v2.2.2 is out.

It is now using the new InstVocHQ model instead of D1581:

https://github.com/jarredou/MVSEP-MDX23-Colab_v2/

Memory issues with 5:33 songs fixed (even 19 minutes long with 500K chunks supported)

It should be slightly faster than the previous version, as the extra processing for the fullband trick is not needed anymore with the new model.

Q: Why is "overlap_MDX" set to 0.0 by default in MVSEP-MDX23-Colab_v2 ?

A: because it's a "doublon" with MDX BigShifts (that is better)

- Stable final version of UVR v5.6.0 has been released along with MDX23C fullband model (the same as on MVSEP) - SDR is 10.17 for vocals & 16.48 for instrumentals.

It's called MDX23C-InstVoc HQ.

<https://github.com/Anjok07/ultimatevocalremovergui/releases/>

Be aware it's taking much more time to process a song with it, than all previous models. Also, it doesn't require volume compensation set. It can leave more vocal residues than HQ_3 models for some songs. On the other hand, it can give very good results with song with "super dense mix like Au5 - Snowblind" but also for older tracks like Queen - March Of The Black Queen (always caused issues, but it gave the best result so far, although still lot of BV is missed).

Performance:

- 3:30 track with HQ_3 takes up to 24 minutes on i3-3217u while the new model takes 737 minutes (precisely 1:34 vs 41:00 for 15 seconds song).

- RTX 3060 12 GB - takes around 15 minutes to process a 25 minutes file with the new model.

- GTX 1080 Ti took about 4 minutes to process, about a 5 min 30 song

- If you upgraded from beta, Matchering might not work correctly. In order to fix the error: Go to the Align tool.

Select another option under "Volume Adjustment", it can be anything.

Now, matchereng should work. The fix may not apply for Linux installations.

- KaraFan [original](#) Colab seems to work now (v. 3.1) but one track with default settings takes 30 minutes for 3:37 track on free T4 (the last files processed are called Final) and it can get you disconnected from runtime quick (especially if you miss some multiple captcha prompts). V. 3.1 can have more vocal residues than in 1.x version and even more than in HQ_3 model on its own.

You might want to consider using older versions of KF with [Kubinka](#) Colab.

- Now 3.2 version was released with less vocal residues.

As mentioned before, after runtime disconnection error, output folder still constantly populated with new files, while progress bar is not being refreshed after clicking close or even after closing your tab with Colab opened.

- "Image-Line the company that made FL Studio 21 took to Instagram announcing a beta build that allows the end users to separate stems from the actual program itself, this is in beta and isn't final product"

People say it's Demucs 4, but maybe not ft model and/or with low parameters applied or/and it's their own model.

"Nothing spectacular, but not bad."

- FL Studio bleeds beats, just like Demucs 4 FT

- FL Studio sounds worse than Demucs 4 FT

- Ripple clearly wins"

- Org. KaraFan [Colab](#) with v. 3.0 should work with the large GPU option disabled (now done by default).

- You may be experiencing issues with KaraFan 3.0 alpha (e.g. lack of 5_F-music with which the result was better before), and using [Kubinka Colab](#) which uses the older version for now has some problems with GPU acceleration. Maybe the previous KF commit will work or even the one before (2.x is used [here](#)).

- New UVR beta patches for Windows/Mac/M1 at the bottom of the release note

<https://github.com/Anjok07/ultimatevocalremovergui/releases/>

Usually check for newer versions above, but this one currently fixes long error on using the new BVE model

https://github.com/Anjok07/ultimatevocalremovergui/releases/download/v5.5.0/UVR_Patch_9_20_23_20_40_BETA.exe

- "The new BVE (Background Vocal Extractor) model [in UVR 5 GUI] has been released!"

To use the BVE model, please make sure you use the [UVR Patch 9 18 23 18 50 BETA](#) patch ([Mac](#)). Remember, it's designed to be used in a chain ensemble, not on its own. It's better to utilize it via "Vocal Splitter Options". ~Anjok"

Using Lead vocal placement = stereo 80% is still only available on X-Minus only. UVR GUI doesn't support this yet - it's for the situation when your main vocals are confused with backing vocals.

- In the latest UVR GUI beta patch, vocal stems of MDX instrumental models have polarity flipped. You might want to flip it back in your DAW.

- Investigating KaraFan shapes issue > [link](#)
- New piano and guitar models added on MVSEP. Use other stem from e.g. "Ensemble 8 models" or [MDX23 Colab](#) or ht demucs_ft for better results.
- To separate electric and acoustic guitar, you can run a song (e.g. other stem) through the Demucs guitar model and then process the guitar stem with GSEP (or MVSEP model instead of one of these).
Gsep only can separate electric guitar so far, so the acoustic one will stay in the "other" stem.
- New UVR beta patch implements chain ensemble from x-minus for splitting backing and lead vocals. To use it:
 1. Enable "Help Hints" (so you can see a description of the options),
 2. Go to any option menu
 3. Click the "*Vocal Splitter Options"
 4. From there you will see the new chain ensemble options.
- [Patch](#) (patching from the app may cause startup issues)
- "New MDX23C model improved on [MVSEP] Leaderboard from 10.858 up to 11.042"
- "For those of you who were running into errors related to missing *"msvcp140d.dll"* and *"VCRUNTIME140D.dll"* after installing the latest patch, it's been fixed." -Anjok
[UVR_Patch_9_13_23_17_17_BETA](#)
- The UVR's latest beta 9 patch causes startup issue for lots of people on even clean Windows 10. No fix for it. Copying libraries manually or installing all possible redistributables doesn't work. In such case, use beta 8 patch.
- If you see an error that you're disconnected from KaraFan Colab, it can still separate files in the background and consume free "credits" till you click Environment>Terminate session. It happens even if you close the Colab.
So, you can see your GDrive output folder still constantly populated with new files, while progress bar is not being refreshed after error of runtime disconnection or even after Closing your tab with Colab.
- KaraFan got updated to 1.2 (eg. model picking was added). Deleting your old KaraFan folder on GDrive can be necessary to avoid an error now in Colab.
- KaraFun - next version of MDX23 fork (originally developed by ZFTurbo, enhanced and forked by jarredou) has been created by Captain FLAM (with jarredou's assistance on tweaks).

Official [Colab](#) (video [guide](#) in case of problems)

[Colab](#) forked by Kubinka (can show error now after 1.2 update)

GUI for offline use: <https://github.com/Captain-FLAM/KaraFan/tree/master>

It gives very clean instrumentals with much less of consistent vocal residues than in MDX23 2.0-2.2 and Ripple/Bytedance.

(might have been changed) You can also disable SRS there to get a bit cleaner result, but in cost of more vocal residues. How detestable it will be without SRS, depends on a track - e.g. if it has heavy compressed modern vocals and lots of places with not busy mix (when not a lot of instruments play). Disabled SRS adds a substantial amount of information above 17.7kHz.

One of our users had problems caused seemingly by empty Colab Notebooks folder which he needed to delete. Could have been something else they did too, though.

- New epoch of new BVE model has been added to x-minus

“In some parts the new BVE is better, in some it's worse. Still a great model”

> To get better results, you can downmix the result to mono and repeat the separation

- For people having issues with Boosty x-minus payment:

<https://boosty.to/uvr/posts/5d88402e-9eb1-4046-a00a-cf8b09e27561>

- Sometimes for instrumental residues in vocals, AIs for voice recorded with home microphone can be used (e.g. Goyo [now Supertone Clear], or even Krisp, RTX Voice, AMD Noise Suppression, Elgato Wave Link 3.0 Voice Focus or Adobe Podcast as a last resort) it all depends on type of vocals and how destructive the AI can get.

- Izotope Ozone 11 has been released. It's 1200\$ for Advanced Edition. It's the only version possessing Spectral Recovery. Music Rebalance is said to have Demucs instead of Spleeter now.

<https://www.izotope.com/en/products/ozone.html>

- Acon Digital has released [Remix](#), their first plug-in capable of real-time separation to five stems: Vocals, Piano, Bass, Drums, and Other.

“Just listened to the demo, not great but still”

- [RemFX](#) for detection and removal of the following effects: chorus, delay, distortion, dynamic range compression, and reverb. [Huggingface](#) (currently stopped working) | [Samples](#)

The [Colab](#) is slow while downloading [checkpoints](#) from zenodo (400KB/s for 1GB file out of 6), later it stopped working.

Outputs in at least Huggingface are mono, may not work in every case, the website in general doesn't work well with big files, keep them short, 0-30 seconds.

Sometimes 30 seconds is still not enough on Colab and it throws OutOfMemoryError.

It's not better than our dereverb model in UVR.

To fix Colab:

"speechbrain lib API was totally changed in recent 1.0.0 version, it's working if you downgrade it:

`!pip install speechbrain==0.5.16"`

OG [repo](#) for running locally.

- Beta UVR patch also released for x86_64 & M1 Macs:

https://github.com/TRVlvr/model_repo/releases/download/uvr_update_patches/UVR_Patch_8_28_23_2_9_BETA_MacOS_x86_64.zip

"If you have any trouble running the application, and you've already followed the "MacOS Users: Having Trouble Opening UVR?" instructions here, try the following:

Right-click the "Ultimate Vocal Remover" file and select "Show Package Contents".

Go to -> Contents -> MacOS ->

Open the "UVR" binary file."

In case of further issues, check this out:

<https://www.youtube.com/watch?v=HQsazeOd2lw&feature=youtu.be>

Looks like e.g. with Denoise Lite models it can ask for parameters. Set 4band_v3 and 16 channels, press yes on empty window.

"The Mac beta is not stable yet." - Anjok

"The new beta [UVR] patch has been released! I made a lot of changes and fixed a ton of bugs. A public release that includes the newest MDX23 model will be released very soon. Please see the change log via the following message -

https://discord.com/channels/708579735583588363/785664354427076648/1145622961039_101982

Patch:

https://github.com/TRVlvr/model_repo/releases/download/uvr_update_patches/UVR_Patch_8_28_23_2_9_BETA.exe

-"I found a way to bypass the free sample limits of Dango.ai. With VPN and incognito, when the limit appears, change the date on the computer or other device (I set the next day) and

close and re-open the incognito tab. Sometimes it can show network error, in such case restart the VPN and re-enter in incognito again" Tachoe Bell

- Bas' guide to change region to US for Ripple on iOS

https://media.discordapp.net/attachments/708595418400817162/1146727313963237406/Ripple_iOS_iPad_mini_2_-_demo.mp4

- Another way to use Ripple without Apple device

Sign up at <https://saucelabs.com/sign-up>

Verify your email, upload this as the IPA:

<https://decrypt.day/app/id6447522624/dl/cllm55sbo01nfoj7yifiyucaa>

Rotating puzzle captcha for TikTok account can be tasking due to low framerate. Some people can do it after two tries, others will sooner run out of credits, or completely unable to do it.

- Every 8 seconds there is an artifact of chunking in Ripple. Heal feature in Adobe Audition works really well for it:

<https://www.youtube.com/watch?v=Qqd8Wjqtx-8>

-The same explained on RX 10 example and its Declick feature:

<https://www.youtube.com/watch?v=pD3D7f3ungk>

- Ripple/SAMI Bytedance's API was found. If you're Chinese, you can go through it easier. The sami-api-bs-4track (the one with 10.8696 SDR Vocals) - you need to pass the Volcengine facial/document recognition apparently only available to Chinese people

<https://www.volcengine.com/docs/6489/72011>

We already evaluated its [SDR](#), and it even scored a bit better than Ripple itself.

This is the Ripple audio uploading API:

<https://github.com/bitelchux/TikTokUploader/blob/2a0f0241a91b558a7574e6689f39f9dd9c39e295/uploader.py>

there's a sample script on the volcengine SAMI page

"API from volcengine only return 1 stem result from 1 request, and it offers vocal+inst only, other stems not provided. So making a quality checker result on vocal + instrument will cost 2x of its API charging

something good is that volcengine API offers 100 min free for new users"

API is paid 0.2 CNY per minute.

It takes around 30 seconds for one song.

It was 1.272 USD for separating 1 stem out MVSEP's multisong dataset (100 tracks x 1 minute).

- (outdated) Using Ripple on an M1 remote machine turned out to be successful but very convoluted.

<https://discord.com/channels/708579735583588363/708579735583588366/1143710971798507520>

-It is possible that "a particular song that an older version of mdx23 (mdx23cmodel3.ckpt) has a much better extraction than D1581 and the current 4 model ensemble on MVSEP for preserving the instruments (also organ-like instruments)"

-Seems like Google raised Colab limit for free users from 1 hour to 5 hours. It depends on a session, but in most cases you should be able to perform tasks taking above 4 hours now.

-How to change region to US in Apple App Store to make "[Ripple - Music Creation Tool](#)" (SAMI-Bytedance) work.

<https://support.apple.com/en-gb/HT201389>

<https://www.bestrandoms.com/random-address-in-us>

Or use [this](#) Walmart address in Texas, the number belongs to an airport.

Do it in App Store (where you have the person-icon in top right).

You don't have to fill credit cards details, when you are rejected, reboot, check region/country... and it can be set to the US already.

Although, it can happen for some users that it won't let you download anything forcing your real country.

"I got an error because the zip code was wrong (I did enter random numbers) and it got stuck even after changing it.

So I started from the beginning, typed in all the correct info, and voilà"

If "you have a store credit balance; you must spend your balance before you can change stores".

It needs (an old?) a sim card to log your old account out if necessary.

- Long awaited app made by Bytedance with one of their SAMI variants from MDX23 competition which holds top of our MVSEP leaderboard was published on iOS and for US region only

(with separate possibility to sign up for beta testing, also not for people outside US, and the app is in the official store already anyway, but it was before official release - at the end of June, so it's older news).

It's a multifunctional app for audio editing, which also contains a separation model.

It's free, called:

"Ripple - Music Creation Tool"

<https://apps.apple.com/us/app/ripple-music-creation-tool/id6447522624>

The app requires iOS 14.1

(it's only for iOS).

Output files are 4 stems 256kbps M4A (320 max).

Currently, the best [SDR](#) for public model/AI, but it gives the best results for vocals in general. For instrumentals, it rather doesn't beat paid Dango.ai (and rather not KaraFan too).

"My only thought is trying an iOS Emulator, but every single free one I've tried isn't far-fetched where you can actually download apps, or import files that is"

Sideloaded of this mobile iOS app is possible on at least M1 Macs.

"If you're desperate, you can rent an M1 Mac on Scaleway and run the app through that for \$0.11 an hour using this <https://github.com/PlayCover/PlayCover>"

IPA file:

https://www.dropbox.com/s/z766tfysix5gt04/com.ripple.ios.appstore_1.9.1_und3fined.ipa?dl=0

"been working like a dream for me on an M1 Pro... I've separated 20+ songs in the last hour"

"bitrise.com claims to have M1s and has a free trial"

Scaleway method:

<https://cdn.discordapp.com/attachments/708579735583588366/1146136170342920302/image.png>

"keep in mind that the vm has to be up for 24 hours before you can remove it, so it'll be a couple bucks in total to use it"

"I used decrypted ipa + sideloaded
seems that it doesn't have internet access or something"

So far, Ripple didn't beat voc_ft (although there might be cases when it's better) and Dango. Samples we got months ago are very similar to those from the app, also *.models files have SAMI header and MSS in model files (which use their own encryption), although processing is probably fully reliable on external servers as the app doesn't work offline (also model files are suspiciously small - few megabytes, although it's specific for mobilenet models). It's probably not the final iteration of their model, as they allegedly told someone they were afraid that their model will leak, but better than the first iteration judging by SDR with even lossy input files.

Later they told that it's different model than the one they previously evaluated, and that time it was trained with lossy 128kbps files due to some "copyright issues".

Most importantly, it's the good for vocals, also cleaning vocal inverts, and surprisingly good for e.g. Christmas songs, (it handled hip-hop, e.g. Drake pretty well). It's better for vocals than instrumentals due to residues in other stem - bass is "so" good, drums also decent. Vocals can be used for inversion to get instrumentals, and it may sound clean, but rather not as good as what 2 stem option or 3 stem mixdown gives.

Other stem residues appear due to the fact they told the other stem is taken from the difference of all remaining stems - they didn't train the other stem model to save on separation time.

"One thing you will notice is that in the Strings & Other stem there is a good chunk of residue/bleed from the other stems, the drum/vocal/bass stems all have very little to no residue/bleed" doesn't exist in all songs.

It's fully server-based, so they may be afraid of heavy traffic publishing the app worldwide, and it's not certain that it will happen.

Thanks to Jorashii, Chris, Cyclcrlclicly, anvuew and Bas.

Press information:

<https://twitter.com/AppAdsai/status/1675692821603549187/photo/1>

<https://techcrunch.com/2023/06/30/tiktok-parent-bytedance-launches-music-creation-audio-editing-app/>

Beta testing

<https://www.ripple.club/>

- Following models added on MVSep:

UVR-De-Echo-Aggressive

UVR-De-Echo-Normal

UVR-DeNoise

UVR-DeEcho-DeReverb

They are all available under the "Ultimate Vocal Remover HQ (vocals, music)" option (MDX FoxJoy MDX Reverb Removal model is available as a separate category).

- If you looked for possibility to pay for Dango using Alipay - they recently introduced the possibility to link foreign cards, and if that option fails (sometimes does), you can open 6 months "tourcard", and open new later if necessary, but only Visa, Mastercard, Diners Club and JCB cards are supported to top tourcard up

<https://ltl-beijing.com/alipay-for-foreigners/>

- Dango no longer supports Gmail email accounts

- New piano model added on MVSEP. SDR-wise it's better than GSep, but GSep is probably also using some kind of processing in order to get better separation results, but e.g. Dango

instrumentals can be inverted to get just vocals despite the fact they claim to use some recovery technology.

- [arigato78 method](#) for main vocals

- Captain Curvy method for instrumentals added in instrumentals models list section (the top link)

- For canceling room reverb check out:

Reverb HQ

then

De-echo model (J2)

- Sometimes vox_ft can pick up SFX

- Install UVR5 GUI only in the default location picked by the installer. Otherwise, you might get python39.dll error on startup. If you see that error after installing the beta patch, reinstall the whole app.

- Few of our users finally evaluated sonically new dango.ai 9.0 models. Turns out the models are not UVR's (or no longer), and actually give pretty close results to original instrumentals, but not so good vocals.

"It's slightly better but still voc_ft keeps more reverb/delays

but again, it's 99% close, Dango has maybe more noise reduction" maybe even less instrumental residues (can be a result of noise reduction).

"A bit cleaner than voc_ft in terms of having synths/instruments, but they do sound a bit filtered at times. [In] overall it's close tho"

"I discovered Dango's conservative mode keeps instrumentals even fuller, but might introduce some background vocals
still quite better than what we have.

I'm still surprised how it's so clean, as if not having vocal residues like any other MDX model.
Sometimes the Dango sounds like a blend of VR's architecture, but I'm probably wrong, it could be the recovery technology" - becruily

<https://tuanzai.com/vocal-remover/upload>

You must use the built-in site translate option in e.g. Google Chrome, because it's Chinese.
On Android, it may not work correctly. In case of further issues, use Google Translate or one of Yandex apps with image to text translators.

You are able to pay for it using Alipay outside China.

Dango redirects to Tuanzai site - it's the same.

<https://tuanzai.com/encouragement>

Here you might get 30 free points (for 2 samples) and 60 paid points (for 1 full songs)
"easily".

Dango.ai scores bad in SDR leaderboards due to recovery algorithms applied. Similar situation probably like in GSep.

- New BVE model on X-Minus for premium users. One of, if not the best so far. It uses `voc_ft` as a preprocessor.

"BVE sounds good for now but being an (u)vr model the vocals are soft (it doesn't extract hard sounds like K, T, S etc. very well)"

"Pretty good, if still [in] training. Seems to begin a phrase with a bit of confusion between lead and backing, but then kicks in with better separation later in the phrase. Might just be the sample I used, though."

- Jarredou published the final [2.2 version](#) of MDX23 Colab (don't confuse it with MDX23C single models v3 arch) - gives more vocal residues than 2.0/2.1, but better SDR. Now it has SRS trick, bigshifts, new fine-tuning, separated overlap parameters for MDX, MDXv3 and Demucs models, and also possess one narrowband MDX23C model D1581 among other MDX ones, which states a new set of models now (also said to use VOC-FT Fullband SRS instead of UVR-MDX-Instr-HQ3, although HQ3 is still listed during processing). You can also use faster optional 2 stem only output (demucs_ft vocal stem is used here only). Float parameter returns WAV 32-bit. Don't set overlap v3 to more than 10, or you'll get error. It can be way more frequent with odd values.

Changing weights added: "For residues, I would first try a different weight balance, with less MDXv3 and more VOC-FT, as model D1581, and current MDXv3 models in general tend to have more residues than VOC-FT."

- New "8K FFT full band" model published on MVSEP. Currently, a better score than only 2.2 Colab above from commonly available solutions, although more vocal residues than current default on MVSEP at least in some cases, and "voice sounded more natural [in default] than the new 10 SDR model" but in some problematic songs it can even give the best results so far.

"Sometimes 8K FFT model is false detect the vocals, in the vocal stem synth was treated as vocal. On instrumental stem, mostly are blur result compared with 12K FFT. But 12K FFT seems to be some vocal residue but very less heard (like a whisper) and happened for several songs, not all songs."

- "The karaoke ensemble works best with isolated vocals rather than the full track itself"
Kashi

- Center isolation method further explained in *Tips to enhance separation, step 19*

- VR Kara models freeze on files over ~6 minutes in UVR beta 2 (GTX 1080).
>Divide your song into two parts.

- New public dataset published by Moises ([MoisesDB](#)). There are some problems with downloading it now, and it's 82,7GB and link expires during downloading after 600 seconds. Not enough for 30MB/s, but good for 10Gbps one. Moises team works on the issue. Probably it's fixed already.
- RipX inside the app uses UVR for gathering stems now. Consider also comparing its stem cleanup feature to RX 10 debleed in RX Editor.
- "RipX is badass for removing residues and harmonics from vocals. The ability to remove harmonics & BGVs using RipX is amazing but is very tedious but so far so good" (Kashi)
- Sometimes using vocal model like voc_ft on the result from instrumental model might give less vocal residues or sometimes even none (Henry)
- mvsep1.ru from now on, contains a content of mvsep.com, so without MDX23/C and login features, while mvsep.com has the richer content of mvsep1.ru
The old leaderboard link has changed and is now:
https://mvsep1.ru/quality_checker/leaderboard2.php?sort=instrum
- old domain is also fixed now, redirecting leaderboard links.
If you're uploading in quality checker is stopped, clear your browser and start over.
- Dereverb and denoiser for VR arch is not compatible with any VR Colab and manual installation of such model will fail with errors. It requires modifying nets and layers. [More](#)
- New best ensemble (all Avg/Avg)
(read entries details on the [chart](#) for settings - they can have very time-consuming parameters and differ in that aspect)
#1 MDX23C_D1581 + Voc FT | #2 MDX23C_D1581 + Inst HQ3 + Voc FT | #3
MDX23C_D1581 + Inst HQ3 + Voc FT

Be aware that above can sound noisy/have vocal leaks at times; consider using HQ_3 or kim inst then, also:

- The best ensembles so far in Kashi's testing for general use:
Kim Vocal 2 + Kim FT other + Inst Main + 406 + 427 + htdemucs_ft avg/avg, or:
Voc FT, inst HQ3, and Kim FT other (kim inst)
"This one's much faster than the first ensemble and sometimes produces better results"

It all depends on a song. Also, sometimes "running one model after another in the right order can yield much better results than ensembling them".

- Disable "stem combining" for vocal inverted against the source. Might be less muddy, possibly better SDR.

It's there in MDX23C because now the new arch supports multiple stems separation in one model file.

- Disabling "match freq cutoff" in advanced MDX settings seems to fix issues with 10kHz cutoff in vocals of HQ3 model.
- New explanations on Demucs parameters added in Demucs 4 section
(shifts 0, overlap 0.99 won in SDR vs shifts 1, overlap 0.99 and even shifts 10, overlap 0.95)

- "Last update of Neutone VST plugin has now a Demucs model to use in realtime in a DAW (it's a 'light' version of Demucs_mmi)

<https://neutone.space/models/1a36cd599cd0c44ec7ccb63e77fe8efc/>

It doesn't use GPU, and it's configured to be fast with very low parameters, also the model is not the best on its own. It doesn't give decent results, so it's better to stick to other realtime alternatives (see document outline)

- Turns out that with a GPU with lots of VRAM e.g. 24GB, you can run two instances of UVR, so the processing will be faster. You only need to use 4096 segmentation instead of 8192.

SDR difference between overlap 0.95 and 0.99 for voc_ft MDX model in (new/beta) UVR is 0.02.

0.8 seems to be the best point for ensembles

12K segmentation performed worse than 4K SDR-wise

- Recommended balanced values between quality and time for 6GB graphic cards in the latest beta:

VR Architecture:

Window Size: 320

MDX-Net:

Segment Size: 2752 (1024 if it's taking too long)

Overlap: 0.7-/0.8

Demucs:

Segment: Default

Shifts: 2 (def)

Overlap: 0.5

(exp. 0.75,

def. 0.25)

"Overlap can reduce/remove artifacts at audio chunks/segments boundaries, and improve a little bit the results the same way the shift trick works (merging multiple passes with slightly different results, each with good and bad).

But it can't fix the model flaws or change its characteristics"

"Best SDR is a hair more SDR and a shitload of more time.

In case of Voc_FT it's more nuanced... there it seems to make a substantial difference SDR-wise.

The question is: how long do u wanna wait vs. quality (SDR-based quality, tho)"

- A script with guide for [separating multiple speakers](#) in a recording added
- If you're stuck at 5% of separation in UVR beta, try to divide your audio into smaller pieces (that's beta's regression)
- A new separation site appeared, giving seemingly better results than Audioshake:
<https://stemz.mwm.io/>
"Guitar stem seems better than Demucs, piano maybe too. Drums sound like Spleeter. Vocal bleeds in most of the stems, or not vocals are picked up, so they end up in the synths. But that's just from one song test" becruily
- Drumsep [Colab](#) now has GPU acceleration and much better max quality optional settings
- 1620 MDX23C model added on x-minus. Opposing the model on UVR, it's fullband and not released yet (16.2 SDR).

"Even if the separations have more bleeding than VOC-FT (and it's an issue), the voice sound itself is much fuller, "in your face" compared to VOC-FT, that I now find it like blurry sounding compared to MDXv3 models.

I think that's why the new MDXv3 models are scoring better despite having more bleeding (at the moment, like I said before, trainers/finetuners have to get familiar with new arch, and that will probably help with that new bleed issue)."

- New MDX23C model added on MVSEP (better SDR - 16.17)
- UVR beta [patch 2](#) repairing no audio issue with GPU separation on the GTX 1600 series using MDX23C arch. Fixes some other bugs too.
- Narrowband MDX23C vocal model (MDX23C_D1581 a.k.a. model_2_stem_061321) trained by UVR team has been released. SDR is said to be better than voc_ft (but the latter was evaluated with older non-beta patch). Be aware that CPU processing returns errors for MDX23C models, at least on some configs ("deserialize model on CUDA" error). Fullband

models will be released in a few weeks (and as it was usually before, on x-minus first for a few weeks later). [Download](#) (install [beta patch](#) first and drop it into the MDX-Net models folder). The patch is for only Windows now, with an upcoming Mac patch planned later. For Linux, there's probably a source of the patch already out.

MDX23C_D1581 parameters are set up with its yaml config file and its n_fft value is 12288, not 7680. It has cutoff at 14.7khz (while VOC-FT cutoff is 17.5khz)

- "(Probably all) models are stereo and can't handle mono audio. You have to create a fake stereo file with the same audio content on the L and R channel if the software doesn't make it by itself." Make sure that the other channel is not empty when isolation is executed - it can produce silent bleeding of vocals in the opposite channel (happens in e.g. MDX23 and GSEP, and errors with mono in MDX-Net)
- "For Unbound local" error while you do anything in UVR since the new model installation, you might be forced to rollback the update
- Clear the Auto-Set Cache in the MDX-Net menu if you set wrong parameter and end up with error
- Pitch shift is the same as soprano mode except in the GUI beta you can choose how many semitones to pitch the conversion
- Dango.ai released a 9.0 model. We received a very positive report on it so far.
- UVR beta patch released. Potentially new SDR increases with the same models.
Added segmentation, overlap for MDX models, batch mode changes.
Soprano trick added. Basically, you can set it by semi-tones.
Support for MDX-NET23 arch. For now, it uses only basic models attached by Kuielab (low SDR, so don't bother for now), but UVR team already trained their own model for that arch, which will be released later, and a few weeks after x-minus and MVSep. And it's performing well already. Wait™. Don't exceed an overlap 0.93-0.95 for MDX models, it's getting tremendously long with not much of a difference, 0.8 might be a good choice as well. Also, segments can ditch the performance AF. 2560 might be still a high but balanced value.
Sadly, it looks like max mag for single models is no longer available - you can use it only under Ensemble Mode for now.

Q: What is Demucs Pre-process model?

A: You can process the input with another model that could do a better job at removing vocals for it to separate into the other 3 stems

Beta UVR patch [link](#)

- "Post-Process [for VR] has been fixed, the very end bits of vocals don't bleed anymore no matter which threshold value is used"

- New BVE model will be ready at the beginning of August (Aufr33).
- MDX23C by ZFTurbo model(s) added on mvsep.com. They're trained by him using the new 2023 MDX-Net V3 arch.
Slightly worse SDR than MDX23 2.1 Colab on its own.
Might be good for rock, the best when all three models are weighted/ensembled)
- MDX23C ensemble/weighted available on mvsep.com for premium users (best SDR for public 2 stem model).
It might still leave some instrumental residues in vocals of some tracks (which can be cleared with MDX-UVR HQ_3 model) but it can be also vice versa - the same issue as kim vocal models, where the vocals are slightly left in the instrumentals [vs e.g. MDX23 2.1 free of the issue]
On some Modern Talking and CC tracks it can give the best results so far).
- If you have problems with "Error when uploading file" on MVSEP, use VPN. Similar issues can happen for free X-Minus for users in Turkey.
- lalal.ai cooperation with MVSEP was fake news. Go along.
- As for Drumsep, besides in fixed [Colab](#), you can also use it (the separation of single percussion instruments from drums stem) in UVR GUI. How to do this:
Go to UVR settings and open the application directory.
Find the folder "models" and go to "demucs models" then "v3_v4"
Copy and paste both the [.th](#) and [.yaml](#) files, and it's good to go.
Overlap above 0.6 or 0.7 becomes placebo, at least for dry track, with no effects.
- Drumsep benefits from shifts a lot (you can use even 20).
- For better results, test out potentially also -6 semitones in UVR beta, or with 31183Hz sample rate with changed tempo.
12 semitones from 44100Hz is 22050 and should be rather less usable in most cases, the same for tempo preservation on.
- If you have a long band_net [error](#) log while using DeNoise model by Fox Joy in UVR, reinstall the app.
- It can happen that every second separation using MDX Colab will fail due to memory issues, at least with Karaoke 2 model.
- New fine-tuned vocal model added to UVR5 GUI download center and HV Colab (slightly better SDR than Kim Vocal 2) it's called "UVR-MDX-Net-Voc_FT" and is narrowband (because it's based on previous models).

- Audioshake 3 stem model is added to <https://myxt.com/> for free demo accounts. Unfortunately, it has WAVs with 16kHz cutoff which Audioshake normally doesn't have. No other stem. Results, maybe slightly better than Demucs. Might be good for vocals.

- Spectralayers 10 received an update of an AI, and they no longer use Spleeter, but Demucs 4, and they now also good kick, snare, cymbals separation too. Good opinions so far. Compared to drumsep sometimes it's better, sometimes it's not. Versus MDX23 Colab V2, instrumentals sometimes sound much worse. "SpectraLayers is great for taking Stems from UVR and then carrying on separating further and editing down. (...) Receives a GPU processing patch soon"

- (?) some) MDX Colabs started causing errors of insufficient driver version.

> "As a temp workaround you can go to "Tools" in the main menu, and "Command Palette", and search for "Use fallback runtime version", and click on it, this will restart the notebook with the previous Ubuntu version in Colab, and things should work as they were before (at least till mid July or earlier [how it was once] where it is currently scheduled to be deleted)" probably it will be fixed.

X: Some people have an error that fallback runtime is unavailable.

- New v2 version of **ZFTurbo's MDX23** [Colab](#) released by jarreadou (now also with denoiser off memory fix added). Now it should have less bleeding in general.

It includes models changed for better ones (Kim Vocal 2 and HQ_3), volume compensation, fullband of vocals, higher frequency bleeding fix. It all manifests in increased SDR.

Instrum is inverted of vocals stem

Instrum2 is the sum of drums+bass+other stems (I used to prefer it, but most people rarely see any difference between both, and it also depends on specific fragments, although instrum gets better SDR and is less muddy, so it's rather better to stick with instrum)

If your separation ends up instantly with path written below, you wrongly wrote it in the cell. Simply remove the `file - name.flac` at the end and leave only path leading to a file.

It's organized in a way that it catches all files within that path/folder.

Suggestion: go to drive.google.com and create a folder `input`, and drop the tracks you want to process in there.

When the process is done, delete them, and add others you want to process.

Overlap large and small are the main settings, higher values = slightly higher score, but way longer processing.

Colab doesn't allow much higher value for chunk size, but you can try little higher ones and see when it crashes because of memory. Higher chunk size give better results.

- [Updated](#) inference with voc_ft model ([Colab](#) v2.1 has denoiser now on, but updated inference not and is essentially what 2.2 currently is).

- *Volume compensation fine-tuning - it is in line 359 (voc_ft), 388 (for ensembling the vocals stem), 394 (for HQ_3 instrumental stem inversion).*

- chunk_size = 500000 will fail with 5:30 track, decrease it to at least 300K in such case. Overlap 0.8 is a good balance between duration and quality.
 - In case of system error wav not found, simply retry separation.
- Nice [instruction](#) how to use the Colab.
The v. 2.1 Colab was firstly evaluated with lower parameters, hence it received slightly worse SDR. Then it was evaluated again and got better score than v2.

WiP Colabs

- 2.2 Beta [1](#) (no voc_ft yet)
- 2.2 Beta [1.5](#)
- 2.2 Beta (1.5.1, [inference](#) with voc_ft, replace in the Colab above; no fine-tuning)
- [v2.2](#) beta 2/3 ([working inference](#)) (MDX bigshifts, overlap added, fine-tuning, no 4 stems > experimental, no support for now, 22 minutes for vocals only, mdx: bsf 21, ov 0.15, 500k, 5:30 track)
- [v2.2](#) (w/ voc_ft [inference](#)) pre beta 3 w/o MDX v3 yet - comment out both bigshifts in the cell - they won't work

- current beta [link](#) (WiP, might be unstable at times; e.g. here for 19.07 bigshifts doesn't work, and you need to look for working inference in history or delete the two bigshifts references in the cell; doesn't seem that MDX v3 model is here yet)

In general -

MDX23 is quite an improvement over htdemucs_ft (...).
Drum stem makes htdemucs_ft sound like lossy in comparison, absolutely beautiful
Bass is significantly more accurate, identifies and retains actual bass guitar frequencies with clarity and accuracy
"Other", equally impressive improvement over htdemucs_ft, much more clarity in guitars"
And problems with vocals they originally described are probably fixed in V2 Colab.

- "I just added 2 new denoise models that were made by FoxJoy. They are both very good at removing any residual noise left by MDX-Net models. You can find them both in the "Download Center". - Anjok
Be aware that they're narrowband (17.7kHz cutoff). Good results.

To download models from Download Center -

In UVR5 GUI, click the tools icon > click Download Center tab > Click radio button of VR architecture > click dropdown > select the model > hit Download button > wait for it to download... Profit.

- New MDX-UVR "HQ_3" model released in UVR5 GUI! The best SDR for a single instrumental model so far. [Model file](#) (but visiting download center is enough). On X-Minus I think too.

-HQ_3 model added to [MDX Colab](#) (old)

-HV just made a new version of her own [updated MDX Colab](#) with all new models, including HQ_3. It lacks e.g. Demucs 2 for Instrumentals of vocal models, but in return it allows using YouTube and Deezer links for lossless tracks, with providing ARL, and allows specifying manually more than one file name to process at the same time. Also, for any new models in the future, there's optional input for model settings, to bypass parameters of parameters autoloader. IRC, the Colab stores its files in different path, so be aware about it when uploading tracks for separations on GDrive.

- she has added volume compensation in new revision (they're applied automatically for each model)

In previous [MDX Colabs](#) there were also min, avg, max, and chunks, but they're gone in HV Colab.

- HV also made a [new](#) VR Colab which irc, now don't clutter all your GDrive, but only downloads models which you use (but without VR ensemble) and probably might work without GDrive mounting, but it lacks VR ensemble.

- New MDX models added to both variants of MVSep (Kim inst, Vocal 1/2, Main [vocal model], HQ_2)

- ZFTurbo's MDX23 code now requires less GPU memory. "I was able to process file on 8 GB card. Now it's default mode.": 6GB VRAM is not enough. Lowering overlaps (e.g. 500000 instead of 1000000) or chunking track manually might be necessary in this case. Also now you can control everything from options: so you can set chunk_size 200000 and single ONNX. It can possibly work with 6GB VRAM that way.

Overlap large and small - controls overlap of song during processing. The larger value the slower processing but better quality (both)

If you have fail to allocate memory error, use --large_gpu parameter

Sometimes turning off use large GPU and reducing chunk size from 1000000 to 500000 helps

- Models/AIs of the 1st and 2nd place winners in MDX23 music challenge (ByteDance's and quickpepper947's) sadly won't be released to the public (at least won't be open-sourced).

Maybe in June, ByteDance will be released as an app in worse quality.

Judging by the few snippets we had:

"the vocal output, yes, better than what can be achieved right now by any other model, it seems.

the instrumental output... meh. I can hear vocals in it, on a low volume level." but be aware that improved their model by the time by a lot.

- MDX23 4 stem model and [source code](#) with dedicated app by ZFTurbo (3rd place) was released publicly with the whole AI and instructions how to run it locally. No longer requires minimum 16GB VRAM Nvidia GPU. It even has a neat GUI (3rd place in leaderboard C, better SDR than demucs ft). You can still use the model online on [mvsep1.ru](#) (now mvsep.com).

The command:

"conda install -c intel icc_rt"
SOLVES the LLVM ERROR

For above, you can get less vocal residues by replacing the Kim Vocal 1 model there manually by newer Kim Vocal 2 and kim inst by and Kim Inst with UVR Inst HQ 292 ("full 292 is a lot more aggressive than kim_inst").
jarreadou forked it with better models and settings already.

Short technical [summary](#) of ZFTurbo about what is under the hood and small [paper](#).
From what I see in the code, it uses inverted vocals output for instrumentals from - Demucs ft, with - hdemucs_mmi, and - Kim vocal 1 and - Kim inst (ft other). More explanations in [MDX23](#) dedicated section of this doc.

- jarreadou made a [Colab](#) version of ZFTurbo MDX23:
"(It's working with `chunk_size = 500000` as default, no memory error at this value after few tests with Colab free)

Output files are saved on Colab drive, in the "results" folder inside MVSep installation folder, not in *your* GDrive."

On 19.05 its SDR was tested, and had better score for instrumentals than UVR5 ensemble for that time being. Currently not, but there are new versions of the Colab planned.

- ByteDance-USS was released with [Colab](#) by jazzpear. It works better than zero-shot for SFX and "user-friendly wise" while zero-shot stil better for instruments.

<https://www.dropbox.com/sh/fel3hung4eb83rs/AAA1WoK3d85W4S4N5HObxhQGa?dl=0>
Queries for ByteDance USS taken from the DNR dataset. Just DL and put these on your drive to use them in the Colab as queries."

[QA](#) section added.

- The [modified](#) MDX Colab - now with automatic models downloading (no more manual GDrive models installations) and Karaoke 2 model.

> Separate input for 3 models parameters added, so you don't need to change models.py every time you switch to some other model. Settings for all models listed in Colab. From now on, it uses reworked main.py and models.py (made by jarreadou) downloaded automatically.

Don't replace models.py from packages with models from [here](#) now. Now denoiser optionally added!

- MDX Colab with newer models is now reworked to use with current Python 3.10 runtime which all Colabs now use.

- Since 28.04 lots of Colabs started having errors like "onnxruntime module not found".

Probably only MDX Colab (was) affected.

(not needed anymore)

> "As a temp workaround you can go to "Tools" in the main menu, and "Command Palette", and search for "Use fallback runtime version", and click on it, this will restart the notebook with the previous python version, and things should work as they were before"

- [OG](#) MDX HV Colab is (also) broken due to torch related issues (reported to HV). To fix it, add new code row with:

`!pip install torch==1.13.1`

below mounting and execute it after mounting

> or use fixed MDX [Colab](#) with newer models and fix added (now with also old Karaoke models).

- While using [OG](#) HV VR Colab, people are currently encountering issues related to **librosa**. The issues are already reported to HV (the author of the Colab).

> use this [fixed](#) VR Colab for now (04.04.23). (the issue itself was fixed by uncommenting librosa line and setting 0.9.1 version - deleted "#" before the lines in Mount to Drive cell, now also fresh installation issues are fixed - probably the previous fix was based on too old HV Colab revision). VR Colab is not affected by May/April runtime issues.

- If you have fast CPU, consider using it for ensemble if you have only 4GB VRAM, otherwise you can encounter more vocal residues in instrumentals. 11GB VRAM is good enough, maybe even 8GB.

- New Kim's instrumental "ft other" model. Already added to UVR's download center with parameters.

Manual settings - dim_f = 3072 n_fft = 7680

<https://drive.google.com/drive/folders/19-jUNQJwls7UyuWO5PWWVUIJQEwpn78>

(Unlike HQ models, it has cutoff, but better SDR than even inst3/464, added to Colab)

- Anjok (UVR5) "I released an additional HQ model to the Download Center today.

UVR-MDX-NET Inst HQ 2 (epoch 498) is better at removing long drawn out vocals than UVR-MDX-NET Inst HQ 1." It has already evaluated slightly better SDR vs HQ_1 both for vocals and instrumentals (HQ_1 evaluation was made once more since introducing Batch

Mode which slightly decreases SDR for only single models vs previous versions incl. beta, but mitigates an issue when there are sudden vocal pop-ins using <11GB VRAM cards)

- Anjok (UVR5, non-beta) "So I fixed MDX-Net to always use **Batch Mode**, even when chunks are on. This means setting the chunk and margin size will solely be for audio output quality. Regardless of PC specs, users will be able to set any chunk or margin size they wish. Resource usage for MDX-Net will solely depend on Batch Size."

Edit. Batch size set to default instead of chunks enabled on 11GB cards for ensemble achieves better SDR, but separation time is longer.

- Public UVR5 patch with batch mode and final **full band** model was released ([MDX HQ_1](#))

- 293/403 and 450/498 (HQ_1 and 2) full band MDX-UVR models added to [Colab](#) and (also in UVR) (PyTorch fix added for Colab)

- **Wind** model (trumpet, sax) beside x-minus, added also to UVR5 GUI
You'll find it in UVR5 in Download Center -> VR Models -> select model 17
(10 seconds of audio separated with Wind model, from a 7-min track, takes 29 minutes to isolate on a 3rd gen i7 - might be your last resort if it crashes your 4GB VRAM GPU as some people reported)

- (x-minus/Aufr33) "1. **Batch mode** is now enabled. This greatly speeds up processing without degrading quality.

2. The **b.v.** models have been renamed to **kar**.

3. A new **Soprano voice** setting has been added for songs with the high-pitched vocals.
This only works with mdx models so far."

It slows down the input file similarly to the method we described in our tip section below.

- New MDX23 vocal model added to beta MVSEP site.

- (no longer necessary) [Fork of UVR GUI](#) and [How to install](#) - support for AMD and Intel GPUs appeared (works only for VR and MDX architectures), Besides W11, also W10 confirmed working, MDX achieves speeds of i5-4460s using 6700 XT, while for VR, speeds are v. fast and comparable to CUDA, so CPU processing might be slower in VR, but for MDX you might want to stick with the official UVR5 GUI.

- *Batch mode seems to fix problems with vocal popping using low chunks values in MDX models, and also enhance separation quality while eliminating lots of out of memory issues. It decreases SDR very slightly for single models, and increases SDR in ensemble.*

- (outdated) New beta MDX model "Inst_full_292" without 14.7kHz cutoff released (performs better than Demucs 4 ft). If the model didn't appear on your list in UVR 5 GUI, make sure

you've redeemed your code

<https://www.buymeacoffee.com/uvr5/vip-model-download-instructions>

Or use [Colab](#).

Newer epochs available for [paid](#) users of <https://x-minus.pro/ai?hp&test-mdx>

- To use Colabs in mobile browsers, you probably no longer need to switch your browser to PC Mode first.

News section continues in [older](#) news/update logs

General reading advice

- If you found this document elsewhere (e.g. as PDF), [here](#) is always up-to-date version of the doc

- If you have anything to add to this doc, ping me @deton24 on our Discord [server](#) from the footer, but rather refrain from PMing directly if not necessary. Every time you request writing privileges via GDoc, God kills a cat. Don't click "ask for privileges"!

- You can use the (rather outdated) [Table of content](#) section, but better go to Options and show "**Document outline**" to see an up-to-date clickable table of content. If you don't have Google Docs installed, and you opened the doc in a mobile browser and no Table of content option appear, use [Table of content](#) or go to options of the mobile browser and run the site in PC mode to show document outline (but it's better to have Google Docs installed on your phone instead as it's more convenient in use).

- Sometimes you cannot scroll down the list of headings on the phone in PC mode. Then you need to tap on the scroll bar in the very left, but it might suddenly look buggy all highlighted, but working nevertheless.

- Downloaded .docx will have a similar document outline as in the GDoc (but more messy - with all headers used in the GDoc). If you have an error on attempt of opening the .docx file on Windows, go to RBM>Properties and check Unlock below Attributes.

- Be aware that the document can hang for a while on the attempt of accessing a specific section of the document - it doesn't happen often on a PC browser - it's the most stable form of reading the doc online. At least on a decent PC (so not C2Q, but even a decade old i7 is usually fine). But it can be stable on Android phone too (e.g. Snapdragon 700 series instead of old 400 series). Google's app support for old 32-bit ROMs in e.g. Android 9 and older is terrible.

- Search and navigating through the document outline works faster when you download the doc as .pdf or .docx, but in the latter you'll have access to the document outline on the left like in GDoc (if not, press CTRL+F>Headings, or check View>Navigation window).

- When visiting the online version of the doc, you can paste whole phrase when searching instead of single letters to avoid severe stuttering during using search function online.

- Use the search bar in Google Documents, not the one from the browser - the browser's search won't find everything unless all the pages were shown before - the doc is huge.

- Sometimes if you search for a specific keyword in the mobile app and the result doesn't show up, you need to go to the document outline, and open its last section and search again (so the whole document will be loaded first, otherwise you won't get all the search results in some cases). But it might happen mostly if you use the wrong search function.

- Make sure you've joined our Discord [server](#) to open some of the Discord links attached below (those without any file extension at the end).

- Download links from Discord with file extensions at the end no longer work, but I reuploaded most of the important links already. If you need to download from previously shared Discord link anyway:

1) Join our Discord server via invitation at the top of the document 2) Delete file name from the link 3) Open our Discord server in the browser 4) Leave everything in the link before the first slash and delete the rest (so channels\xxxx\ 5) paste two identifiers divided by slashes afterwards, but without file name (so channels\xxxx\xxxx\xxxx - where the last two are taken from inactive file link) *) If you paste offline link in any channel on the source server, the link will work again

- If you have a crash on opening the doc in the app, e.g. on Android - reset the app cache and data. Keep the app updates or find some old version (e.g. even from period when your phone was released or uninstall all updates if it's GDoc is your stock app

- If it loads 4 minutes/ininitely in the doc app, update your Google Docs app and reset the app cache/data, e.g. if you started to have crashes after the app update.

- You can share a specific section of this document by opening it on PC or on a mobile browser set in PC mode by clicking on one of the sections in the document outline (or hyperlinks leading to specific sections). Now it will add a reference to the section in the link in your address bar, which you can copy and paste, so opening this link will redirect someone straight to the section after opening the link (in some specific cases, some people won't be redirected, but in fact, you only need to wait a few seconds after the first page of the doc has been shown, and then the proper redirect starts). Some headers not referred to in the outline are also set in a way that when you click them, the address bar will change with a link leading to that specific section. Not all headers present in the doc are shown in the outline to preserve better readability.

- In the GDoc app sometimes you need to tap "wait" a few times when the app freezes. Afterwards, searching will start working all the time (at least till the next time). The doc is huge and the GDoc app on at least low-end Androids is cursed (desktop version on PC behaves the most stable, as long as decent phones). You've been warned.

- If you feel overwhelmed by the doc size, theoretically you can load the doc into Google Gemini or Google NotebookLM and ask questions from there, but I encourage befriending with the document outline and the content of an interesting section yourself - asking the AI chat for the best models leads to hallucinating of the model and providing list of outdated separation models from this doc. Also, they all miserably fail with generating model and config links from even cut fragments of the GDoc. They also cannot edit the document directly (unless you paste the text, but it will rather delete all the formatting and hyperlinks which are essential to the task), maybe Office 365 with paid subscription is capable of editing docs with AI directly already.

- Descriptions on the list of models are usually shortened compared to the news section information added when the model was released. You can use search with the model name for more possible descriptions.
- Published audio demos of models pasted from MVSEP get online after a while, so most links to audio files from there will be offline after a week or more.
- Without GDoc app installed, or when in not PC mode of the browser, if you open links to this document ending with e.g. "#heading=h.hk34hc4d1ah7" or similar, you won't be redirected to a specific section of this document referred to in the heading. Redirections from such links doesn't work in mobile version of the GDoc site.
To be redirected after a moment to proper section from outside links with "heading" in the URL, you should open these links with GDoc app installed, or on PC browser, or mobile browser with PC Mode turned on (in Chrome that option appears when you open a page already).
- Sometimes when you click on an entry in the document outline, it might not react the first time straight up after you load the document. Most likely it's still loading and you need to click it twice or more and then you'll be redirected.
- Even on a powerful phone with lots of RAM, GDoc app can occasionally crash, esp. while browsing it before it's fully loaded, and even deleting the app and reopening it won't help.
- If you click on any hotlink redirecting to a specific part of this document from the mobile version of GDoc in the browser, you won't be able to show options to display the document outline after switching your browser into PC mode - it will remain in the mobile layout. It's because redirections in mobile versions have their own linking scheme adding to the site address - you need to delete its ending or reopen the doc.
- Besides me, jarredou (Discord: rigo2) and dca100fb8, currently no one else has writing privileges to this document, although they're reluctant to be active editors, they were granted the privileges as the last resort for possible cases of my longer absence in the future.

(I'm trying to keep the following list always updated with the Last updates/news section at the top)

Everyone asks **which service and/or model is the best** for **instrumentals, vocals or stems**. The answer is - we have listed a few models and services which behave the best in most cases, but the truth is - the result also strictly depends on the genre, specific song, and how aggressive and heavily processed vocals it has. Also, how much distortion instruments have, style of mixing, etc. Sometimes one specific album gets the best results with one specific tool/AI/model, but there might be some exceptions for specific songs, so just feel free to experiment with each, to get the best result possible using various models, ensembles and services/AIs from those listed. SDR on MVSEP doesn't always reflect bleeding well. That's why we introduced bleedless and fullness metrics for evaluation of the models as well. You'll read more about it in [this](#) section.

"Some people don't realize that if you want something to sound as clean as possible, you'll have to work for it. Making an instrumental/acapella sounding good takes time and effort. It's not something that can be rushed. Think of it like (...) love to a woman. You wouldn't want to just rush through it, would you? Running your song through different models/algorithms, then manually filtering, EQing, noise/bleed removing the rest is a start. You can't just run a song through one of these models and expect it to immediately sound like this" rAN

Sometimes you might want to combine results of specific models in specific song fragments. If the song is too muddy, you might want to use demudder in [newer](#) UVR patches and/or use some free [AIs](#) like AudioSR, Apollo or other, to further enhance the result. If you're still not happy, you might want to manually mix separated song stems and/or master it using plugins or AI mastering services (more about it [here](#)).

Sometimes you might be capable of creating a loop out of the fade ins/outs/intros/outros so you could totally refrain from using AI separation in the key fragments of the song, so you could just only use AI as reference to arrange the song as it was, and only fill missing fragments with separation.

A good starting point is to have [a lossless song](#) (the result will be a bit less muddy after separation).

Now, from free separation AIs/models, to get a decent instrumental/vocals, you can use the solutions below, starting from the models at top (every song might work differently with different models - find the best for your song - also, various headphones and speakers might be more or less sensitive to show you bleeding in song - lots of the time it will be imminent without phase fixer in songs with dense mix):

The best models for specific stems

*There's no such thing like the best model. It depends on a song, even in specific genre, mixing, effects, etc.
You need to test the best models posted at the top here, and see what fits the best for your song.*

- Most models here are Mel-Roformer and BS-Roformer model type in the compatible [UVR](#) version - not v2 model type (there's only one V2 model so far); don't confuse it with e.g. v2 versions/iterations of models below, which are just their names
- *Reading about [SDR](#) and [fullness](#) metric. [Evaluations](#) made on the multisong dataset on MVSEP. (table can be sorted by also fullness/bleedless and other metrics, once you open a result, fullness/bleedless metrics are shown too, excluding old results)*

2 stems:

- > for instrumentals ([click here](#) for vocal models)
- Model names starting with MVSEP can be used only on [MVSEP](#) (no download links available)

Good all-rounders from various categories (balanced, fullness, bleedless):

- Unwa [BS-Roformer Resurrection inst \(yaml\)](#) | a.k.a. "unwa high fullness inst" on MVSEP | [uvronline.app/x-minus.pro](#) | [Colab](#) | [Kaggle](#) | [UVR](#) (don't confuse with Resurrection vocals variant)

Inst. fullness: 34.93, bleedless: 40.14, SDR: 17.25

MVSEP BS 2025.07 works as a reference for phase fix with 3000/5000 settings.

Only 200MB. Some people might prefer it over V1e+, although it's more muddy.

"use if the others [below] are noisy"

Models working for phase fixer (to alleviate the noise) are only BS-Roformer 1296/1297 by viperx and BS Large V1 by unwa, but generally the model might require phase fixing less than other models here - dca

"One of my favorite fullness inst models ATM. Sounds like v1e to me, but cleaner. Especially with guitar/piano where v1e tended to add more phase distortion, I guess that's what you'd call it lol. This model preserves their purity better IMO" - Musicalman

"I like resurrection inst for segments of piano, a lot of other models are too noisy there (...) I also needed to turn overlap up for piano" (from 2 to 8). FNO was less noisy for it, but "the hit to fullness was extremely apparent" - rainboomdash

"The way it sounds, is indeed the best fullness model, it's like between v1e and v1e+, so not so noisy and full enough, though it creates problems with instruments gone in the instrumental sadly, but apparently it seems Roformer inst models will always have problems with instruments it seems, seems like a rule. (...) Instrument preservation (...) is between v1e and v1e+" - dca100fb8

"it seems to just nip some bits of random instruments like saxophone or guitar whereas v1e+ leaves them intact." - dennis777

"Some songs leaves vocal residue. It is heard little but felt" - Fabio

"Almost loses some sounds that v1e+ picks up just fine" - neoculture

Mushes some synths a bit in e.g. trap/drill tune compared to inst Mel-Roformers like INSTV7/Becruily/FVX/inst3, but the residues/vocal shells are a bit quieter, although the clarity is also decreased a bit. Kind of a trade.

BS 2025.07/BS 2024.04/BS 2024.08/SW removes less noise than viperx models for phase fixer.

- Unwa [BS-Roformer-HyperACE](#) | [separate Colab](#) (doesn't work in UVR)

Inst. fullness 36.91, bleedless 38.77, SDR 17.27

"sounding just like v1e+ after phase fix, but straight out of one single model

(...) quite bleedly, but honestly it's a fair price to pay, I guess" - santilli_

Although for some people it can be even on par with v1e+ bleed-wise, so check it out too (more fullness).

Note: It uses its own inference script. "You can use this model by replacing the [MSST](#) repository's models/bs_roformer.py with the repository's bs_roformer.py."

To not affect functionality of other BS-Roformer models by it, you can add it as new model_type by editing utils/settings.py and models/bs_roformer/init.py [here](#) (thx anvuel).

For error while installing py file for HyperACE model in Sacial's WebUI:

```
from models.bs_roformer.attend import Attend
```

```
ModuleNotFoundError: No module named 'models'
```

The fix: "SUC-DriverOld/MSST-WebUI use the name "modules" and ZFTurbo/Music-Source-Separation-Training use the name "models". And Unwa's bs_roformer.py that you replace with, also use "models". So you'll have to do some coding and symlink to make it work." - fjordfish

Metrically less fullness than v1e+: 37.89, but more bleedless: 36.53, SDR: 16.65 (v1e+).

While using locally, consider changing overlap from default 4 to 2 in the yaml of the model.

The difference won't be really noticeable for most people, but it will be faster.

"Currently, this model holds the highest aura_mrstft score on the instrumental side of the Multisong dataset. (...)

This weight is based on the following [weights](#). Thank you, anvuel!" - unwa

"Does seem like HyperACE is picking up more instruments than v1e+

does seem like slightly worse vocal bleed overall (still need to test this more, though)...

haven't encountered the super tinny vocal bleed like v1e+, at least

still fails to pick up that brass instrument on one song... Not really any worse than v1e+, though (...) resurrection inst does sound more muddy, but also a lot less noise.. which makes sense... IDK, a little muddy for my tastes.

I did find one song/spot and resurrection inst was on par with hyperace in picking up the wind instrument, v1e+ lost it for a bit.

I have found in the past that resurrection inst generally picks up more instruments than v1e+

(...) fullness of HyperACE is much closer to v1e+ than resurrection inst (...) it gets pretty

staticy compared to v1e+ [on some drums] (...) v1e+ does this to a lot less extent

it's not super common, though... (...) I'm very confident in saying bshyperace picks up more stuff than v1e+.

resurrection inst does pick it up much better than v1e+, but I think it's still too quiet

resurrection inst really does just pick up so much more instruments, despite having a lot less fullness" - rainboomdash

"fullness that is comparable to v1e+, but has significant more vocal crossbleeding in instrumental than BS Roformer Resurrection Inst, but still less than v1e+ and v1e" - dca100fb8

- Unwa Mel-Roformer V1e+ [model \(yaml\)](#) | [UVR \(guide\)](#) | [MVSEP](#) | [x-minus/uvronline](#) | [Colab](#) | [SESA](#) Colab | [Huggingface / 2](#) | [Kaggle](#)

Inst. fullness: 37.89, bleedless: 36.53, SDR: 16.65

*) [Phase fixer](#) Colab (e.g. with FT3 as src)/[UVR](#)>Tools, or on x-minus with bercuily vocal model used as reference model (premium) - for less noise.

*) introC [script](#) to get rid of vocal leakage in this model

*) "If you use Gabox Mel [denoise/debleed](#) model | [yaml](#) ([Colab](#)) on mixture then put the "denoised" inst stem of that into unwa inst v1e+ you get a very clean result with good fullness and very little noise" - 5b. But it can't remove vocal residues, just vocal noise. Might also sound interesting when using as target in Phase fixer, and with source set as Bercuily inst model (overlap 50/chunk_size 112455 was used; very slow - gustownis). Or bigbeta5e as a source to get rid of vocal residues - santilli_

(single model inference descriptions below)

"strange leakage [robot-like] in the vocal-only section with no instrumentation" - Unwa "less noise than v1e (probably due to different loss function), but it's also less full, "somewhere between v1e and v1"

"sometimes a detail piece of instrumental sound was lost, while on bercuily inst [below] can pick that sound". Might be too strong for MIDI sounds - kittykatkat_uwu.

Problems with broken lead-ins not happening in instv7 and v1e. Some issues with cymbals bleed in vocals - dca.

Better than v1+. "has fewer problems with quiet vocals in instrumentals than the V1+, "issues with harmonica, saxophone, electric guitar and synth seem to have been fixed" - dca100fb8. "has this faint pitched noise whenever vocals hit in dead silence, you may need to manually cut it out." - dynamic64. Check out also BS_ResurrectioN later below, it's like v1e++ (more fullness).

- Gabox [inst_fv4 \(yaml\)](#) | [Colab](#)

Inst. fullness 39.40, bleedless 33.49, SDR 16.44

Don't confuse it with inst_fv4noise - the regular variant was never released before (and with voc_fv4).

"Seems to be erasing a xylophone instrument. Does sound not too noisy and not muddy, I like it. (...) A little noisy with piano (I split the song up and process with resurrection inst there). (...) Does have some issues that resurrection inst doesn't have, but it doesn't sound muddy! It usually works great. (...) In my opinion, fv4 still has vocal traces, I don't know if in all of its songs and v1e plus doesn't have them, but the noise can bother you even though it's not much. Does have more vocal bleed at times. I think a lot of what I thought was vocal bleed was a synth, it did a pretty good job... There was one segment on a song where it caught vocal residues, though" - rainboomdash

- MVSep SCNet vocals model: SCNet XL IHF (high instrum fullness by bercuily).

Inst. fullness 32.31, inst. bleedless 38.15, SDR 17.20

“One of my favorite instrumental models, Roformer-like quality.

For busy songs it works great, for trap/acoustic etc. Roformer is better due to SCNet bleed” - becruily

“bring[s] such near perfect instrumentals”

vs the previous XL models “It's high fullness version for instrumental prepared by becruily.”

It can also be an insane vocal model too.

- Inst_GaboxFv8 v1 [model \(yaml\)](#)

Inst. fullness: 35.57, bleedless: 38.06, SDR: 16.51

The OG link to the model changed to the v2 variant of the model, but the old link to the v1 was retrieved above.

It has “v1+ metallic noise” - Gabox

VS V1e+ “A bit cleaner-sounding and has less filtering/watery artifacts. Both models are prone to very strange vocal leakage [“especially in the chorus”].

And because Fv8 can be so clean at times, the leakage can be fairly obvious. For now, my vote is for Fv8, but I'll still probably be switching back and forth a lot. Still has ringing” - Musicalman. Although, you might still prefer it over V1e+.

Might have some “ugly vocal residues” at times (Phil Collins - In The Air Tonight) - 00:46, 02:56 - dca.

“Sometimes V1e+ has vocal residues which sound like you were speaking through a fan/low quality mp3” - dca

“Seems to pick up some instruments better” Gabox.

- Gabox [Inst_FV8b \(yaml\)](#)

Inst. fullness: 35.05, bleedless: 36.90, SDR 16.59

If so, it's called V8 there (at least it's not INSTV8), maybe not fv8 v1.

Muddier than V1e+, but cleaner. Some people might prefer it over INSTV7.

“Preserves its volume stability to the original sound of the songs, it does not go down or lose strength, which is the most important thing, it manages to capture clear vocal chops, the voice is eliminated to 99 or 100% depending on its condition, it captures the entire instrumental and when making a mix it remains like the original that with other models the volume was lowered.” - Billie O'Connell

- Gabox experimental [Inst_FV8](#) | uvronline via special link ([free/premium](#))

Exclusive previous epoch of the FV8B model, previously only on uvronline.

“it's on the higher bleedless side” - Rainboomdash

Others

- Gabox [INSTV7 \(yaml\)](#) | [MVSEP](#) | [Colab](#) | [Huggingface / 2](#) | uvronline via special link for: [free/premium](#) | “F”, for fullness, “V” for version.

Inst. fullness: 33.22, bleedless: 40.71, SDR: 16.51

*) [Phase fixer](#) Colab/UVR's Phase Swapper (for less noise; e.g. with FT3 by Unwa vocal model as source).

"I hear less noise compared to v1e, but it has a worse bleedless metric" and might be less full.

It might still have too much noise like v1e for some people, but less.

"Relatively full/noisy model. Fvx [below] is a sort of middle ground between v3 and v7."

More fullness than V6, but vs v1e, sometimes "leaves noises throughout the song, sometimes vocal remnants in the verse of the song, and some instruments are erased."

Less muddy than Mel 2024.10 on MVSEP, and V7 doesn't preserve vocal chops/SFX.

- [Inst_GaboxFv8](#) v2 model ([yaml](#)) | [Colab](#)

Inst. fullness: 33.21, bleedless: 40.73, SDR: 16.57

Usually referred as just Inst_GaboxFv8 without v2. Since its release, the checkpoint has been updated on 11.05.25 (same file name), metrics have changed (updated above).

"v8 from uvronline and Fv8 from huggingface are completely different models" - maybe it's the v1 model. Also, don't confuse with INSTV8.

"Good result for bleedless instead, fullness went down instead of up a little."

Might be an interesting competitor to Unwa inst v2 which is muddier.

Inst. fullness: 35.57, bleedless: 38.06, SDR: 16.51 are the metrics of the old v1 model (unavailable). Unsure if uvronline uses the old fv8.

- Beccruily's inst | [Model files](#) | [Colab](#) | [Huggingface / 2](#) | on MVSEP a.k.a. Mel-Roformer "high fullness" | uvronline via special link for: [free/premium](#) (scroll down)

Inst. [fullness](#) 33.98, inst. bleedless 40.48, SDR 16.47

or on x-minus/uvronline (with optional phase correction feature in premium) | [UVR](#)

*) For less vocal residues use phase fixer [Colab](#) (also in UVR>Tools) and "beccruily's vocals as source and inst as target".

Alone, it's as clean as unwa's v1, but has less noise, and it can also be got rid well by:

*) Mel [denoise](#) and/or Roformer [bleed suppressor](#) by unwa/97chris. That model "removed some of the faint vocals that even the bleed suppressor didn't manage to filter out" before". Doesn't require phase fix. Try out denoising on a mixture first, then use the model.

On its own, the inst model correctly removes SFX voices. The instrumental model pulled out more adlibs than the released vocal model variant, when it can pull out nothing.

Currently, the only model capable of keeping vocal chops.

"Struggles a lot with low passed vocals"

More instruments correctly recognized as instruments and not vocals, although not as much as Mel 2024.10 & BS 2024.08 on MVSEP, but still more than unwa's inst v1e/v1/v2.

- If you use lower [dim_t](#) like 256 (or maybe also corresponding [chunk_size](#)) on weaker GPUs, these are the first Mel inst models to have muddy results with it.

- In the phase fixer you can experiment with "using beccruily's vocals as source and inst as target, and changing high frequency weight from 0.8 to 2 makes for impressive results" you can do it automatically after separation in this [Colab](#) (santilli_ suggestion).

Using Kim Mel FT2 as source instead might be more problematic as it tends to be more harmful to instruments, and in noise removal both are similar (dca).

- To demud the results from phase fixer, you can use Matchering and a well sounding fragment of single instrumental model separation with high fullness metric (e.g. 7N) as a reference and becruily inst/voc phase fixed result set as target (e.g. in UVR>Tools>Matchering). It will have less bleeding than models with low bleedless metric, but still fuller than phase-fixed results (more [here](#) and [here](#)). Phase fixer can also be used in a standalone Python [script](#) or in the [latest UVR](#). Matchering can be used in [Colab](#) or [songmastr](#) or [locally](#) (it's very lightweight and doesn't require a GPU).

Recent bleedless models

- Unwa [BS-Roformer-Inst-FNO](#)

Inst. fullness: 32.03, bleedless: 42.87, SDR: 17.60

Incompatible with UVR, install [MSST](#), then read model instructions [here](#) (requires modifying bs_roformer.py file in MSST, potentially also models_utils.py in some cases).

Actually similar results to BS-Resurrection inst model above, less fullness.

Some people even prefer Gabox BS_ResurrectioN instead.

“Very small amount of noise compared to other fullness inst models, while keeping enough fullness IMO. I don't even know if phase fix is needed. Maybe it's still needed a little bit.” dca “seems less full than the Resurrection, which I would expect given the MVSEP [metric] results. (...) I'd say it's roughly comparable to Gabox inst v7”

“I replaced the MLP of the BS-Roformer mask estimator with FNO1d [Fourier Neural Operator], froze everything except the mask estimator, and trained it, which yielded good results. (...) While MLP is a universal function approximator, FNO learns mappings (operators) on function spaces.”

“(The base weight is Resurrection Inst)”

- Gabox [Inst_GaboxFv7z](#) Mel Roformer ([yaml](#)) | [Colab](#) | [uvronline.app/x-minus.pro](#)

Inst. fullness: 29.96, bleedless: 44.61, SDR: 16.62

Becruily vocal used for phase fixer on x-minus.pro/uvronline (premium feature).

“Focusing on the less amount of noise, keeping fullness”

“the results were similar to INSTV7 but with less noise” but “the drums are totally fine with this model” - neoculture

“it seems to capture some vocals better” - Gabox

In some songs, “it leaves a lot of reverb or noise from the vocals. unva v1e+ a little better” - GameAgainPL

“[one of the] best bleedless, good fullness, almost noiseless” - Aufr33

- Gabox [inst_fv7b](#) Mel Roformer | [yaml](#)

Inst. fullness 27.07, **bleedless** 47.49, SDR 16.71

Fullness worse than even most vocal Mel-Roformers (incl. BS-RoFormer SW and Mel Kim OG model).

"on the fuller side, somewhere around inst v1e+, maybe a tiny bit below. The main thing I notice is it captures more instruments than v1e+, but isn't muddy like [HyperACE] (which also captures more instruments)
can be a little on the noisy side sometimes... but it at least isn't muddy and sounds natural
(...) I'd still ensemble if you want the noise reduced - rainboomdash
([src](#))

Lower fullness models
(if you find the ones above too muddy, but here you get more noise)

0) Gabox [inst_gabox3 \(yaml\)](#) | [Colab](#) | [Huggingface / 2](#) | [Phase fixer Colab](#)

Inst. fullness 37.69, bleedless 35.93, SDR 16.50

Actually worse fullness than v1e+ (37.89), and lower bleedless (36.53).
When used with Unwa's beta 6 as reference for phase fixer (thx John UVR), slightly less muddy results than phase-fixed Becruily inst-voc results, but also slightly more vocal residues and a bit more inconsistent sound, fluctuations across the whole separation at times.

0) Gabox [INSTV7N](#) | [Huggingface / 2](#)

Inst. fullness 36.83, bleedless 35.47, SDR: 16.65

More noisy than INSTV7; "it's [even] closer to v7 than inst3"

—
0) SCNet XL model called "very high fullness" | [MVSEP](#)

Inst. fullness 34.04, bleedless 35.15, SDR 16.60

It might work better than Roformers for less noisy/loud/busy mixes or genres like alt-pop, orchestral tracks with choir, sometimes giving more full results than even v1e, but at the cost of more noise. Might struggle with some vocal reverbs or effects.

"Very hit or miss. When they're good they're really good but when they're bad there's nothing you can do other than use a different model"

Compared to the high fullness variant, more crossbleeding of vocals in instrumentals (along with SCNet XL basic model). Some songs which sound full enough even with basic SCNet XL (and HF variant) while others will sound muddy (dca)

"has a lot of noise/bleed, and I haven't found the best way to get rid of it, but it does tend to pick up harmonies and subtle BGV that other models don't." dynamic64

0) MVSEP SCNet XL high fullness

Inst. fullness 31.95, bleedless 34.06, SDR 17.26

"I have a few examples where it's better than v1e+

Sometimes there is too much residue but most of the time it's fine" dca

"Really loving the way SCnet high fullness [variant] handles lower frequencies, below 2K [let's] say. Roformers are better with the transients up high, but decay on guitars/keys on the SCnet is more natural"

"seems to also confuse less "difficult" instruments for vocals"

"I noticed classic SCNet XL preserves more instruments than the high fullness one, but has more vocal crossbleeding in instrumental compared to high fullness
So if you want instrument preservation use SCNet XL 1727 but if you want less crossbleeding of vocals in instrumental use SCNet XL high fullness
I ignore the very high fullness one because it has too much vocal residue" dca
(regular SCNet XL moved below)

-

- Gabox BS_ResurrectioN [model](#) | [yaml](#)

"It is a fine-tune of BS Roformer Resurrection Inst but with higher fullness (like v1e for example), it needs [MVSEP's] BS 2025.07 (as a source/reference) phase fix
I requested it because I found some songs where Resur Inst was producing muddy instrum results (...) I requested it not just for me because I saw other people were looking for something like v1e++" - dca

Higher fullness (but with more noise)
(sorted by fullness)

0) Gabox [INSTV6N](#) (N for noise/fullness) | [yaml](#) | [Colab](#) | [SESA](#) | [Huggingface](#) / 2 | [metrics](#):

Inst. **fullness**: 41.68 (more than v1e), bleedless: 32.63 (N "noisier but fuller")

To get rid of noise in INSTV6N, use Gabox [denoise/debleed](#) model ([yaml](#)) on mixture first, then use INSTV6N - "for some reason it gives cleaner results" (Gabox), but it can't remove vocal residues.

Some people find it having less noise vs v1e and more fullness.

Also, it has more fullness vs INSTV6, and more noise, but some people might still prefer v1e.

"v1e sounds like an "overall" noise on the song, while v6n kind of mixes into it.

v6n also sounds like two layers, one of noise that's just there. And the other one mixes into the song somehow. Using the phase swap barely makes it any better than phase swapping with v1e though" - vernight

Also Kim model for phase swap seems to give less noise than unwa ft2 bleedless

"Comparing V6N with v1e and couldn't hear a fullness difference despite the metrics being approx 39 for v1e and 41 for V6N" - dca

"my all-time favorite" - ezequielcasas

0) Gabox [inst_Fv4Noise](#) | [yaml](#) | [Colab](#) | [Huggingface](#) / 2

Inst. fullness 40.40, bleedless 28.57, SDR 15.25

Can be better than INSTV6 for some people, but overkill for others. Bigger fullness metric than even v1e.

"Despite v4's significant amount of noise, it seems to be the only model [till 8 February] that gave me a fuller sounding result compared to v1e that's actually perceivable by my ears." - Shintaro

"although the fullness metric increases when there is more noise, it doesn't always mean it's a better instrumental — an example of this is the fv4noise metrics" - Gabox

- [Neo_InstVFX](#) Mel-Roformer by neoculture | [yaml](#) | [Colab](#) | [Huggingface / 2](#)

Inst. fullness 39.88, bleedless: 32.56, SDR: 14.35

Focused on preserving vocal chops.

"great model (at least for K-pop it achieved the clarity and quality that no other model managed to have) it should be noted that it has a bit of noise even in its latest update, its stability is impressive, how it captures vocal chops, in blank spaces it does not leave a vocal record, sometimes the voice on certain occasions tries to eliminate them confusing them with noise, but in general it was a model that impressed me. It captures the instruments very clearly" - billieoconnell.

"NOISY AF, this is probably the dumbest idea ever had for an instrumental model. Don't use it as your main one, some vocals will leak because I added tracks with vocal chops to the dataset. Just use this model for songs that have vocal chops" - neoculture

0) Unwa Inst V1e (don't confuse with newer +/plus variant above) [Model files](#) (yaml from v1)
Inst. fullness 38.87, bleedless 35.59, SDR 16.37

[Colab](#) | [MSST-GUI](#) | [UVR instructions](#) | [Huggingface / 2](#) | uvronline via special link for: [free/premium](#) (scroll down) | MVSEP

One of the first Mel Kim model fine-tunes trained with instrumental (other) target. High fullness metric, noisy at times and on some songs. To alleviate it, it can be used with automated [phase fixer Colab](#) or UVR>Tools (Kim Mel as reference removes more noise than 2024.10 vs muddier Unwa v1/2 on their own; optionally use VOCALS-MelBand-Roformer by Becriuly or unwa's kim ft; you can also use FT2 as reference, but it "cuts instruments" vs FT3 which can be rather better alternative). Optionally, in Phase Fixer you can set 420 for low and 4200 for high or 500 for both and Mel-Kim model for source; and [bleed suppressor](#) (by unwa/97chris) to alleviate the noise further (e.g. phase fixer on its own works better with v1 model to alleviate the residues). Besides the default UVR default 500/5000 and Colab default 500/9000 values, you could potentially "even try like 200/1000 or even below for 2nd value." "I would say that the more noisy the input is, the lower you have to set the frequency for the phase fixer."

V1e might catch more instruments and vocals than INSTV6N. Even fuller model with more noise is instfv4noise below by Gabox.

"The "e" stands for emphasis, indicating that this is a model that emphasizes fullness."

"However, compared to v1, while the [fullness](#) score has increased, there is a possibility that noise has also increased." "lighter compared to v2." Like other unwa's models, it can struggle with flute, sax and trumpet (unlike Mel 2024.10, and BS 2024.08 on MVSEP respectively - you can max ensemble all the three as a fix [dca100fb8]). Also, sometimes unwa's big beta5e can retrieve missing instruments vs v1e when those two above fails. Possible residues of dual layer vocals from suno songs.

0) [inst_gaboxFv3](#) | [yaml](#) | [Huggingface / 2](#) - F for fullness

inst. fullness 38.71, inst. bleedless 35.62 ("F" stands for fullness) | Inst SDR 16.43

Like v1e when it comes to fullness, but less bleeding.

Vs v1e "it's slightly better with some instruments", It might pick up an entire sax in the vocal stem.

It doesn't have that weird fullness noise that fullness models produce, but still gives pretty full results and the phase swapper (with big beta 6 as reference) gets rid of that weird buzzing sound" John UVR

0) Gabox experimental "[fullness.ckpt](#)" inst Mel-Roformer ([yaml](#)).

Inst. fullness: 37.66, bleedless: 35.53, SDR: 15.91

"this isn't called fullness.ckpt for nothing." - Musicalman

Sorted by the biggest fullness metric on the list:

INSTV6N (41.68)>inst_Fv4Noise (40.40)/INSTV7N (no metrics)/Inst V1e (38.87)>Inst Fv3 (38.71).

While V1e+ (37.89) might be already muddy in some cases.

Sorted by bleedless metric [here](#)

Lower bleedless models/balanced

Still less noise even when using without phase fixer

0) Unwa [BS-Roformer Resurrection inst \(yaml\)](#) | a.k.a. "unwa high fullness inst" on MVSEP | uvronline.app/x-minus.pro | [Colab](#) | [UVR](#) (don't confuse with Resurrection vocals variant)
Inst. fullness: 34.93, bleedless: 40.14, SDR: 17.25
(duplicate from the above, because it fits metrically and categorization-wise here, more info above)

0) Unwa [Mel-Roformer inst v1 \(yaml\)](#) | [Colab](#) | UVR [installation](#) | MVSEP | uvronline via special link for: [free/premium](#) (scroll down)

inst. fullness 35.69, bleedless 37.59

*) Denoising for v1/2/1e recommended with: 1) ensemble noise/phase fix option for x-minus premium 1b) Beccruily [phase fixer](#) (also since UVR beta patch #7) 2) Mel-Roformer [de-noise](#) non-agg. (might be better solution) 3) UVR-Denoise medium aggression (default for free users) 4) minimum aggression for premium/[link](#) (damages some instruments less) 5) UVR-Denoise-Lite [agg. 4, no TTA] in UVR - more aggressive method 6) UVR-Denoise [agg. 30/25, hi-end proc., 320 w.s., p.pr.] - even more muddy but preserves trumpets better v1 might have more instruments missing vs v1e and less noise

0) [inst_gabox2 \(yaml\)](#) | [Huggingface](#) / [2](#)

inst. fullness 36.03, bleedless: 38.02

-

0) Beccruily's inst [model](#) (again, because it fits here metrically)

Inst. [fullness](#) 33.98, bleedless 40.48, SDR 16.47

[Colab](#) | on MVSEP a.k.a. “high fullness” (the same model) | x-minus (w/ optional phase correction feature in premium) | [UVR](#)

For less vocal residues use phase fixer [Colab](#) (also in UVR>Tools) and “begruij's vocals as source and inst as target”

0) [Inst_GaboxFv8](#) v2 model ([yaml](#)) | [Colab](#)

Inst. fullness: 33.21, bleedless: 40.73, SDR: 16.57
(again, just for metrics)

0) Gabox [instV7plus](#) bleedless model (experimental)

inst. fullness: 29.83, bleedless: 39.36, SDR 16.51

—

0) MVSEP SCNet XL (don't confuse with undertrained weights on ZFTurbo's GitHub)

inst. fullness 28.74, bleedless 39.42, SDR 17.27

“I've come across a lot of songs where high fullness [SCNet variant above] gives that annoying static noise. I'm starting to like basic SCNet XL more to the high fullness [model]. And also, less vocal residues.” - dca. There is crossbleeding of vocals in some songs. You can find the dca's list for that model in further parts of [this](#) section.

0) MVSEP SCNet XL IHF

inst. fullness 28.87, bleedless 40.37, SDR 17.41

Some songs struggling with previous models might yield better results.

0) MVSEP SCNet Large

inst. fullness 27.10, bleedless 41.47, SDR 17.05

Higher bleedless (not so full)

0) Gabox B/bleedless v3 ([inst_gaboxBv3](#)) | [Huggingface / 2](#)

Inst. fullness: 32.13, bleedless: 41.69, SDR 16.60

“can be muddy sometimes” but still fuller than the older one below

0) Unwa Mel-Roformer inst v2 (similar but fewer vocal residues (not always), muddier, bigger, heavier model)

Inst. fullness 31.85, bleedless 41.73 (less bleeding than Gabox instfv5/6)

[Model files](#) | [Colab](#) | [Huggingface / 2](#) | uvrone via special link for: [free/premium](#) (scroll down) | [MSST-GUI](#) (or OG [repo](#)) | [UVR](#) Download Center)

Might miss flute. “Sounds very similar to v1 but has less noise, pretty good” “the aforementioned noise from the V1 is less noticeable to none at all, depending on the track”. “V2 is more muddy than V1 (on some songs), but less muddy than the Kim model. (...) [As for V1,] sometimes it's better at high frequencies” Aufr33

Might miss some samples or adlibs while cleaning inverts. SDR got a bit bigger (16.845 vs 16.595).

"Significantly less noise than v1e, sounds full enough, despite the fullness inst score, and that it recognizes more instruments than v1 and v1e, added to the fact it has higher SDR so also slightly less vocal crossbleeding in instrumental." - dca100fb8

0) Unwa [BS-Roformer-Inst-FNO](#)

Inst. fullness: 32.03, bleedless: 42.87, SDR: 17.60

(again, because it fits metrically, more info moved near the top to recent bleedless section)

0) Gabox [Inst_GaboxFv7z](#) Mel Roformer | [yaml](#) | [Colab](#) | uvrone/x-minus.pro

Becruily vocal used for phase fixer on x-minus.pro/uvrone (premium).

Fullness: 29.96, **bleedless**: 44.61

(again, because it fits metrically, -||-)

Last resort - muddier but cleaner single vocal models with more bleedless tested for instrumentals (sorted by bleedless) [here](#) | [descriptions](#)

Other or older instrumental models (less muddy than vocal models)

- Full release of mesk's rifforge Mel-Roformer [model](#) focused on inst/voc separation for metal music

"The model can have some quirks (just like most models) but it's all around clean for me to release." "It kinda also fucks up in like cleans [not distorted/non-metal vocals]"

Training details:

"Characteristics:

This is a dimension 512 depth 24 model (so fairly large file size at 1.9 GB!), with an SDR of 14.2436.

It's finetuned from an older Melband Roformer checkpoint with an SDR of 13.7."

- "I think I found the (IMO) the best process for metal:

1. inferencing using the BS 07.2025 model on MVSEP

2. inferencing using my rifforge model

3. ensembling both with a min_fft ensemble" - mesk

it keeps the "fullness" of the rifforge model being an instrumental focused model but then also removes more stuff than my base model thanks to 07.2025"

- (old) mesk's "Rifforge" metal Mel-Roformer fine-tune instrumental model (focused more on bleedless).

Inst. fullness: 28.49, bleedless: 42.38, SDR 16.67

"training is still in progress, that's why it's a beta test of the model; It should work fine for a lot of things, but it HAS quirks on some tracks + to me there's some vocal stuff still audible on some tracks, I'm mostly trying to get feedback on how I could improve it" [known issues](#).

<https://drive.proton.me/urls/5XM3PR1M7G#F3UhCU8RDGhX>

Be aware that MVSep's BS-Roformer 2025.07 can be better for metal both for vocal and instrumentals than these mesk's models, a lot of the times. It was also trained on mesk's metal dataset.

- Custom model import Colab has currently some issues with it. Probably, using that old version will work (at least locally).

"My old MSST repo I'm using, but I removed all the training stuff

<https://drive.proton.me/urls/P530GFQR4W/#VCAsF0E1TPje>

pip install -r requirements.txt (u gotta have Python and PyTorch installed as well) for the script to work.

You just gotta put all the tracks you want to test on in the ***"tracks"*** folder then double-click on ***"inference.bat"*** to run the inference script

its like if you were to type in the command in cmd but its simpler, and I'm lazy" - mesk

- Older Mesk metal Mel-Roformer preview instrumental model

Inst. fullness: 28.81, bleedless: 42.16, SDR 16.66

Retrained from Mel Kim on metal dataset consisting of a few thousands of songs.

https://huggingface.co/meskville33/metal_roformer_preview/tree/main | Colab

(previous metrics were made on private dataset)

"Should work fine for all genres of metal, but doesn't work on:

- hard compressed screams

- some background vocals

- weird tracks (think Meshuggah's "The Ayahuasca Experience")"

P.S: Use the [Colab](#) "or training repo [MSST](#) if you want to [separate] with it. UVR will be abysmally slow (because of chunk_size [introduced since [UVR Roformer beta #3](#)])"

0) [INSTV6](#) by Gabox | [yaml](#) | x-minus | [Colab](#)

Inst. fullness 37.62, bleedless 35.07, SDR 16.43

v1e still gives better fullness, but the noise in it is a problem

Opinions are divided whether v5 or v6 is better.

"Seems like a mix between brecuily and unwa's models"

"Slightly better than v5 (...) less muddy and also removes the vocals without adding that low EQ effect when the vocals would come in, so I feel it's better" zzz

Old viperx' 12xx models have fewer problems with sax.

- [Inst_GaboxFVX](#) | [yaml](#) | [Huggingface](#) / 2

Inst. fullness 38.25, bleedless 35.35, SDR 16.49

"instv7+3" - fuller than instv3

- Gabox [instv10](#) (experimental) | [yaml](#)

Less noise and vocal residues than V7, but muddier

0) Gabox Mel-Roformer instrumental model “[inst_gabox.ckpt](#)” (Kim/Unwa/Becruily fine-tuned)

Gabox’ [models](#) repo | [Colab](#) | [Huggingface](#) / 2

inst fullness 37.07 (better than unwa inst v1 and v2), bleedless 37.40 (better than v1e by 1.8, slightly worse than unwa’s v1)

“It’s like the v1 model with phase fixer, but it gets more instruments, like, it prevents some instruments from getting into the vocals”, “sometimes both models don’t get choirs”.

Faster inference models

(small model size/potentially workable on not ancient CPUs):

MDX-Net HQ_3, 4, 5 (the last is the fastest, 56 MB)

MDX-Net inst3, Kim inst (older, narrowband, but can be useful too in some cases, 63 MB)

Not sure if on CPU, but rather light, small (the lightest Roformers, while most have 870 MB):

Unwa BS-Roformer-Inst-FNO (works only in MSST after modifying py file like in the model card, vs Resurrection inst model “it’s a considerably higher bleedless/lower fullness model” - rainboomdash, 332 MB)

Gabox Mel-Roformer [small_inst](#) | [yaml](#) (experimental. even smaller - 203 MB)

Unwa BS-Roformer-Inst-EXP-Value-Residual (low performance, use v2 model type in UVR)

Older fullness models

0) Gabox F/fullness v1 | [Huggingface](#) / 2

inst fullness 37.26 | bleedless: 37.19

0) Gabox F/fullness v2 | [Huggingface](#) / 2

inst fullness 37.46 | bleedless: 37.09

*) Gabox [inst_Fv4](#) (F - fullness/v4) | (don’t confuse with vocal fv4) | [yaml](#) | [Colab](#)

inst fullness 39.40 | bleedless 33.49

Duplicate from the above

Others

0) [instrumental_gabox](#) | [yaml](#) | [Huggingface](#) / 2

0) Gabox B/bleedless v1 instrumental [model](#) | [yaml](#) | [Huggingface](#) / 2

Inst. fullness 35.03, bleedless 39.10, SDR 16.49

0) Gabox B/bleedless v2 instrumental [model](#) | [yaml](#) | [Huggingface / 2](#)

Inst. fullness 35.09, bleedless 38.38, SDR 16.49

([Gabox models repo](#))

0) Cut your song into fragments consisting from the best moments of e.g. v1e/v1/v2 into one (and optionally Mel-Roformer Bas Curtiz FT on MVSEP as it will give you even less vocal bleeding, but more muddiness if necessary)

0) Propositions of models for phase fixer to alleviate vocal residues (from the above)

a) Bebruily voc with Bebruily inst (muddy but very few residues if any)

b) FT3 with V1e+

c) Unwa Beta 6 with inst_gabox3 (although it might be less consistent than the top)

d) Unwa Revive model is also good with any instrumental model

e) Unwa Bigbeta5 used to be not bad either.

f) Or any of the vocal models above with e.g. V1e (it's pretty full, and it might be not enough for it nevertheless)

How to use the phase fixer in UVR?

Separate with vocal model, then with instrumental model. Go to Audio Tools>Phase swapper, and use vocal model result as reference, and instrumental as target

—

Ensembles

(for instrumentals; check out also [DAW ensemble](#) with the below)

If you find some phase fixer results (e.g. ensembled) unsatisfactory, use the Phase Fixer [Colab](#) - it's tweaked for better results than UVR and standalone scripts.

0) BS HyperACE + BS 2025.07 (Max FFT with BS 2025.07 as phase fix reference and 3000/5000 for the values)

—>My favorite ensemble right now. Though, I notice it produce vocal bleed sometimes and it can be noisy at some parts of the songs while the noise might be totally absent in other parts of the song.

- dca100fb8 (if not said otherwise)

0) BS-Roformer Resurrection Inst (phase fixed with BS 2025.07 using Low Cutoff 3000 and High Cutoff 5000) + BS 2025.07 (Max Spec)

—>Former best

0) Unwa Mel inst v1e+ with FNO with Bebruilly inst (Max Spec)

—> The best result back then

- sakkuhantano

0) Unwa Mel inst V1e + MVSEP BS 2025.07 (Max Spec) using BS 2025.07 as phase fix reference with 200/200 100/100 (it's better) as Low Cutoff and High Cutoff values

—>It's very aggressive values because V1e is noisy, and it works quite well.

The older best for then, “BS Roformer Resur Inst [ensemble right below] is muddy compared to v1e, and I think fullness is the way. After phase fix the noise is barely noticeable”

0) Unwa BS Roformer Resurrection Inst (BS 2025.07 as a reference for phase fix) + MVSEP BS Roformer 2025.07 (Max Spec)

—>The least vocal crossbleeding (step-by-step process explained [here](#))

Alternatively, you can use becruily vocal model instead of 2025.07 for the ensemble -

“Becruily vocal correctly recognize instruments far better than the instrumental one”

(Note: BS 2025.07, BS 2024.04, BS 2024.08 and SW were worse for Resurrection model as a phase fix source, viperx BS-Roformer 1297 better, but not so good for instruments preservation as BS-2025.07)

0) unwa v1e + Mel becruily vocal (Max Spec) + phase fix (using becruily vocal again as a source)

—>The best instruments preservation (with more possible crossbleed)

0) Mel Gabox Fv7z + BS 2025.07 (Max Spec)

—>The least amount of noise (with more possible crossbleed)

0) Mel Gabox Inst V8 + BS 2025.07 (Max Spec) + phase fix (becruily vocal as reference)

—>A good balance between presence of noise and level of fullness
(occasional vocal crossbleeding)

0) Mel Becriuly Instrumental (with phase fix, becruily vocal as reference) + SCNet XL IHF (Max FFT)

—>Why SCNet? Because it's better than Mel Roformer at the low frequencies, so why not ensemble both arch. (...) SCNet is already noisy from the start so the fullness models are even noisier obviously

0a) Unwa v1e+ + BS-Roformer 12xx by viperx (Max Spec) - musicalman

0a) FNO inst by unwa + BS-Roformer 12xx by viperx (might be optional) + v1e+ (or becruily inst Mel-Roformer)

“beware of the song where there's a vocal at the beginning of the song, using v1e+ will leave vocal residue. So decide to change into becruily inst as well.” - Sakkuhantano

0*) Mesks's metal min_fft ensemble of BS 07.2025 model on MVSEP + rifforge [model](#)
"it keeps the "fullness" of the rifforge model being an instrumental focused model but then
also removes more stuff than my base model thanks to 07.2025"

0*) Chained separation method by fabio06844 for "very clean and full" instrumental.
1) Go to MVSep and separate your song with the latest Karaoke BS-Roformer by MVSep Team
2) On its instrumental stem result use DEBLEED-MelBand-Roformer (by unwa/97chris)
([model](#) | [yaml](#) | [Colab](#))
(despite the fact that "the MVSep Team Karaoke uses the MVSep BS model to extract/remove vocals, then applies [the] karaoke model to that", it was told to be not enough to just use BS 2025.07 model instead, leaving a little more residues).

0b) v1e phase swapped from Beccruily vocals + BS 2025.07 (Max Spec)
(Phase fixer [Colab](#)/or UVR's phase swapper+MVSEP separation>UVR Manual Ensemble)
("brings: max fullness without a lot of noise since phase fix, rarely missing instruments, no robotic voice problem, rare vocal crossbleeding in instrumental ") - older favourite dca100fb8's ensemble

0b) v1e + Beccruily vocal (Max Spec) ("If you had to keep one ensemble right now. v1e+ unfortunately is muddier than v1e and has that robotic issue sometimes") - dca100fb8

0b) v1e + Beccruily inst + Beccruily vocal (Max Spec) (Beccruily inst turned out to be "useless" in this bag of models) - -||-

0b) Unwa v1e (with phase fix) + BS Large V1 (Max Spec) (doesn't need [Beccruily vocal](#) here as the third) - -||-

0) v1e + INSTV7 (Max Spec) - neoculture

0) Use MVSEP's SCNet XL high fullness below 1000 Hz, and unwa's v1e above 1000 Hz, and join the two in e.g. Izotope RX - "You can use vertical select in RX with feathering set to 1.00" (heuhew) or you can use linear phase EQ like e.g. free "Ikjb QRange" (ensure to not overlap frequencies in the output spectrogram in the crossover point)

0) v1e+ + beccruily inst
—> fixes some missing instruments occasionally in v1e+
- Sakku

0b) INSTV7 + Inst_FV8 (to check)

0b) unwa's instv1e+, instv1+, instv2 and inst gabox, instv8 and instv7 - max FFT
"Then, I upscaled using Apollo. Afterward, I applied [Mel-Roformer] de-noise to remove background noise as needed and performed mastering" (Sir Joseph)

0b) Max Spec manual ensemble of: v1e+ + MVSEP BS-Roformer 2025.07 model

- It's a Senn's method below, simplified (IntroC)

"I think (...) [it] would be good enough. The way v1e+ keeps the noise is usually fine, and the mvsep 2025.07 model should bring back the lost masked frequencies for v1e+. Otherwise, just adjust the weight of v1e+ for maxspec to reduce the noise"

0b) Max Spec manual ensemble of: v1e+ + MVSEP BS-Roformer 2025.07 model + Bebruily inst

- Senn's method simplified (IntroC/Sakku)

"sometimes bebruily can catch a tiny instr while v1e+ can't" - Sakku

*) Senn's OG method:

Use the highest SDR BS Roformer model on MVSEP and the best Fullness Melband Ro-Former model (unwa instrumental v1e plus) - mixed both with one of them phase-inverted, then use Soothe2 to filter out resonants, leaving mostly only noise, further filter and mix them, use a few plugins to do a spectral flattening, very minor.

In other words:

"Pass the music through BS RoFormer's best SDR and the best Fullness, which is Unwa Instrumental v1e plus

in iZotope RX, Invert the phase of any of the tracks, and copy n paste to the other track, the result should be some ghastly sounding reverb of the vocals

using iZotope RX's Deconstruct, you wanna filter out the tonal signal of the voice as to remove the more obvious "sinusoidal" signals. It has to be fairly subtle as to not damage the noisy residuals

Now with Soothe2, you wanna filter out any of the more aggressive noisy components, I use this setting, but it might not work 100% for everyone <https://imgur.com/2A5yn3c> (You can replace this with any plugin that acts similar to Soothe2, but Soothe2 is the best compromise)

If needed, you can use Deconstruct as well but reducing the noisy aspects just to wipe out that aggressive noisy artifact

Mess with the gain, and then add it back to the BS RoFormer track (don't forget to invert the phase again), Ideally it should be fairly subtle

For post-processing:

I use MSpectralDynamics to add a slight spectral flattening to the track, and Unchirp to denoise very slightly the higher frequencies to remove that digital hissy artifact and also tighten more of the sound

A very subtle Gullfoss can also brighten the track slightly as well to compensate here is the result" (thx senn)

Older ensembles (from before bebruily inst/voc Mel models release)

UVR>Audio Tools>Manual ensemble (for models from outside UVR)

0) Unwa's v1e inst + phase fixer/swapper (from Mel-Kim or sometimes Mel 2024.10 on MVSEP for less noise) + BS 2024.08 on MVSEP (Max Spec)
(fullness with less noise + retrieved missing wind instruments from v1e) - dca100fb8
Becruily vocal is even better at recognizing instruments compared to Mel 2024.10 or BS 2024.08 (vocal Roformer models have fewer problems with recognizing instruments than inst Roformers)

- 0) Other dca's Max Spec ensembles of v1e with other Roformer models
 - II) v1e + phase fixer/swapper + Mel 1143 (because of fullness with less noise + retrieved missing wind instruments from v1e)
 - III) v1e + phase fixer/swapper + BS 1296 (because of fullness with less noise + retrieved missing wind instruments from v1e)
 - IV) v1e + phase fixer/swapper + BS 1297 (because of fullness with less noise + retrieved missing wind instruments from v1e)
 - V) v1e + phase fixer/swapper + BS Large V1 (because of fullness with less noise + retrieved missing wind instruments from v1e)
- Recommended (or with v2/v1 instead) esp. when using a phase fixer due to bleed of instruments in the vocal track in Kim or its fine-tunes used for the tool.
- VI) (extra) for slow CPU/GPU: Voc FT + HQ5 (Max Spec)
- VII) HQ_5 with UVR's Phase Rotate (which now can replace the above)
- VIII) v1e + BS 2025.06 (Max Spec) - manual ensemble - the latter on MVSEP (because it keeps instruments in instrumental correctly [though less than becruily vocal] and it has less vocal crossbleeding in instrumental compared to becruily vocal)

- 0) Unwa's Inst V2 and Inst Gabox (Avg) (1120 segments [6GB GPU]/4 overlaps) - cypha_sarin
- 0) Max ensemble of: instv1, instv2 and inst v1e - erdzo125
(better fullness than inst v1e itself, but more noise)
- 0b) Models ensembled - available only for premium users on mvsep.com
Now also added "instrumental high fullness" variant for inst, voc ensemble.
For example, some lower inst, voc SDR ensembles available might be less muddy than 11.50 (e.g. 10.44), but the 11.50 one has the fewer amounts of vocal residues according to **bleedless** metric, but it can also sound very filtered. Newer ensembles added since then (track [the leaderboard](#) and click on entries to see also bleedless/fullness metrics).
(ensembles on MVSEP provide currently the best of SDR scores for 2 and 4 stem separators, higher SDR than free v.2.4/2.5 Colabs below; 2025.06.28 has currently the biggest SDR metric, and surpassed ByteDance private model)
There are shorter queues for single model separation for registered users with at least one point.
Possibly shorter queues between 10:00 PM - 1:00 AM UTC.
The ensemble option fixes some issues with general muddiness of older vocal Roformer models (but 11.50 is muddier than v. 2.4 Colab).

0) [KaraFan](#) (e.g. preset 5; fork of original ZFTurbo's MDX23 fork with new features by Captain FLAM with jarredou's help on some tweaks), [offline version](#), org. [Colab](#) and [Kubinka](#) Colab (older version, less vocal residues vs. v.3.1, although v.3.2-4.2/+ were released with fewer residues).

Used to be one of the best free solutions for instrumentals (before some newer Roformers like unwa's inst [v1](#) were released), with not big amounts of vocal residues (sometimes more than below), and clear outputs. But no 4 stems unlike below:

0a) *MDX23 by ZFTurbo (weighted UVR/ZF/VPR models)* - free modified Colab fork v. [2.1](#) - [2.4](#), [2.5](#) with fixes and enhancements by jarredou (one of the best SDR scores for publicly available 2-4 stem separator, [v2.2.2](#)) Colab with fullband MDX23C model might have more residues in instrumentals vs v. 2.1, but better SDR, [2.7b](#) (with SCNet XL, not SDR evaluated - weight set by ear), [2.3](#), 2.4 with also 12xx BS-Roformer, v. 2.5 with also Kim's Mel-Roformer (default settings can be already good and balanced, and weights further adjusted, [read](#) for more settings).
The caveat - it haven't been updated by newer Roformer instrumental models at the top).

0a) dango.ai ([tuanzai.com/en-US](#)) - 8\$ every 10 songs, currently one of (if not) the best instrumental separator so far; at least till unwa inst models came to the level of fullness of Dango now (you might find the latter even too muddy), but in cost of more noise, although "it can handle complex vocals/songs well so it's more reliable, and no vocal bleed in background of instrumental".
Dango's 10 Conservative mode give more fullness to instrumentals in cost of whispering artefacts (experimental for the time being - along with the Aggressive mode), and it doesn't fix vocal popping using Smart Mode (default). Now Dango 11 is available. More crossbleeding than Unwa Bs-Roformer inst. What changed for better is less noise and better instrument detection

Ensembles (from before becruily models release; less noisy single models later below)

0b) Avg Spec ensemble of unwa inst v1 and v2

0b) Min Spec manual ensemble of vocals stems from these models>inversion with the original song (fuller, more noise)
(UVR>Audio Tools>Manual Ensemble)

0b) Max ensemble of: unwa's v1e + Mel 2024.10 + BS 2024.08 - dca100fb8
(older bag of models; 2024.08 on MVSEP; fixes the flute and trumpet issues)

0b) Separate the song twice - first with v1e, then with Unwa's BS-Roformer Large and do a Manual Max Spec Ensemble via UVR - dca100fb8
(old; BS-Roformer is here to retrieve the missing instruments from v1e result, though BS 2024.08 & Mel 2024.10 on MVSEP work better for this task already)

(Ensembles from before unwa inst. models release - you might also try out replacing all the 12xx BS-Roformers below with newer unwa's/becruily/Gabox models)

0b) Models ensembled - available only for premium users on [x-minus.pro](#)

- max_mag ensemble (with viperx 1297 Roformer)
- demudder (on Mel-Roformer)
- Mel-Roformer + MDX23C

UVR 5 ensembles (although beta 4 and inst [v1](#) on their own might be better already)

*For Roformers, min. RTX 3050 8GB or faster AMD/Intel ARC/Apple M1-3 recommended
(OpenCL is not as fast as CUDA in UVR; 6GB VRAM on CUDA [should](#) be enough too, min.
2K+ CUDA cores recommended)*

0b) 1296 + 1143 (BS-Roformer in [beta](#) UVR) + MDX-Net HQ_4 (dopfunk)

[potentially try out Mel Kim instead of 1143 above already]

0b) Manual ensemble (in UVR's Audio Tools) of:

BS-Roformer 1296 + file copy of the result + MDX23C HQ (jarredou; [src](#))
or just 1296 + 1297 + MDX23C HQ for slower separation and similar result

0b) Manual ensemble of:

- BS-Roformer 1296 + drums stem from demucs_ft or
- Bs-Roformer 1143 result passed through demucs_ft for drums to ensemble with 1296
(max/max)

0b) MDXv2 HQ_4 + BS-Roformer 1296 + BS-Roformer 1297 + Melband RoFormer 1143
(Max Spec) “Godsend ensemble for demuddiness” (dca100fb8)

0b) Manual ensemble of [HQ_5 \(paid users\)](#) and Kim's Mel-[Roformer](#) (max_spec)

0b) (for metal) “1 – pass through Kim's Vocal Melband Roformer (link in Single models below)

2 - Multi-stem Ensemble (Average algo):

1_HP-UVR

MGM_HIGHEND_v4

MGM_LOWEND_A_v4

(VR advanced settings: 320 window size, 5 aggression setting, batch size default, TTA enabled | Post Process and High-End Process CHECKED OFF”

3 – Manual Ensemble both your Melband output and the Multi-stem instrumental output (with Average algorithm)

“best settings/models for metal” (~mesk)

0b) (older version of the above) 1_HP_UVR + UVR_MDX-NET-Inst HQ 4 +

UVR_MDX-NET-Inst_Main 438 ([VIP](#) model)

(Min Spec / Average, WS 512, TTA Enabled, Post-Process and High-End Process off)

0b) 9_HP2-UVR and Kim Mel-Roformer (newer one for metal; mesk)

but not in multi-stem cos you need 3 or more models
VR: 320 window size, 1 aggression setting, Default batch size, TTA enabled (post process and high-end process isn't enabled)

0b) Mateus Contini's [method](#) e.g. #2 or #4

0b) 9_HP2-UVR + BS-Roformer 1297

0b) BS-Roformer ver. 2024.08 + MelBand Roforrmmer (Bas Curtiz edition) + MDX-Net HQ4 + SCNet Large, Max Spec Ensemble (dca100fb8)

0b) BS-Roformer ver. 2024.08 + MelBand Roforrmmer (Bas Curtiz edition) (Max Spec Ensemble) --> result. Result + MDX-Net HQ4 + SCNet Large (Average Ensemble) - -||-

0b) 1297 (ev. 1296) + MDX23C HQ2 (CZ-84)
[or potentially unwa's BS-Roformer instead of 12xx]

See also [DAW ensemble](#) (*older ensembles later below*)

(more about) unwa's instrumental Mel-Roformer v1 model | MVSEP | x-minus.pro
<https://huggingface.co/pcunwa/Mel-Band-Roformer-Inst/tree/main> | [Colab](#) | [UVR instructions](#)

"much less muddy (..) but carries the exact same UVR noise from the [MDX-Net v2] models"
But it's a different type of noise, so aufr33 denoiser won't work on it.

"you can "remove" [the] noise with uvr denoise aggr -10 or 0" although with -10 it will make it sound more muddy like Kim model and synths and bass are sometimes removed with the denoiser (~becruily)

" if there is any voice left [or also background noise], use the Mel-Roformer de-noise with minimal aggression.

This inst model "doesn't eliminate vocoder voices well from an instrumental".

For the noise in the model, vs the ensemble trick on x-minus using Mel-Roformer de-noise might be better alternative:

"removes more noise from the song keeping overall instrument quality more than the new button [on x-minus]" koseidon72. But the more aggressive variant of the Mel model sometimes deletes parts of the mix, like snares. UVR-Denoise-Lite doesn't seem to damage instruments like non-lite UVR-Denoise in UVR, but still more than Mel denoise (recommended aggr. - 4, with 272 vs 512 windows size it's less muddy, TTA can stress the noise more, somewhere above 10 aggr. it gets too muddy). UVR-Denoise on x-minus is even less aggressive (it's medium aggression model for free users who don't have aggression pick), but it might catch ends of some instruments like bass occasionally.

Premium minimum aggression model is somehow more muddy, but doesn't damage instruments.

For more muddy Roformers consider using Aufr's [demudder](#) (it's used for premium on x-minus for Kim Mel model) although it might increase vocal residues, and UVR demudder (explained there later below).

Muddier but cleaner single models (Roformer vocal models with fewer instrumental residues vs instrumental models without necessity of using phase fixer)

0c) MVSep BS-Roformer (2025.07.20)
Inst. fullness 27.83, vleedless 49.12, inst SDR 18.20
Probably a retrain of the SW model on a bigger dataset.

0c) BS-RoFormer SW 6 stem (MVSEP/Colab/undef13 splifft) / vocals only
Inst. fullness 27.45, bleedless 47.41, inst SDR 17.67
(use inversion from vocals and not mixed stems for better instrumental metrics)
Known for being good on some songs previously giving bad results.

0c) 10.2024 Mel-Roformer vocal model on MVSEP
Inst. fullness 27.84, bleedless 47.37, inst SDR 17.59
The cleanest, but muddy compared to models trained for instrumentals
Capable of detecting sax and trumpet, but still muddier than instrumental models above.
Bas Curtiz vocal model fine-tuned by ZFTurbo.

0c) Gabox [voc_fv4](#) | [yaml](#) | [Colab](#)
Good for anime and RVC purposes (codename)
And also for instrumentals, if you need less vocal residues than typical instrumental Roformers (even less than Mel Kim, FT2 Bleedless, or Beta 6X - makidanyee).

0c) Unwa's beta 5e model originally dedicated for vocals | [Colab](#) | [MSST-GUI](#) | [UVR instr](#)
Model [files](#) | yaml: big_beta5e.yaml or [fixed](#) for AttributeError in UVR
Inst. fullness: 27.63 (bigger than Mel-Kim) | bleedless 45.90 (bigger than Kim FT by unwa, worse than Mel-Kim) | Inst. SDR 16.89
Mainly for vocals, but can be still a decent all-rounder deprived of noise present in unwa's inst v1e/v1/v2 models, also with fewer residues than in Kim FT by unwa, and also more consistent model than Kim Mel model in not muffling instrumental a bit in sudden moments.
The third highest bleedless instrumental [metric](#) after Mel-Kim model (after unwa ft2 bleedless in [vocals](#)).

It seems to fix some issues with trumpets in vocal stem (maxi74x1).
It handles reverb tails much better (jarredou/Rage123).

Noisier/grainier than beta 4 (a bit similarly to Apollo lew's vocal enhancer), but less muddy.
“The noise is terrible when that model is used for very intense songs” - unwa
Phase fixer for v1 inst model doesn't help with the noise here (becruily).
“It's a miracle LMAO, slow instrumentation like violin, piano, not too many drums...
it's perfect... but unfortunately it can't process Pop or Rock correctly” gilliaan
“the vocal stem of beta5e may have fullness and noise level like duality v1, but it may also suffer kind of robotic phase distortion, yet may also remove some kind of bleed present in other melrofo's.” Alisa/makidanyee
“particularly helpful when you invert an instrumental and then process the track with it.”
gilliaan

0c) Unwa's Kim Mel-Band Roformer FT2 | [download](#) | [Colab](#)

Inst. fullness: 28.36, bleedless: 45.58

Decent all-rounder too, sometimes less bleeding in instrumentals than 5e.

0c) Unwa Kim Mel-Band Roformer Bleedless FT2 | [download](#) | [Colab](#)

0c) Bas Curtiz' edition Mel-Roformer vocal model on MVSEP

(it was trained also on ZFTurbo dataset)

“Music sounds fuller than original Kim's one & the finetuned version from ZFTurbo [iirc below]. Even [though] the SDR is smaller than BS Roformer finetuned last version, but almost song has the best result in instrumental.” Henri

It can struggle with trumpets more than the other Mel-Roformer on MVSEP [whether 08.2024 or Mel-Kim, can't remember].

0c) BS-Roformer 2024.08.07 vocal model on MVSEP

Inst. fullness 26.56 (less than Mel-Kim), Inst. bleedless 47.48 (the only single model with that better metric than Mel-Kim)

Inst SDR 17.62

vs 2024.04 model +0.1 SDR and “it seems to be much better at taking the vocals when there are a lot of vocal harmonies” also good for Dolby channels.

Capable of detecting flute correctly

0c) Mel-Roformer vocal model by KimberleyJSN - [model](#) | [config](#) | [Colab](#)

Inst. fullness 27.44 (worse than beta 5e and duality, but better than current BS-Roformers)

Inst. bleedless 46.56 (the best [metric](#) from public models)

Inst SDR 17.32

It became a base for many Mel-Roformer fine-tunes here.

(works in UVR [beta Roformer/Colab/CML inference/x-minus/MDX23 2.5](#) (when weight is set only for Mel model)/simple model [Colab](#) (might have problems with mp3 files)

It's less muddy than older viperx' Roformer model, but can have more vocal residues e.g. in silent parts of instrumentals, plus, it can be more problematic with wind instruments putting

them in vocals, and it might leave more instrumental residues in vocals. SDR is higher than viperx model (UVR/MVSEP) but lower than fine-tuned 2024.04 model on MVSEP.

0c) Unwa Revive 2 BS-Roformer (“my first impression is it may have less low end noise than fv4 but not the best in the overall quality and amount of residues in vocal” - makidanyee)

0c) BS-Roformer Large vocal model by unwa (viperx 1297 model fine-tune) [download](#)
Older BS model. It picks more instruments than 12xx models. More muddy than Kim’s Roformer, a bit less of vocal residues, a bit more artificial sound. Also tends to be more muddy than viperx 1297, sometimes muffling instrumental at times, but a bit less of vocal residues, a bit more artificial sound/a bit less musical. Sometimes it has more vocal residues than beta 5e.

Compared to BS, Mel-Roformers can be a good balance between muddiness and clarity for some instrumentals.

Compared to ZFTurbo (MVSEP) and viperx models, Kim’s trained on Aufr33’s and Anjok’s dataset.

UVR manual model installation (Model install option added in newer [patches](#)):

Place the model file to Ultimate Vocal Remover\models\MDX_Net_Models and the config to model_data\mdx_c_configs subfolder and “when it will ask you for the unrecognised model when you run it for the first time, you’ll get some box that you’ll need to tick “Roformer model” and choose its yaml” some models here are available in Download Center too.

Other unwa fine-tunes (originally vocal models)

0c) Mel-Roformer Kim | FT (by unwa) | [Colab](#)

<https://huggingface.co/pcunwa/Kim-Mel-Band-Roformer-FT/tree/main>

Inst. fullness 29.18 (lower than only unwa inst models)

Inst. bleedless 45.36 (lower than Beta 5e)

Inst. SDR 17.32

Has more vocal residues than Beta 5e

- Aname Mel-Roformer [duality model](#).

It’s focused more on bleedless than fullness metric contrary to the unwa’s duality v2 model, but with bigger SDR.

Inst. fullness 24.36, bleedless 46.52, SDR: 17.15

- Mel-Roformer unwa’s inst-voc model called “duality v1/2” (focused on both instrumental and vocal stem during training; two independent and not inversible stems inside one weight file).

<https://huggingface.co/pcunwa/Mel-Band-Roformer-InstVoc-Duality> | [Colab](#) | [MVSEP](#)

V1: Inst fullness 28.03, bleedless 44.16, SDR 16.69.

V2: Inst SDR 16.67

Outperformed in both metrics by the unwa's Kim FT.
Vocals sound similar to beta 4 model, instrumentals are deprived of the noise present in inst v1/e models, but in result, they don't sound similarly muddy to previous Roformers.
Compared to beta 4 and BS-Roformer Large or other archs' models, it has fewer problems with reverb residues, and vs v1e, with vocal residues in e.g. Suno AI songs.
"other" is output from model, "Instrumental" is inverted vocals against input audio.
The latter has lower SDR and more holes in the spectrum, using MSST-GUI, leave the checkbox "extract instrumental" disabled for duality models (now it's also in the Colab with "extract_instrumental" option) and probably for inst vx models.
You can use it in the Bas Curtiz' [GUI](#) for ZFTurbo script or with the OG ZF's repo code.

- unwa's Mel-Roformer fine-tuned beta 3 (based on Kim's model)
<https://huggingface.co/pcunwa/Mel-Band-Roformer-big/tree/main> | [Colab](#)
Inst SDR: 17.30
Since beta 3 there's no ringing issues in higher frequencies like in previous betas.
Sometimes better for instrumentals than beta 4 - but tends to be too muddy at times, but with fewer vocal residues than beta 5.
- unwa's Mel-Roformer beta 4 (Kim's model fine-tuned)
<https://huggingface.co/pcunwa/Mel-Band-Roformer-big/tree/main> | [Colab](#)
Outperformed in both metrics by beta 5e.

Be aware that the yaml config is different in this model.
"Metrics on my test dataset have improved over beta3, but are probably not accurate due to the small test dataset. (...) The high frequencies of vocals are now extracted more aggressively. However, leakage may have increased." - unwa
"one of the best at isolating most vocals with very little vocal bleed and still doesn't sound muddy" Can be a better choice on its own than some ensembles.
0c) SCNet XL (vocals, instum)
Inst SDR: 17.2785
Vocals have similar SDR to viperx 1297 model,
and instrumental has a tiny bit worse score vs Mel-Kim model.

0c) Older SCNet Large vocal model on MVSEP
"just like the new BS-Roformer ft model, but with more bleed. [BS] catches vocals with more harmonies/bgv" - isling. "it's like improved HQ4" - dca100fb8
Issues with horizontal lines on spectrogram.

0d) Aname Mel [model](#) trained from scratch a.k.a. Full Scratch
Inst. fullness: 25.10, bleedless: 37.13

Models for older archs

0c) MDX23C 1666 model exclusively on mvsep.com
(vocal Roformers are much more muddy than MDX23C/MDX-Net in general, but can be cleaner)

0c) MDX23C 1648 model in UVR 5 GUI (a.k.a. MDX23C-InstVoc HQ / 8K FFT) and mvsep.com, also on x-minus.pro/uvronline.app
Both sometimes have more bleeding vs MDX-Net HQ_3, but also less muddiness.
Possible horizontal lines/resonances in the output - fix DC offset and or use overlap “starting from 7 and going multiples up - 14 and so on.” Artim Lusis

0c) MDX23C-InstVoc HQ 2 - [VIP model](#) for UVR 5. It's a slightly fine-tuned version of MDX23C-InstVoc HQ. “The SDR is a tiny bit lower, but I found that it leaves less vocal bleeding.” ~Anjok

It's not always the case, sometimes it can be even the opposite, but as always, all may depend on a specific song.

0d) MDX-Net HQ_4/3/2 (UVR/MVSEP/x-minus/[Colab/alt](#)) - small amounts of vocal residues at times, while not muffling the sound too much like in old BS-Roformer v2 (2024.02) on MVSEP, although it still can be muddy at times (esp. vs MDX23C HQ models), HQ_4 tends to be the least muddy out of all HQ_X models (although not always), and is faster than HQ_3 and below, it tends to have less vocal residues vs MDX23C.
Final MDX-Net HQ_5 seems to be muddier for instrumentals, although slightly less noisy, but better for vocals than HQ_4.

0d) MDX HQ_5 final model in UVR (available in its Download center and [Colab](#))
Versus HQ_4, less vocal residues, but also muddier at times and a bit lower, 21,5kHz cutoff.
Sometimes even more muddy than narrowband inst 3 to the point it can spoil some hi hats occasionally.
Versus unwa's v1e “HQ5 has less bleed but is prone to dips in certain situations. (...) Unwa has more stability, but the faint bleed is more audible. So I'd say it's situational. Use both. (...) Splice the two into one track depending on which part works better in whichever part of the song is what I'd do.” CC Karaoke

[Model](#) | config: "compensate": 1.010, "mdx_dim_f_set": 2560, "mdx_dim_t_set": 8,
"mdx_n_fft_scale_set": 5120

0d) MDX HQ5 beta model on uvronline via special link for: [free/premium](#) (scroll down)
Go to "music and vocals" and there you will see it
It's not a final model yet, the model was in training since April.
It seems to be muddier than HQ_4 (and more than Kim's and MVSEP's Mel-Roformer), it has less vocal bleeding than before, but more than Kim Mel-Roformer.
“Almost perfectly placed all the guitar in the vocal stem” it might get potentially fixed in the final version of the model.

0e) Other single MDX23C full band models on mvsep.com (queues for free unregistered users can be long)

(SDR is better when three or more of these models are ensembled on MVSEP; alternatively in UVR 5 GUI's via "manual ensemble" of single models (worse SDR) or at best, weighted manually e.g. in DAW, but the MVSEP "ensemble" option is specific method - not all fullband MDX23C models on MVSEP, that's including 04.24 BS-Roformer model are available in UVR)

- BS-Roformer model ver. 2024.04.04 on MVSEP (further trained from viperx' checkpoint on a different dataset). SDR vocals: 11.24, instrumental: 17.55 (vs 17.17 in the base viperx model). Bad on sax. Less muddy than the three below.

Though, all might share same advantages and problems (filtered results, muddiness, but the least of residues)

- Mel-Roformer model ver. 2024.08.15 on MVSEP (fine-tuned on prob. Kim's model)

- BS-Roformer 12xx models by viperx model in UVR [beta](#)/MVSEP and x-minus (struggles with saxophone too, but less (also vs Gabox inst v6), also struggles with some Arabic guitars, bad on vocoders)

"does NOT pick up on large screams that much (example being Shed by Meshuggah in my tests), well at least [vs] [kim's] x-minus mel-rofo"

1297 variant is being used on x-minus. It tends to be better for instrumentals than the 1296 model.

- Older BS-Roformer v2 model on MVSEP (2024.02) (a bit lower SDR)

All vocal Roformer models may sound clean, but filtered at the same time - a bit artificial [it tends to be characteristic of the arch], but great for instrumentals with heavy compressed vocals and no bass and drums - the least amount of residues and noise - very aggressive.

- old MelBand Roformer model on MVSEP (don't confuse with the Kim's one x-minus - they're different)

- [GSEP](#) (now paid) -

Inst fullness: 28.83, bleedless: 31.18, SDR: 12.59

Check out also 4-6 stem separation option and perform mixdown for instrumental manually, as it can contain less noise/residues vs 2 stem in light mix without bass and drums too (although more than first vocal fine-tunes of like MVSEP's BS-Roformer v2 back then). Regular 2 stem option can be good for e.g. hip-hop, and 4/+ stems a bit too filtered for instrumentals with busy mix. GSEP tends to preserve flute or similar instruments better than some Reformers and HQ_X above (for this use cases, check out also kim inst and inst 3 models in UVR) and is not so aggressive in taking out vocal chops and loops from hip-hop beats. Sometimes might be good or even the best for instrumentals of more lo-fi hip-hop of the pre 2000s era, e.g. where vocals are not so bright but even still compressed/heavily processed/loud or when instrumental sound more specific to that era. For newer stuff from ~2014 onward, it produces vocal bleeding in instrumentals much sooner than the above

models. "gsep loves to show off with loud synths and orchestra elements, every other mdx v2/demucs model fail with those types of things".

Older ensembles (among others from the [leaderboard](#))

Q: How to ensemble BS-Roformer 1296 with Kim Mel-Roformer using UVR GUI?

I choose max/max vocal/instrumental, but on the list there is only 1296, and no Kim Mel-Roformer like in MDX-Net option [might have been fixed already]

A: "You have to set the stem pair to multi-stem ensemble, it can generate both vocal and instrumental from both models at the same time. Be sure to set the algorithm to max/max. Once that's done, find the ensemble folder and put the two instrumental files/two vocal files onto the input, provided that you have to go to audio tools first. Then set the algorithm to average and click on the start processing button" - imogen

0f. [#4626](#):

MDX23C_D1581 + Voc FT

0g) [#4595](#):

MDX23C_D1581 + HQ_3 (or HQ_4 now)

0h) Kim Vocal 2 + Kim Inst (a.k.a. Kim FT/other) + Inst Main + 406 + 427 + htodemucs_ft (avg/avg)

0i) Voc FT, inst HQ3, and Kim Inst

0j) Kim Inst + Kim Vocal 1 + Kim Vocal 2 + HQ 3 + voc_ft + htodemucs ft (avg/avg).

0k) MDX23C InstVoc HQ + MDX23C InstVoc HQ 2 + MDX23C InstVoc D1581 + UVR-MDX-NET-Inst HQ 3 (or HQ 4)

"A lot of that guitar/bass/drum/etc reverb ends up being preserved with Max Spec [in this ensemble]. The drawback is possible vocal bleed." ~Anjok

0l) MDX23C InstVoc HQ + MDX23C InstVoc HQ 2 + UVR-MDX-Net Inst Main (496) + UVR-MDX-Net HQ 1

"This ensemble with Avg/Avg seems good to keep the instruments which are counted as vocals by other MDXv2/Demucs/VR models in the instrumental (like saxophone, harmonica) [but not flute in every case]" ~dca100fb8

0m) MDX23C InstVoc HQ + HQ4

0n) [Ripple](#) (no longer works) / Capcut.cn (uses SAMI-ByteDance a.k.a. BS-Roformer arch) - Ripple is for iOS 14.1 and US region set only - despite high SDR, it's better for vocals than instrumentals which are not so good due to noise in other stem (can be alleviated by decreasing volume by -3dB).

0n) Capcut (for Windows) allows separation only for the Chinese version above (and returns stems in worse quality). See [more](#) for a workaround. Sadly, it normalizes input already, so -3dB trick won't work in Capcut. Also, it has worse quality than Ripple

The best single MDX-UVR non-Roformer models for instrumentals explained in more detail ([UVR 5 GUI](#)/Colabs/MVSEP/x-minus):

0. full band MDX-Net **HQ_4** - faster, and an improvement over HQ_3 (it was trained for epoch 1149). In rare cases there's more vocal bleeding vs HQ_3 (sometimes "at points where only the vocal part starts without music then you can hear vocal residue, when the music starts then the voice disappears altogether"). Also, it can leave some vocal residues in fadeouts. More often instrumental bleeding in vocals, but the model is made mainly for instrumentals (like HQ_3 in general)

0b) full band MDX-Net HQ_5 - similarly fast, might be less noisy, but more muddy, although better for vocals, but "it seems it's the best workaround when there is vocal bleed caused by Roformers"

1. full band MDX-Net **HQ_3** - like above, might be sometimes simply the best, pretty aggressive as for instrumental model, but still leaving small amounts of vocal residues at times - but not like BS-Roformer v2/viperx, so results are not so filtered like in these. HQ_3 filters out flute into vocals. Can be still useful to this day for specific use cases "the only model that kept some gated FX vocals I wanted to keep". It all depends on a song, what's the best - e.g. the one below might give better clarity:

2. full band **MDX23C-InstVoc HQ** (since UVR 5.60; 22kHz/fullband as well) - tends to have more vocal residues in instrumentals, but can give the best results for a lot of songs. Added also in MDX23 [2.2.2](#) Colab, possibly when weights include only that model, but UVR's implementation might be more correct for only that single model. Available also in [KaraFan](#) so it can be used there only as a solo model.

2b. MDX23C-InstVoc HQ 2 - worse SDR, sometimes less vocal residues

Older MDX models

2c. narrowband **MDX23C_D1581** (model_2_stem_061321, 14.7kHz) - better SDR vs HQ_3 and voc_ft (single model file [download](#) [just for archiving purposes])
"really good, but (...) it filters some string and electric guitar sounds into the vocals output" also has more vocal residues vs HQ_3.

- *. narrowband **Kim inst** (a.k.a. "ft other", 17.7kHz) - for the least vocal residues than both above in some cases, and sometimes even vs HQ_3
- *. narrowband **inst 3** - similar results, a bit more muddy results, but also a bit more balanced in some cases

- Gabox “[small](#)” inst Mel Roformer model for faster inference than most Roformers | [yaml](#)
Be aware that it can have some audible faint constant residues.

- *. narrowband inst 1 (418) - might preserve hihats a bit better than in inst 3.
- 3. narrowband voc_ft - sometimes can give better results with more clarity than even HQ_3 and kim inst for instrumentals, but it can produce more vocal residues, as it's typically a vocal model and that's how these models behave in MDX-Net v2 arch (you can use it e.g. as input for Matchering for cleaner, but more muddy model result)
- *. less often - inst main (496) [less aggressive vs inst3, but gives more vocal residues]
- *. or eventually also try out HQ_1 - (epoch 450)/HQ_2 (epoch 498) or earlier 403, 338 epochs, or even 292 is also used frequently from time to time) when VIP code is used.

[*Recommended MDX and Demucs parameters in UVR*](#)

- Ensemble of only models without bleeding in single models results for specific song
- [DAW ensemble](#) of various separation models - import the results of the best models into DAW session set custom weights by changing their volume proportions
- Captain Curvy method:
"I just usually get the instrumentals [with MDX23C] to phase invert with the original song, and later [!] clean up [the result using] with voc ft"

[*How to check whether a model in UVR5 GUI is vocal or instrumental?*](#)

(although in MDX23C there is no clear boundary in that regard)

> **for vocals**

(click [here](#) for Karaoke, or [here](#) for instrumentals)

MVSEP models without download links can be used only on MVSEP

(removing/isolating vocals from AI music can give muddy results and capture other unrelated instruments easily; also, Roformers tend to stress plosives which weren't in the original vocals at time - cristouk)

* - commonly used public models at the moment

There's no one, the best model. It depends on a song.

Most commonly used models for the doc's date (categorized below):

BS-Roformer 2025.07, Big Beta 6X/6, vocfv7beta1 & 2, voc_fv4, Big Beta 5e

Resurrection (voc. variant below), Revive 3e/2.

Anvuev BS-Roformer vocals, Mel FT2 Bleedless, voc_fv6, voc_fv5, Beccruily voc, FT3

Preview.

Bleedless models #1

- BS-Roformer 2025.07 only on MVSEP - free with longer queue

Vocals bleedless: 38.25, fullness: 17.23, SDR: 11.89

The biggest bleedless metric for a single model so far. Compared to previous models, picks up backing vocals and vocal chops greatly where 6X struggles, and fixes crossbleeding and reverbs where in some songs previous models struggled before.

Sometimes you might still get better results with Beta 6X or voc_fv4 (depending on a song).

"Very similar to SCNet very high fullness without the crazy noise" - dynamic64, "handles speech very well. Most models get confused by stuff like birds chirping (they put it in the vocal stem), but this model keeps them out of the vocal stem way more than most. I love it!" Works the best for orchestral choirs out of the long [list](#) of other models (.elgiano).

It can be better for metal both for vocal and instrumentals than the mesk's models, a lot of the times (and sometimes the best).

The first iteration of the model (2025.06: 37.83/17.30/11.82) received two small updates and was replaced by 2025.07.

- Mel-Roformer 2024.10 (Bas Curtiz model fine-tuned by ZFTurbo) on MVSEP

Vocals bleedless: 37.80, fullness: 17.07, SDR 11.28

Small amounts of bleeding from instrumentals (inst. bleedless 39.20), might struggle with flute occasionally, good enough for [creating RVC datasets](#).

- Mel-Roformer Bas Curtiz edition (/w Marekkon5) (trained on also ZFTurbo dataset) on MVSEP (older version of 2024.10 model)

Vocals bleedless: 39.20, fullness: 16.24, SDR 11.18.

- Unwa Kim Mel-Band Roformer Bleedless FT2 | [download](#) | [Colab](#) | [Huggingface / 2](#) | [Kaggle](#) | [UVR instruction](#)

Vocals bleedless 39.30 (better than Mel-Kim), fullness 15.77 | SDR 11.05

(voc. fullness is worse than Mel Kim - 16.26,

inst. bleedless is still lower than base Mel-Kim model: 46.30 vs 46.56)

"I usually use big beta 6x, big beta 5e if that fails and FT2 bleedless if I want very low noise or instruments are quiet (it gets muddy quick)" - Rainboom Dash

- Anvuew BS-Roformer vocal model | [download](#)

Doesn't work on the UVR's RTX 5000 patch - then use [MSST](#) instead.

Can be muddy. Not so balanced like Beta6X or vocf7beta1, but "it properly doesn't capture the instrument [here](#). Even FT2 bleedless gets tricked by this part, but this does just fine." - rainboomdash

- Unwa's BS-Roformer [Resurrection](#) | [yaml](#) | [Colab](#) *

Vocal bleedless: 39.99, fullness: 15.14, SDR: 11.34

Shares some similarities with the SW model, including small size (might be a retrain). The default chunk_size is pretty big, so if you run out of memory, decrease it to e.g. 523776.

- Unwa's [Revive 2](#) BS-Roformer fine-tune of viperx 1297 model | [config](#) | [Colab](#)

Voc. **bleedless**: 40.07, fullness: 15.13, SDR: 10.97

"has a Bleedless score that surpasses the FT2 Bleedless"

"can keep the string well"

It's depth 12 and dim 512, so the inference is much slower than some newer Mel-Roformers.

- BS-Roformer 2024.08 (viperx model fine-tuned v2 by ZFTurbo) on MVSEP

Vocals *bleedless*: 37.61, fullness: 15.89, SDR: 11.32

Good for inverts, Dolby, lots of harmonies, BGVs. Good or even the best vocal fullness for some genres ~Isling, decent all-rounder, but might be muddier than Mel models here, although it gives less vocal residues than all the Mel Kim fine-tune models here, can be also used for RVC). "I've found it very useful for extremely quiet vocals that Mel couldn't extract" - Dry Paint Dealer. It's a second MVSEP's fine-tune of viperx model.

Iirc, it's used as a preprocessor model for "Extract from vocals part" feature on MVSEP.

- MVSep Ensemble 11.93 (vocals, instrum) (2025.06.28) - only for paid premium users

Vocals bleedless: 36.30, fullness: 17.73, SDR: 11.93

Surpassed sami-bytedance-v.1.1 on the multisong dataset SDR-wise.

- BS-Roformer SW 6 stem ([MVSEP](#), [Colab](#)) / Vocals only *

Vocals bleedless: 36.06, fullness: 16.95, SDR 11.36

Good for some deep voices.

Bleedless #2 (less)

- Unwa Mel-Roformer Big Beta 6X vocal [model](#) | [yaml](#) | [Colab](#) | AI Hub [Colab](#) | [Huggingface](#) | [uvronline](#)

voc bleedless: 35.16, fullness: 17.77, SDR: 11.12

"it is probably the highest SDR or log wmse score in my model to date."

"There's some noise audible, it doesn't sound as clean when you compare to a more bleedless model (...) but it's certainly not fullness... (...) I think calling it bleedless wouldn't be crazy... makes more sense than "middle of the road" - rainboomdash

"Significantly better" than 5e for some people, although slower. Some leaks into vocal might occur, plus "The biggest problem with the model is the remaining background noise. If it were cleaner, it would already be an almost perfect result." - musictrack

"6X has a lot less noise on vocals, but it's pretty muddy. I would prefer something in between [5e and 6X]. I tried to apply the phase [fixer/swapper] to the vocals and the noise was reduced, but only slightly." - Aufr33

Some people might prefer fv5 instead [at least on some songs] ~5b

"6x is picking up BV just fine, where voc fv4 is failing" - Rainboom Dash

Training details:

"dim 512, depth 12. It is the largest Mel-Band Roformer model I have ever uploaded." - "the same as Bas Curtiz Edition" model. It has a bigger SDR vs smaller depth 6 Big Beta 6 model. "I've added dozens of samples and songs that use a lot of them to the dataset"

Fullness models

- * Unwa [bs_roformer_revive3e](#) | [config](#) | [Colab](#)

voc bleedless: 30.51, **fullness**: 21.43, SDR: 10.98'

"A vocal model specialized in fullness.

Revive 3e is the opposite of version 2 — it pushes fullness to the extreme.

Also, the training dataset was provided by Aufr33. Many thanks for that." - Unwa
"seems to sound better than beta5e, it sounds fuller, but this also means it sounds noisier" - gilliaan. For some people, it's even the best.

More fullness less bleedless

- * Gabox experimental Mel-Roformer voc_fv6 [model](#) | [yaml](#) | [Colab](#)

voc bleedless: 26.61, **fullness**: 24.93, SDR: 10.64

"Definitely not bleedless" - rainboomdash, "Sounds like b5e with vocal enhancer. Needs more training, some instruments are confused as vocals" - Gabox. "fv6 = fv4 but with better background vocal capture" - neoculture

"very indecisive about whether to put vocal chops in the vocal stem or instrumental stem. sometimes it plays in vocals and fades out into instrumental stem and sometimes it just splits it in half kinda and plays in both at the same time lol" - lsling

"I think is the fullest vocal model I've heard, aside from maybe the scnet high fullness ones lol/ Oh and revive 3e and b5e are full too but yeah." - Musicalman

- SCNet XL very high fullness on MVSEP

voc bleedless: 25.30, fullness: 23.50, SDR: 10.40

- SCNet XL IHF (high instrum fullness by bercuily)

voc bleedless: 25.48, fullness: 22.70, SDR: 10.87

(it was made mainly for instrumentals, but “It can also be an insane vocal model too”)

- MVSEP SCNet XL IHF

voc bleedless 28.31, fullness 17.98, SDR: 11.11

“It has a better SDR than previous versions. Very close to Roformers now.” also, vocal bleedless is the best among all SCNet variants on MVSEP. Metrics. IHF - “Improved high frequencies”.

“Certainly sounds better than classic SCNet XL (...) less crossbleeding of vocals in instrumental so far, and handle complex vocals better” - dca

Middle of the road #1 (lower fullness)

- Gabox vocfv7 beta 2 Mel-Roformer [model](#) | [yaml](#) | [Colab](#)

voc bleedless: 31.55, fullness: 20.44, SDR: 10.87

“fullness went down a little bit” vs beta 1 (...) Definitely an improvement over fv4 (...) still quite a bit fuller than big beta 6x, but has less noise than even fv4 (...) at least when the instruments are loud, fv7beta2 is usually quite a bit less noisy than fv4, while still maintaining a decent amount of fullness... it is a bit less, but not too much (...) both are pretty noisy with fv4 (...) sometimes the noise can be pretty significant with fv7beta1, and fv7beta2 may have the fullness you desire. (...) “I’m really liking the balance of fullness and noise for most songs. fv4 and fv6/fv7beta1 are usually pretty noisy... this is less noisy, but still has a good amount of fullness.” still gonna have an issue with backing vocals compared to fv7beta1 sometimes... (...) “Fv7beta2 has still been significantly better with BV than fv4, despite quite a bit less noise” but “significant issues on one song, while fv6/fv7beta1 didn’t” - rainboomdash

- Gabox [vocfv7beta3](#) Mel-Roformer | [yaml](#) | [Colab](#)

voc bleedless 30.83, fullness 21.82, SDR 10.80

“beta 1 and 2... eh, pretty close to same instrumental bleed, but beta 3 def a step up from the two songs I compared (...) most songs so far, fv7beta3 is fuller than fv7beta1, def less robotic sounding at times (when a voice gets quiet/hard to capture, and it just fails). Just had another song where fv7beta1 was fuller than fv7beta3, but it was also a lot noisier large majority of the songs I tested, fv7beta3 was fuller... I think fv7beta3 is usually a bit noisier than fv7beta1? But also sounds fuller in those cases, I’d say it’s generally worth it instrumental bleed, usually worse with fv7beta3 versus fv7beta1, but it depends fv7beta2 is always less full/less noise, but only slightly less instrumental bleed than fv7beta1” - rainboomdash

- Gabox Mel-Roformer voc_fv7 beta 1 (a.k.a. vocfv7beta1) [model](#) | [yaml](#) | [Colab](#)

voc bleedless: 30.81, fullness: 21.21, SDR: 10.96

"one step below the extreme fullness models (...) fv6 on average is more full" - rainboomdash. "Just a better fv4 it seems, better bleedless" (fullness: 21.33, bleedless: 29.07, SDR 10.58)

vs voc_fv4 "It is noisier... Kinda closer to beta 5e?" "It's slightly less noise and fullness than beta 5e but picking up the backing vocals REALLY well, significantly better than beta 5e" But it's pulling the backing vocals out even better than 5e" "the backing vocals are so good! "it does have significant synth bleed, too... it at least wasn't coming through at full volume when I say fullness, I specifically mean how muddy it sounds" - Raiboom Dash

- Gabox Mel-Reformer [voc_fv4](#) | [yaml](#) | [Colab](#) | [Huggingface / 2](#) *

voc bleedless 29.07, fullness 21.33, SDR 10.58

"Very clean, non-muddy vocals. Loving this model so far" (mrmason347)

Good for anime and **RVC** purposes, currently the best public model for it (codename)

"The important thing for an RVC dataset is to get lead vocals so fv4 is good for that

The newer karaoke models are also helpful" - Ryan

Some might prefer voc_gabox2 instead, occasionally - chroniclaugh.

The opposite of Beta6x which has "lower noise but [is] less full/muddier (...) noise/muddiness seems between 6x and 5e, but even 6x is picking up BV just fine, where voc fv4 is failing"

Some people might want to test it with even overlap 32, and then:

"It's close to perfect, the only thing is it kinda struggled with picking up the adlibs and the delay, but the lead vocal is almost perfect I think. (...) on another song (...) 5e is just too noisy and 6x is muddy, fv4 is best of both worlds (...) has segments with constant significant vocal bleed (for the most part, it's not audible at all) (...) I was trying to get an acapella and every model failed except this one. It's not perfect, but I guess some songs are just too hard for the AI." - Rainboom Dash

Good also for instrumentals, if you need less vocal residues than typical instrumental

Reformers (even less than Mel Kim, FT2 Bleedless, or Beta 6X - makidanyee.

"even beta 6x is a lot better at pulling that background vocal out than voc fv4...

and that's a less full model. hmm, fv6 is noisier and also not picking up the backing vocals as full as the last mel band roformer" - Rainboom Dash

- Unwa Mel big beta 5e vocal model | [Colab](#) | [Huggingface / 2](#) | MVSEP | uvronline via special link [free/premium](#) | [MSST-GUI](#) | [UVR](#)

Model [files](#) | yaml: big_beta5e.yaml or [fixed](#) yaml for AttributeError in UVR

voc bleedless: 32.07, fullness: 20.77 (the biggest for now), vocals SDR: 10.66

"feel so full AF, but it has noticeable noise similar to lew's vocal enhancer"

You can alleviate some of this noise/residues by using phase fixer/swapper and using becruily vocals model as reference (imogen).

It seems to fix some issues with trumpets in vocal stem - maxi74x1.

"It's noisy and, IDK, grainy? When the accompaniment gets too loud. (...) Definitely not muddy though, which is a welcome change IMHO. I think I prefer beta 4 overall" - Musicalman "ending of the words also have a robotic noise" - John UVR

"Perhaps a phase problem is occurring" - unwa. Phase swapper doesn't fix the issue (it works for inst unwa's models).

If you try big beta 5e on a song that has lots of vocal chops, the vocal chops will be phasing in and out and sound muddy (Isling).

“Excellent for ASMR, for separating Whispers and noise, the quality is super good
That's good when your mic/pc makes a lot of noise. All the denoise models are a bit too harsh for ASMR (giliaan)”

Worse for RVC than Beta 4 model below (codename/NotEddy)

- Mel-Roformer vocal by becruily [model](#) | [config](#) for ensemble in UVR | MVSEP | [Colab](#)
voc bleedless: 31.26, fullness: 20.72 (on pair with 5e), SDR: 10.55 | [Huggingface / 2](#)
Lower bleedless than 5e, “pulling almost studio quality metal screams effortlessly, wOw ive NEVER heard that scream so cleanly”
(on older UVR beta patches) If you use lower dim_t like 256 at the bottom of config for slower GPU these are the first models to have muddy results with it.
Consider setting 485100 chunk_size in the yaml for the highest SDR.
Currently used on x-minus/uvronline as a model for phase fixer.

- Gabox Mel-Roformer [voc_fv5](#) | [yaml](#) | [Colab](#)
voc bleedless: 29.50, fullness: 20.67, SDR: 10.56
“fv5 sounds a bit fuller than fv4, but the vocal chops end up in the vocal stem. In my opinion, fv4 is better for removing vocal chops from the vocal stem” - neoculture. [Examples](#)

Other/older models

- Gabox Mel-Roformer [voc_gabox2](#) model | [yaml](#) | [Colab](#)
Vocal bleedless: 33.13, fullness: 18.98, SDR: 10.98

- Gabox Mel-Roformer Vocal F (fullness) v3 [model](#) | [Colab](#) | [Huggingface / 2](#)
voc bleedless: 32.15, fullness 19.97

- Gabox Mel-Roformer Vocal F (fullness) v2 [model](#) | [Colab](#) | [Huggingface / 2](#)
voc bleedless: 33.40, fullness: 19.31

- Aname Mel [FullnessVocalModel \(yaml\)](#) model | [Colab](#) | [Huggingface / 2](#)
Vocals bleedless: 32.98 (less than beta 4), fullness: 18.83 (less than big beta 5e/voc_fv4/becruily, more than beta 4)

- Gabox Mel-Roformer voc_gabox (Kim/Unwa/Becruily FT) [model](#) | [Colab](#) [Huggingface / 2](#)
voc bleedless: 34.66 (better than 5e, beta 4 and becruily voc), fullness 18.10 (on pair with beta 4, worse than 5e and becruily)

- Mel-Roformer unwa's beta 4 (Kim's model fine-tuned) [download](#) | [Colab](#) | [Huggingface / 2](#)
Vocals bleedless: 33.76, fullness: 18.09
“Clarity and fullness” - even compared to newer models above.

Beta 1/2 were more muddy than Kim's Roformer, potentially a bit less of residues, a bit more artificial sound. Ringing issues in higher frequencies fixed in beta 3 and later. It's good for RVC (and favourite codename's public model for RVC before voc_fv4 was released). Fuller vocals than Bas Curtiz FT on MVSEP (but can bleed more synths) ~becruily Unwa's vocal models are capable of handling sidechain in songs - John UVR

Bleedless models #2

- BS-Roformer Revive unwa's vocal [model](#) experimental | [yaml](#)
(viperx 1297 model fine-tuned)

Voc. bleedless: 38.80, fullness: 15.48, SDR: 11.03

"Less instrument bleed in vocal track compared to BS 1296/1297" but it still has many [issues](#), "has fewer problems with instruments bleeding it seems compared to Mel. (...) 1297 had very few instrument bleeding in vocal, and that Revive model is even better at this. Works great as a phase fixer reference to remove Mel Roformer inst models noise" (dca)

- SYHFT V5 Beta - only on x-minus/uvronline (still available only with [this](#) link for premium users, and for [free](#))

Vocal bleedless: 37.27, fullness, 16.18, SDR: 10.82

Other models #2

- Unwa's Kim Mel-Band Roformer FT2 | [model](#) | [Colab](#)

Vocals bleedless: 37.06, fullness: 16.61 (fullness worse vs the previous FT, but both metrics are better than Kim's)

It tends to muddy instrumental outputs at times, similarly like the OG Kim's model was doing, which didn't happen in the previous FT below. [Metrics](#)

- Unwa Kim Mel-Band Roformer FT3 Preview | [model](#) | [yaml](#) | [Colab](#) | uvronline via special link for: [free/premium](#) (scroll down)

Vocal bleedless: 36.11, fullness: 16.80, SDR: 11.05

"primarily aimed at reducing leakage of wind instruments to vocals."

For now, FT2 has less leakage for some songs (maybe till the next FT will be released)

- Unwa's Mel Big Beta 6 vocal [model](#) | [yaml](#) | [Colab](#) | [Huggingface / 2](#) | AI Hub [Colab](#)

Similar to FT series. "Although it belongs to the Big series, the characteristics of the model are similar to those of the FT series. (...) this model is based on FT2 bleedless with the dim increased to 512".

Muddier than Big Beta 5[e], might be better than FT2 at times.

"If you liked the output of the Big Beta 5e model, you may not like 6 as much; it does not have the output noise problem of 5e, but instead sacrifices Fullness. (...) Simply put, it is a more conservative model" (unwa)

For anime and RVC “isn't as audibly and spectrally full as fv4 + can at times have flat-line artifact at the very top, but then, fv4 can sometimes have “crunchy” noise present at some places, so an ensemble of those 2 is probs a good idea (or might be fv4 flash more on less aggressive scenes).” codename

- Unwa's Kim Mel-Band Roformer FT vocal [model](#) | [Colab](#)

Enhanced both voc bleedless 36.75 (vs 36.95) and fullness 16.40 (vs 16.26) [metric](#) for vocals vs the original Mel Kim model. [SDR](#)-wise it's a tad lower (10.97 vs 11.02).

Tips for separating vocals

- Separate with becruily Mel Vocal model and its instrumental model variant, then get vocals from the vocal model, and instrumental from instrumental model, import both stems for the DAW of your choice (can be Audacity) so you'll get a file sounding like original file, then export - so perform a mixdown of both stems, then separate it with vocal model (mrmason347 /Havoc)
- “In my testing, I've found that SCNet very high fullness (on MVSEP) put through Mel-Roformer denoise (average) and UVR denoise (minimum) has the best acapella result” dynamic
- You can consider using Lew vocal enhancer v1 ([model](#) | [config](#)), v2 ([model](#) | [config](#)) and added to the [Colab](#) ([this](#) now probably works instead), and it also can be used in the latest UVR Roformer [beta](#).
- Sometimes using EQ stressing vocals properly might be beneficial for separation too
- You might potentially also try to experiment with demudder added with the beta patch #14 linked above. Normally demudder works only for instrumentals, but when you switch in the config editor to vocal stem being instrumental and in reverse, then demudder will work vocals. If your model have “other” stem instead of “instrumental” or “vocal”, you'll need to rename it. Demudder requires stem labelled as instrumental to work with.

Ensembles

(for vocals)

- BS Revive 3e + BS 2025.07 (Max FFT) (“the best Ensemble for vocals for now”) (dca100fb8)
- Mel Becriuly Vocal + MVSEP's BS 2025.07 (Max FFT) (former “best vocal ensemble”) (-||-)

- unwa's bigbeta5 + becruily vocal - Max spec (midol)
- voc_gaboxFv2 + becruilys vocal (heauxdontlast /gilian)
- Unwa "Big Beta 4 + Big Beta 5e - Average Spec ("really good to reduce the noise while keeping the fullness") (heauxdontlast)
- unwa beta6 + voc_fv4 (good for anime and **RVC**)
- unwa beta6x + voc_fv4 ("some songs I can use big beta 6x, and it's enough, others I need to ensemble it with voc_fv4") (Rainboom Dash)
- unwa beta6x + voc_fv6 ("would make a good ensemble, but the amount of noise is horrific and I heard that [phase swapper](#) would fix it")
- BSRoformer-Viperx1297, BSRoformer-LargeV1 by Unwa, unwa_ft2_bleedless, mel_band_roformer_vocals_becruily, Gabox voc_fv4 - Average/Average Spec (good for cleaning inverts) (AG89)
- Models ensembled (inst, voc) available for premium users on [mvsep.com](#) (SDR 10.44-11.93 and "High Vocal Fullness" variants)

RVC models choice by [AI Hub](#) (subject to change;
read their current docs too)

If you can separate with these models downloaded from above locally, see also [here](#) for the list of all cloud sites and Colabs.

"If you need to remove multiple noises, follow this pipeline for the best results:
Remove instrumental -> Remove reverb [probably on vocals] -> Extract main vocals -> Remove noise"
 Or also Isling's approach "gives insanely clean results":
Vocals>De-reverb>Karaoke

Vocals

- MelBand Roformer | Vocals FV4 (a.k.a. voc_fv4) by Gabox also
 (Gabox vocfv7beta1 "seems to give better results than fv4", also Mel 2024.10 is mentioned in MVSEP section, but BS-Roformer 2025.07 now has all the metrics better, unwa beta6/x + voc_fv4 ensemble is also good for RVC, unwa beta 4 was better than big beta v5e (NotEddy/codename), research also voc_gabox2)

Instrumentals

- MelBand Roformer | INSTV7 by Gabox
(unwa instrumental v1e+ OR Mel 2024.10 are also mentioned in their MVSEP section and Gabox Fv7z is mentioned in the x-minus)

De-reverb

- MelBand Roformer | De-Reverb by anvuew
(it's probably v2 variant [also mentioned there], or also Sacial V2 (MelRoformer) mentioned in their MVSEP section ["if I'm unhappy with the results I go for Sacial - isling"] - it probably follows the model naming scheme of UVR UI on [HF](#), also the new mono-dereverb model is being used occasionally)

Backing Vocals

- Mel-Roformer-Karaoke-Aufr33-Viperx (surpassed by Becriuly and Frazer Karaoke, but the first can be more consistent; anvuew's Karaoke model have fuller lead vocals; also older Model fuzed gabox & aufr33/viperx (SDR: 9.85) is mentioned in their MVSEP section)

De-noise

- Mel-Roformer-Denoise-Aufr33-Aggr (they mention also "Mel denoiser v2" in UVR section)

Restoration

- For lossy mp3/mixtures: Apollo Universal by Lew (sometimes AudioSR can be better)
- For voice: AP-BWE or ClearerVoice-Studio's Clear Voice "my favorite is the 2nd one" - codename0)

Fast inference models

Above an hour on i3-7100u, rather light, small - the lightest Reformers, while most have 870 MB):

For vocals

- [Unwa Resurrection](#) BS-Roformer ([yaml](#) | [Colab](#), 195 MB)
- BS-Roformer SW vocals only (mask_estimators.0 on the regular 6 stem model, 195 MB)

Older models

- [Aname Mel-Roformer small](#) (203MB)
- [Unwa Mel-Roformer small](#) (203MB)

Older arch (faster; 25-60 minutes+ on weak i3u/C2Q respectively)

- voc_ft (probably the fastest, but uses outperformed MDX-Net v2 arch, also it's narrowband)
- Kim Vocal 2 (or ev. 1, -||-, older model)

For instrumentals

- Unwa [BS-Roformer Resurrection inst \(yaml\)](#) | a.k.a. "unwa high fullness inst" on MVSEP | uvronline [free/premium](#) | [Colab](#) | [UVR](#) (don't confuse with Resurrection vocals variant, 204 MB)
- Gabox BS_ResurrectioN ([model](#) | [yaml](#), 204 MB)

Older

- Unwa BS-Roformer-Inst-FNO (works only in MSST after modifying py file like in the model card, similar to decently performing Resurrection inst model, 332 MB)
- Gabox Mel-Roformer [small_inst](#) | [yaml](#) (experimental, 203 MB)
- Unwa BS-Roformer-Inst-EXP-Value-Residual (low performance, use v2 model type in UVR)

MDX-Net (faster, usually lower quality)

- MDX-Net HQ_3, 4, 5 (the last is the fastest, 56 MB)
- MDX-Net inst3, Kim inst (older, narrowband, but can be useful too in some cases, 63 MB)

4 stems

- Faster FP16 version of BS-Roformer 6 stems called splifft (by undef13; a tad lower SDR; only 334MB vs 700 MB in the OG weight, CPU/NVIDIA compatible, and potentially AMD ROCm, only bigger variant works in UVR; the OG "Conversion done after 2 hours for a 2 minute 49 second file" on 2/4 i3 7100u) - on CPU it might be slower than the OG, as it might not support FP16 natively due to even possible emulation. But probably Turing GPUs with tensors (e.g. RTX or T4) and newer, probably have FP16 acceleration, while non-RTX 16XX sometimes not.

Faster, lower quality:

- [KUIELab-MDXNET23C](#) (4 stems) - its first scores were probably from ensemble of its five models, and in that configuration it had better SDR than demucs_ft on its own, and drums had better SDR than "SCNet-large_starrytong" (so single models' score of any of these MDX23C models is probably lower than in demucs_ft).

> Lighter "model1" drums sounds surprisingly better than htdemucs non_ft v4 on previously separated instrumental. It handles trap really well and preserves hi-hats correctly, but in cost of other stem bleeding. v4 model can be used to clean it a bit further,

- htdemucs v4 non-ft (UVR default) - it can clean up other stem bleeding of the above
- htdemucs_mmi - probably faster, but worse quality, v3
- kuielab_b - lightning-fast, but quality is mediocre (but rather still better than Spleeter)

Older vocal models (legacy section)

- Mel-Roformer unwa's inst-voc model called "duality v1/2" (focused on both instrumental and vocal stem during training, but you can now test newer V1e+ single stem for this purpose too).

<https://huggingface.co/pcunwa/Mel-Band-Roformer-InstVoc-Duality> | Colab | MVSEP

Vocals sound similar to beta 4 model, but with more noise,
instrumentals are deprived of the noise present in inst v1 and later inst models, but as a
downside, they're more muddy for instrumentals.
v2 have slightly a bit better SDR and fewer residues

Because duality is a two stems target model.

"other" is output from model

"Instrumental" is inverted vocals against input audio.

The latter has lower SDR and more holes in the spectrum.

So, using MSST-GUI, leave the checkbox "extract instrumental" disabled for duality models.
You can use it in the Bas Curtiz' [GUI](#) for ZFTurbo script (already added) or with the OG ZF's
repo, or in the Colab.

- Aname duality Mel [model](#)

- Aname Full Scratch Mel-Band Roformer [model](#)

bleedless 30.75 fullness 13.24, SDR: 8.01

- SYHFT (a.k.a. SYH99999/yukunelatyh) MelBandRoformer V3 | [model](#)

VS previous SYH's models "this version is more consistent with separation. It's not what I'd call a clean model; It sometimes lets background noise bleed into the vocal stem. But only somewhat, and depending on how you look at it, it can be a good thing since it makes the vocals sound less muddy." Musicalman

- MelBandRoformerBigSYHFTV1Fast | [model](#) - more vocal fullness metric, but more bleeding (although less than duality models and even Kim's purely [metric-wise](#)). "same parameters size with Kim's. Other models are 2x scale parameter size to compare my model"

- **Mel-Roformer model by Kim** | [model](#) | [config](#)

Vocals bleedless: 36.75 | fullness: 16.26 | SDR: 11.07

([Colab/Huggingface/2/MVSEP/uvronline](#) via special link for: [free/premium](#) (scroll down)/UVR [beta Roformer](#) (available in Download Center)/[MSST-GUI/simple Colab/CML inference](#))

Usual base for lots of Mel fine-tunes on that list.

Sometimes might leave instrumental residues in vocals, but can be less muddy than other BS-Roformers - the same goes to any fine-tunes of this model vs BS 2024.08, so effectively all the Mel models above)

"godsend for voice modulated in synth/electronic songs" vs 1296 can be more problematic with wind instruments putting them in vocals.

- unwa's instrumental Mel-Roformer v1e+

- unwa's instrumental Mel-Roformer v2 model (similar to v1, but less noise, muddier, bigger, heavier model)

[Model files](#) | [Colab](#) | uvronline via special link for: [free/premium](#) (scroll down) | [MSST-GUI](#)
(It's now included in ZFTurbo's [repo](#), it's the "gui-wx.py" file)

Might miss some samples or adlibs while cleaning inverts. SDR got a bit bigger (16.845 vs 16.595) "Sounds very similar to v1 but has less noise, pretty good" "the aforementioned noise from the V1 is less noticeable to none at all, depending on the track". "V2 is more muddy than V1 (on some songs), but less muddy than the Kim model. (...) [As for V1,] sometimes it's better at high frequencies" Aufr33

- older BS-Roformer 2024.02 on MVSEP (generally BS-Roformer models "can be slappy with choir-like vocals and background vocals" but "hot on pre-2000 rock")
These older Roformers "kinda does poorly on large screams" in metal music, but not always. Sometimes even HQ_4 can catch them better than, e.g. viperx models.
- Mel-Roformer fine-tuned 17.48 model on MVSEP (works e.g. for live shows that have crowd)
(it's different from the one on x-minus)
- Gabox BS-Roformer instrumental, which doesn't struggle so much with choirs like most Mel-Roformers, although it may not help in all cases ([link](#))
- "ver. 2024.04" SDR 17.55 on MVSEP - fine-tuned viperx model v1 (can pick in adlibs better, occasionally picks some SFX', sometimes one, sometimes the other is "slightly worse at pulling out difficult vocals")
- BS-Roformer Large unwa's vocal model (viperx 1297 model fine-tuned) [download](#) | [Colab](#)
More muddy than Kim's Roformer, potentially a bit less of residues, a bit more artificial sound. Better than viperx model - "captures more nuances, subtle elements and details" ~A5
It can be better for some older music like The Beatles than above models.
- BS-Roformer viperx 1297 model (UVR beta/MVSEP a.k.a. SDR 17.17 for "1296" variant iirc/called just "BS-Roformer" on uvronline via special link for: [free/premium](#) (scroll down))
- Mel-Roformer viperx 1143 model (UVR>Download More Models)
(don't confuse with 1053 which separates drums and bass in one stem).
The first Mel-Roformer vocal model trained by viperx before Kim model which introduced changes to the config, which fixed the problem of lower SDR vs models trained on BS-Roformer.
Most people back then preferred Kim Mel-Roformer instead, but Mel viperx' "does background voices correctly not unlike Kim's (it does not recognise background 'breee's)" "lirc Viperx Mel Rofo doesn't struggle with instruments counted as vocals".
Also, both Mel and BS variants of viperx model struggle with saxophone and e.g. some Arabic guitars. It can still depend on a song whether these are better than even the second oldest Roformer than on MVSEP (from before viperx model got fine-tuned version). Beside

problems with recognizing instruments, they're very good for vocals (although Mel-Roformer by Kim on x-minus tends to be better).

Muddy instrumentals when not ensembled with other archs (but we didn't have typically instrumental stem target models back then), maybe Mel variant less.

Be aware that names of these models on UVR refer to SDR measurements of vocals conducted on private viperx dataset, not even older Synthetic dataset, instead of on multisong dataset on MVSEP, hence the numbers are higher than in the multisong chart on MVSEP.

Depending on a model, Roformers might be muddy. Consider using ensembles or Apollo enhancer model by Lew v2 ([Colab](#)/[Model](#)/[Inference](#)), although it might be noisy. Might work the best on BS+Mel ensembles (max spec, though avg might work better in some cases).

Older ensembles for vocals

- Models ensembled option on x-minus.pro (available only for premium users)
- > Mel-Roformer + MDX23C (can be picked after you uploaded/processed a track [at least with Mel-Roformer model chosen]).
- > Mel-Roformer + demudder
 - "I recommend mel-roformer + demudder to remove vocals from songs that contain only backing vocals that are so faint that our ears can barely hear them."
- MDX23 by ZFTurbo (v. [2.5](#) jarredou Colab fork)
- Ensembles on MVSEP.com (for premium users)
- Ensembles in UVR 5:
 - a) 1296 + 1143 (BS-Roformer in [beta](#) UVR) + Inst HQ4 (dopfunk)
(there might be instrumental residues from HQ4 in some cases)
 - b) 1296 + 1297 + MDX23C HQ
 - c) Manual ensemble in UVR of models BS-Roformer 1296 + copy of the result + MDX23C HQ (jarredou) - for faster result and similar quality vs the one above
- More ensembles beneath
- [KaraFan](#) (preset 4, but may give worse results than Mel-Roformer)

Older single models for vocals (available in [UVR 5](#) | inference [Colab](#) | [MDX-Net](#) | [MVSEP](#))

- **UVR-MDX-Net-Voc_FT** (narrowband, further trained, fine-tuned version of the Kim vocal model; Roformers might be better now)
>If you still have instrumental bleeding, process the result with Kim vocal 2
>Alternatively use MDX23C narrowband (D1581) then Voc-FT, "great combination" (or MDX23C-InstVoc HQ instead of D1581)
(so separate with the D1581 or InstVoc model first, then use the separated result as input, and separate it further with voc_ft)
- Kim Vocal 1 (can bleed less than 2, but more than voc_ft, might depend on a song)
- Kim Vocal 2
>MDX-Net HQ_3/4/5 (HQ_4 can be sometimes not bad on vocals too, even less muddy than voc_ft, though more noisy, and e.g. HQ_3 had more vocal residues than Kim Vocal 2 in general, HQ_5 have stronger and fuller vocals than HQ_4)
>**MDX23C-InstVoc HQ** (can have some instruments residues at times, but it's fullband - better clarity vs voc_ft and Kim Vocal 1/2 -
"This new model is [vs the narrowband vocal models], by far, the best in removing the most non-vocal information from an audio and recovering formants from buried passages... But in some cases, also removes some airy parts from specific words, and some non-verbal sounds (breathing, moaning)."
- newer MDX23C epochs available on MVSEP like 16.66.
MDX23C models are go-to models for live recorded vocals
(available also in MDX23 Colab v2.3/2.4 when weight set only for InstVoc model)

Older UVR ensembles (from before Roformer models release)

>Voc FT + MDX23C_D1581 (avg/avg)
>292, 496, 406, 427, Kim Vocal 1, Kim Inst + Demucs ft ([#1449](#))
>Kim Inst, Kim Vocal 1 (or/and voc_ft), Kim Vocal 2, UVR-MDX-NET Inst HQ 2 (or 3/4), UVR-MDX-NET_Main_427, htdemucs_ft (avg/avg IRC)
>Kim Vocal 1+2, MDX23C-InstVoc HQ, UVR-MDX-NET-Voc_FT
(jaredou)
> [More ensembles](#)
>You can also check some ensembles for [Instrumentals](#)

Your choice of the best vocal models only (up to 4-5 max for the best SDR - [more](#))

If your separation still bleeds, consider processing it further with models in [Debleeding](#) section further below.

Other services (multipurpose)

- [Ripple](#) (no longer works; since BS-Roformer models release it might be obsolete; it's very good at recognizing what is vocals and what's not and tends to not bleed instrumental into vocal stem; very good if not the best solutions for vocals)

- music.ai (paid; presumably in-house BS-Roformer models)

"almost the same as my cleaned up work (...) It seems to get the instrument bleed out quite well")

"Beware, I've experienced some very weird phase issues with music.ai. I use it for bass, but vocals are too filtered/denoised IMO, and you can't choose to not filter it all so heavily. " - Sam Hocking

- <https://myxt.com/> (paid; uses Audioshake)

- moises.ai (paid; uses in-house BS-Roformer models, sometimes better results than the one on MVSEP)

- ZFTurbo's VitLarge23 e.g. on MVSEP or 2.3/2.4 Colab (it's based on a new transformers arch. SDR-wise it's not better than MDX23C (9.78 vs 10.17), but works "great" for an ensemble consisting of two models with weights 2, 1. It's been added in 4 models ensembled on MVSEP (although the bag of current models is a subject to change any time))

- ZFTurbo's Bandit Plus (MVSEP)

Other decent single UVR models

- Main ([427](#)) or 406, 340, MDXNET_2_9682 - all available in UVR5, some appear in download center after entering [VIP](#) code)

- or also instrumental models: Kim Inst and HQ_3 (via applied inversion automatically)

Other models

- ZFTurbo's [Demucs v4 vocals 2023](#) (on MVSEP, unavailable in Colab, good when everything else fails)

- MDX23 Colab fork [2.1](#) / [2.2](#) (this might be slow) / [2.3](#) / [2.4](#) / [2.5](#) (it's generally better than UVR ensembles SDR-wise, but it's not available in UVR5) (MDX23 Colab is good also for instrumentals and 4 stems, very clean, sometimes more vocal residues in specific places vs single MDX-UVR inst3/Kim inst/HQ models, but it sounds better in overall, especially the Colab modification/fork with fixes made by jarredou)

- HQ_3 (inverted result giving vocals from instrumental in 2nd stem) - more instrumental residues than e.g. Kim Vocal 2, but no 17.7 cutoff)

- Narrowband MDX23C_D1581 "Leaves too much instrumental bleeding / non-vocal sounds behind the vocals. Formants are less refined than on any of the top vocal models (Voc FT, Kim 1, Kim 2 and MDX23C-InstVoc HQ)."

- Kavas' methods for HQ vocals:

Ensemble (Max/Max) - Low pass filter (brickwall) at 2k:

- MDX23C

- Voc FT

Voc FT - High Pass Filter (brickwall) at 2k

("Sometimes it leaves some synth bleeding in the mids" then try out min/min)

Or:

Multiband EQ split at 2kHz with a low & high pass brickwall filter with:

-MDX23C-InstVoc from 0 to 2kHz and:

-Voc_FT from 2kHz onwards

(InstVoc gives fuller mids, but leaves transients from hats in the high end, whereas Voc ft lacks the mids, but gets rid of most transients. Combine the best of both for optimal results.)

- Any [top](#) ensemble or AI appearing on MVSEP leaderboard (but it depends, - sometimes it can be better for instrumental, sometimes vocals)

Ensembles are resource consuming, no cutoff if one model is fullband and the other is narrowband. Random ensembles can result in more vocal or instrumental residues, as mentioned above.

Models not exclusive for MVSEP are all available in [UVR5 GUI](#), or optionally you can separate MDX models in [Colab](#) and perform manual ensemble in UVR5 (no GPU or fast CPU required for this task) or use manual ensemble in [Colab](#) [may not work anymore]) or also in DAW by importing all the stems together and decreasing volume (you might want to turn on limiter on the sum).

Speech separation

"There isn't one specifically trained for anime, try your luck with the current available models"

The list by Musicalman (mostly from before the Gabox models release, check [vocals](#) too)

"Any vocal model in the past few years should work for speech separation. My favorites at the moment are:

- MDX23c Inst-Voc HQ

- other similar MDX models for least aggressive, but bleed, only really useful for denoising

- Unwa's Mel-Roformer big beta 4 or beta 5e vocal models - for less bleed. Atm, 5e is my go-to as it sounds less filtered.

~ I've heard people praise BS-Roformers a lot, haven't really tested those much, though.

- Bebruily's vocal model can also be better at SFX separation, but can overestimate reverb in vocal stem sometimes.

- Mel-Roformer Karaoke by viperx and aufr33 - for more aggressive separation (removes a bit more SFX)

- And the most aggressive are Bandit models and the DNR v3 models on MVSEP, though they tend to be a bit too aggressive for my taste, so I only use them selectively.
This is just my own opinions though, subject to change at a moment's notice lol"
[you'll find them in [SFX](#) section]
- [clearvoice](#) - it's a set of speech enhancement/separation models. My favorite model of the set is MossFormer2_SE_48K. Its dialog extraction seems to be similar to Bandit v2, though clearervoice sounds fuller to me, and separation is usually a bit better. Might be especially good in an ensemble with Bandit or vocal sep models eg. unwa, gabox etc.
- BS-Roformer 2025.06 on MVSEP - "handle speech very well. Most models get confused by stuff like birds chirping (they put it in the vocal stem), but this model keeps them out of the vocal stem way more than most. I love it!"

See also:

- [various speakers isolation](#)
- [harmonies](#)
- [two singers isolation](#)
- [karaoke](#)

Can't find a model?

- Results containing models in e.g. [#946](#) (e.g. 406, 427, 438) or other ensembles mentioned above, still have public models available in UVR, but you can access them by entering the [download/vip code](#) in UVR, so more models will show up

You cannot use VIP code on older beta UVR Roformer patches ([updates](#)), then to use any other VIP model with Roformers (e.g. D1581), you need to install the stable 5.6 from official GH repo, download the model, and update the installation with the old Roformer patch afterwards if you need such version

- Be aware that MDX23C Inst Voc HQ2 is not accessible in beta Roformer patch when VIP code is inserted. You need to [download](#) the model file manually, and paste into models\MDX_Net_Models folder.
(Config is detected automatically, as it uses existing model_2_stem_full_band_8k config - the same as for Inst Voc HQ)

- UVR Denoise non-lite model disappeared from Download Center. Here it is:
https://github.com/TRvlvr/model_repo/releases/download/all_public_uvr_models/UVR-DeNoise.pth

- You cannot use some models from x-minus/uvronline.app or used on MVSEP, e.g. used for Ensemble of 4 and 8 models in UVR, as they contain models not available in UVR, and not available for download. You can only perform manual ensemble of single models processed by MVSEP or x-minus, in UVR, but it will not give the same result as ensemble on MVSEP,

as it uses code more similar to MDX23 Colab code, so sometimes weighted ensemble instead of. e.g. avg spec (don't confuse with MDX23C arch models).

- E.g. for 16.10.23, "MVSep Ensemble of 4" consists of 1648 previous epoch (maybe later updated to 16.66), VitLarge, and Demucs 2023 Vocals and beside the first, none of these models work in UVR, even if downloaded manually (plus VitLarge arch is not supported in UVR at all). Currently, there are various ensembles to choose from on MVSEP.

As for 4/8 models ensemble on MVSEP - they're all only for premium users, as many resources and models are being used to output these results

- 1648 on MVSEP is MDX23C HQ1 model (a.k.a. 8K FFT)

- SYHFT V4 and V5 beta by SYH99999 were never publicly released.
V5 Beta is only on x-minus.pro/uvronline and got deleted from the main models view, but might be still accessible via the following links:

<https://uvronline.app/ai?hp&test> (premium)

<https://uvronline.app/ai?test> (free)

UVR models repository

UVR5 single models' repository backup as separate links (excluding VIP models, which are offline after decrypting):

https://github.com/TRvIvr/model_repo/releases/tag/all_public_uvr_models

All of publicly available MVSEP models (including checkpoints just for further training):

<https://github.com/ZFTurbo/Music-Source-Separation-Training/releases>

(refer to the list of models in this document for descriptions of the best models)

Alternatives models' links list repo:

<https://bascurtiz.x10.mx/models-checkpoint-config-urls.html> (some can be offline)

<https://github.com/SiftedSand/MusicSepGUI/blob/main/models.json>

https://huggingface.co/spaces/TheStinger/UVR5_UI/blob/main/assets/models.json

Some of the older UVR5 GUI models described in this guide can be downloaded via expansion packs:

https://github.com/Anjok07/ultimatevocalremovergui/releases/download/v5.3.0/v5_model_expansion_pack.zip

<https://github.com/Anjok07/ultimatevocalremovergui/releases/download/v5.3.0/models.zip>

<https://github.com/Anjok07/ultimatevocalremovergui/releases/download/v4.0.1/models.zip>

Some of the models used by KaraFan:

https://github.com/Eddycrack864/KaraFan/releases/tag/karafan_models

MDX23C HQ 2

https://github.com/deton24/Colab-for-new-MDX_UVR_models/releases/download/v1.0.0/MDX23C-8KFFT-InstVoc_HQ_2.ckpt

427:

https://drive.google.com/drive/folders/16sEox9Z_rGTngFUtJceQ63O5S9hhjjDk?usp=drive_link (just in case)

Copy it to Ultimate Vocal Remover\models\MDX_Net_Models and rename the model name to: UVR-MDX-NET_Main_427

Jarredou's models mirror

<https://huggingface.co/jarredou> (6+ models)

Some direct links

VOCALS-InstVocHQ

Config:

https://raw.githubusercontent.com/ZFTurbo/Music-Source-Separation-Training/main/configs/config_vocals_mdx23c.yaml

Checkpoint:

https://github.com/ZFTurbo/Music-Source-Separation-Training/releases/download/v1.0.0/model_vocals_mdx23c_sdr_10.17.ckpt

VOCALS-MelBand-Roformer (by KimberleyJSON)

Config:

https://raw.githubusercontent.com/ZFTurbo/Music-Source-Separation-Training/main/configs/KimberleyJensen/config_vocals_mel_band_roformer_kj.yaml

Checkpoint:

<https://huggingface.co/KimberleyJSON/melbandroformer/resolve/main/MelBandRoformer.ckpt>

VOCALS-BS-Roformer_1297 (by viperx)

Config:

https://raw.githubusercontent.com/ZFTurbo/Music-Source-Separation-Training/main/configs/viperx/model_bs_roformer_ep_317_sdr_12.9755.yaml

Checkpoint:

https://github.com/TRvlvr/model_repo/releases/download/all_public_uvr_models/model_bs_roformer_ep_317_sdr_12.9755.ckpt

VOCALS-BS-Roformer_1296 (by viperx)

Config:

https://raw.githubusercontent.com/TRVlvr/application_data/main/mdx_model_data/mdx_c_configs/model_bs_roformer_ep_368_sdr_12.9628.yaml

Checkpoint:

https://github.com/TRVlvr/model_repo/releases/download/all_public_uvr_models/model_bs_roformer_ep_368_sdr_12.9628.ckpt

VOCALS-BS-RoformerLargev1 (by unwa)

Config: https://huggingface.co/jarredou/unwa_bs_roformer/raw/main/config_bsrofoL.yaml

Checkpoint:

https://huggingface.co/jarredou/unwa_bs_roformer/resolve/main/BS-Roformer_LargeV1.ckpt

KARAOKE-MelBand-Roformer (by aufr33 & viperx)

Config:

https://huggingface.co/jarredou/aufr33-viperx-karaoke-melroformer-model/resolve/main/config_mel_band_roformer_karaoke.yaml

Checkpoint:

https://huggingface.co/jarredou/aufr33-viperx-karaoke-melroformer-model/resolve/main/mel_band_roformer_karaoke_aufr33_viperx_sdr_10.1956.ckpt

OTHER-BS-Roformer_1053 (by viperx)

Config:

https://raw.githubusercontent.com/TRVlvr/application_data/main/mdx_model_data/mdx_c_configs/model_bs_roformer_ep_937_sdr_10.5309.yaml

Checkpoint:

https://github.com/TRVlvr/model_repo/releases/download/all_public_uvr_models/model_bs_roformer_ep_937_sdr_10.5309.ckpt

CROWD-REMOVAL-MelBand-Roformer (by aufr33)

Config:

https://github.com/ZFTurbo/Music-Source-Separation-Training/releases/download/v.1.0.4/model_mel_band_roformer_crowd.yaml

Checkpoint:

https://github.com/ZFTurbo/Music-Source-Separation-Training/releases/download/v.1.0.4/mel_band_roformer_crowd_aufr33_viperx_sdr_8.7144.ckpt

VOCALS-ViTLarge23 (by ZFTurbo)

Config:

https://raw.githubusercontent.com/ZFTurbo/Music-Source-Separation-Training/refs/heads/main/configs/config_vocals_segm_models.yaml

Checkpoint:

https://github.com/ZFTurbo/Music-Source-Separation-Training/releases/download/v1.0.0/model_vocals_segm_models_sdr_9.77.ckpt

CINEMATIC-BandIt_Plus (by kwatcharasupat)

Config:

https://github.com/ZFTurbo/Music-Source-Separation-Training/releases/download/v.1.0.3/config_dnr_bandit_bsrnn_multi_mus64.yaml

Checkpoint:

https://github.com/ZFTurbo/Music-Source-Separation-Training/releases/download/v.1.0.3/model_bandit_plus_dnr_sdr_11.47.ckpt

DRUMSEP-MDX23C_DrumSep_6stem (by aufr33 & jarredou)

Config:

https://github.com/jarredou/models/releases/download/aufr33-jarredou_MDX23C_DrumSep_model_v0.1/aufr33-jarredou_DrumSep_model_mdx23c_ep_141_sdr_10.8059.yaml

Checkpoint:

https://github.com/jarredou/models/releases/download/aufr33-jarredou_MDX23C_DrumSep_model_v0.1/aufr33-jarredou_DrumSep_model_mdx23c_ep_141_sdr_10.8059.ckpt

4STEMS-SCNet_MUSDB18 (by starrytong)

Config:

https://github.com/ZFTurbo/Music-Source-Separation-Training/releases/download/v.1.0.6/config_musdb18_scnet.yaml

Checkpoint:

https://github.com/ZFTurbo/Music-Source-Separation-Training/releases/download/v.1.0.6/scnet_checkpoint_musdb18.ckpt

DE-REVERB-MDX23C (by aufr33 & jarredou)

Config:

https://huggingface.co/jarredou/aufr33_jarredou_MDXv3_DeReverb/resolve/main/config_dereverb_mdx23c.yaml

Checkpoint:

https://huggingface.co/jarredou/aufr33_jarredou_MDXv3_DeReverb/resolve/main/dereverb_mdx23c_sdr_6.9096.ckpt

DENOISE-MelBand-Roformer-1 (by aufr33)

Config:

https://huggingface.co/jarredou/aufr33_MelBand_Denoise/resolve/main/model_mel_band_roformer_denoise.yaml

Checkpoint:

https://huggingface.co/jarredou/aufr33_MelBand_Denoise/resolve/main/denoise_mel_band_roformer_aufr33_sdr_27.9959.ckpt

DENOISE-MelBand-Roformer-2 (by aufr33)

Config:

https://huggingface.co/jarredou/aufr33_MelBand_Denoise/resolve/main/model_mel_band_roformer_denoise.yaml

Checkpoint:

https://huggingface.co/jarredou/aufr33_MelBand_Denoise/resolve/main/denoise_mel_band_roformer_aufr33_aggr_sdr_27.9768.ckpt

For a more recent list see [this](#) Colab and cells containing all the links there too.

Why not to use more than 4-5 models for ensemble in UVR - [click](#)

Other models

- [GSEP AI](#) - new model called “Vocal Remover”, and old instrumental, vocal, 4-6 stem model (it applies additional denoiser for 4/6 stems) - piano and guitar (free). As for 2 stems, it gives very good instrumentals for songs with very loud and harsh vocals and a bit lo-fi hip-hop beats, as it can remove vocals very aggressively. Sometimes even more than HQ_3. The new model might be good at removing SFX (instrumental stem is the old model).

In specific cases (can have more vocal residues in instrumentals vs HQ_3 at times - less in jarredou's Colab):

- original [MDX23](#) by ZFTurbo (only this OG version of MDX23 still works in the offline app, min. 8GB Nvidia card required [6GB with specific parameters]) - sounds very clean though, and not that muddy like inst MDX models, in this means, comparable with even VR arch or better (because of much less vocal residues).

- [Demucs_ft](#) model (both 3 stems to mix in e.g. Audacity for instrumental) / sometimes 6s model gives better results, or in very specific cases when vocals are easy to filter out - even the old 4 stem mdx_extra model (but SDR wise full band MDX 292 is already better than even ft model). The 6s model is worth checking with shifts 20.

Might be still usable in some specific cases, despite the fact that MDX23 uses demucs_ft and other models combined.

- [VR models](#) settings + VR-only [ensemble settings](#) (generally deprecated, but sometimes more clarity vs MDX v1, though frequently more vocal residues. Some people still uses it e.g. for some rock, when it can still give better results than other models, and also for fun dubs, but for it if you have two language tracks of the same movie, you can test out [Similarity Extractor](#) instead, but Audacity center extraction works better than that linked Colab)

- Alternatively, you can consider using narrowband Kim other ft model with fullband model settings parameters in [this](#) or the new HV Colab instead. Useful in some specific parts of songs like chorus, where there are still no persistent vocal residues using this method

(clearer results than even Max-Spec) or e.g. MDX23 still doesn't give you enough clarity in such places to maybe merge fragments manually of results from different models.

Paid

- [Audioshake](#) (non-copyrighted music only, can be more aggressive than above and pickup some lo-fi vocals where other fails [a bit in manner of HQ models])

How to bypass the non-copyright music restriction ([1](#), [2](#)).

"They also reserve themselves the right to keep your money and not let you download the song you split if they discover that you are using a commercially released song and that you don't have the rights to it." but generally we didn't have such a case with slowed down songs (otherwise they might not pass anyway)

4 stems might be better at times than Demucs ft model.

- [Dango.AI](#) (a.k.a. tuanziai.com) free 30 seconds samples; can be the most aggressive for instrumentals vs, e.g. inst 3, tested on Childish Gambino - Algorithm). Since then, models/arch were updated and instrumentals in 9.0 seem to be **the cleanest** or the closest to original instrumentals for 12.08.23 at least in some cases (despite low SDR).

> If you care only about specific snippet in a song, then since 30 second samples to separate are taken randomly from a whole song, to have specific fragment separated, you can copy the same fragment over and over to make a full-length track of it, and it will eventually pick up a whole snippet for separation.

X Uploading snippet shorter than or exactly 30 seconds will not result in the whole fragment being processed from the beginning to the ending.

> Sometimes using other devices or virtual machine in addition to incognito/VPN/new email might even be necessary to reset free credits. It's pretty persistent.

<https://tuanziai.com/encouragement>

Here you might get 30 free points (for 2 samples) and 60 paid points (for 1 full songs) "easily".

>>>

Everything else for 2 or 4 stems than above is worse for separation tasks:

Lalal, RipX (although now it uses some UVR models (?), Demix, RX Editor 8-11, Spleeter and its online derivatives.

Debleeding/cleaning vocals/instrumentals/inverts

- Roformers: Mel 2024.10 on MVSEP a.k.a. Bas Curtiz FT (for vocals)
- Or earlier Kim Mel-Roformer model (for vocals; if other model was prev. used)
- Unwa inst v1 (to clean-up vocals from Mel 2024.10 model)
- unwa's inst v1/e/2 (for OG instrumentals with bleeding [better than Dango for it])
- unwa big beta 5 ("my go-to clean-up artifacts model after phase inverting master + official instrumental")
- Unwa Revive 3e

- voc_fv6 (for vocal inverts - ezequielcasas)
- DEBLEED-MelBand-Roformer (by unwa/97chris) [model](#) | [yaml](#) | [Colab](#)
(it can work with e.g. inst v1 and its noise, or with v1e, or even MVSEP Karaoke
BS-Roformer instrumental stem for “very clean and full” result. Also, sometimes the debleed
model can remove some bleed also after using phase fixer)
- Mel avuew’s v2 [de-reverb](#) or unwa’s BS Large, MDX-Net HQ_5
 (“great for cleaning acapellas from bits of instrumentals”)
- syftbeta 5 on x-minus.pro (probably still available with [this](#) link for premium, and for [free](#))
- Ensemble of BSRoformer-Viperx1297, Unwa BSRoformer-LargeV1, unwa_ft2_bleedless,
mel_band_roformer_vocals_becruily, Gabox voc_fv4 on Average/Average
(good for cleaning inverts - AG89)
- Ensemble of big beta6x, roformer revive2, unwa_ft2_bleedless
(for cleaning instrumental inverts - AG89)
- Or just use models with the highest bleedless metric ([instrumental](#) | [vocals](#))
- RX10 De-bleed feature for instrumentals ([video](#))
(older methods)
 - Gabox Mel [denoise/debleed](#) model | [yaml](#) | [Colab](#) | SESA [Colab v3](#) - for noise from fullness
models (tested on v5n) - it can't remove the vocal residues
 - Mel [denoise](#) - that model “removed some of the faint vocals that even the bleed suppressor
didn't manage to filter out” before”. Try out denoising on a mixture first, then use the model.
 - (*for saxophone bleeding*) ~1. Take the original song in FLAC or WAV 2. Use MVSEP
Saxophone 3. Take the other stem from it - there should be everything else and most of the
sax should be gone (for me, there was a small part left) 4. Use Unwa Big Beta 5 on it (so
Other-> uvronline/xminus/Colab Unwa Big Beta 5) - then vocals should be very clean no sax
bleeding” cali_tay98
 - (“In case there's any wind instruments that could potentially bleed into the vocals”)
MVSep Wind
 - (*when “models don't pick up the noise*), gently bring back a bit of the original
music/instrumental on the inverted track and use AI again.
By gently, I mean no more than 6 dB“ - becruily
 - (*If your result have “vocal chops”*) left in the instrumental separation and no models could
remove them completely, then it's likely MDX HQ_5 or VitLarge23 v2 will fix it” dca
 - Acon Digital DeBleed:Drums “it's just an advanced gate. It doesn't remove bleed when it's
overlapping the wanted audio (we can still hear hihat/cymbals on snare with the plugin
enabled in their [demo](#))” - jaredou.
 - [Audio-Bleeding-Removal](#) - by its-rajesh
 - Try out some L/R inverting, try out to separate multiple times to get rid of some vocal
pop-ins like this (fix for ~“ah ha hah ah” vocal residues)

Older models

- [Ripple](#) (defunct) “AWESOME to use after inverting songs with the official instrumental”
Instrumentals can be also further cleaned with Ripple, and then with [Bandlab Splitter](#)
(Roformer models may potentially replace Ripple models in that matter now)

- [Top](#) ensemble in UVR5 (starting from point 0d)
- [GSEP](#) - very minor difference between both for cleaning vocals (maybe GSEP is better by a pinch).
You can try separating e.g. vocal result double using different settings (e.g. voc_ft>kim vocal 2)
- MVSEP 11.50 Ensemble (the least amount of bleeding in inst. separations at least)
- MDX23 jarredou's fork Colab (maybe [this](#) version at first)
- use voc_ft model on the result you got (so separate twice if you already used that model)

Cleaning inverts means - cleaning up residues - e.g. left by the instrumental after an imperfect phase cancellation, e.g. when audio is lossy, or maybe even not from the same mixing session -

Aligning

"[Utagoe](#) bruteforces alignment every few ms or so to make sure it's aligned in the case that you're trying to get the instrumental of a song that was on [e.g.] vinyl."
[The previous] UVR's align tool is handy just for digital recordings... [so those] which don't suffer from that [issue] at all."
Utagoe will not fix fluctuating speed issues, only the constant ones.

Anjok already "cracked" how that specific Utagoe feature works, and introduced it to UVR.
"Updated "Align Tool" [is] to align inputs with timing variations, like Utagoe."

"Some users had good results with Auto Align Post 2 plugin to resync tracks before inverting them."

For problematic inverts, you can also try out azimuth correction in e.g. iZotope RX.

Declicking vocals

- BS-Rofomer ("Can also fix hard clicks in vocals. It is even better than RX in this, but still there is a tiny wave fade residue in some cases")
- *Kim vocal* first and then separate with instrumental model (e.g. HQ_3 or 4). You might want to perform additional separation steps to clean up the vocal from instrumental residues first, and invert it manually to get cleaner instrumental to separate with instrumental model to get rid of vocal residues

Removing metronome

- Use a good instrumental model so you will be left with metronome + vocals in one stem, then use a drums model - "Then the drum trick worked better but still not very good, a regular extraction worked better this time though!" - brianghost

Removing bleeding of hi-hats in vocals

- Kim Mel-Roformer model

- Use MelBand RoFormer v2 on MVSEP (e.g. after using MDX23C Inst HQ)

Bleeding in other stems

- RipX Stem cleanup feature (possibly)
- SpectraLayers 10 (eliminates lots of bleeding and noise from MDX23 Colab ensembles)
"You debleed the layer to debleed from using the debleed source. Results vary. Usually it's better to debleed using Unmix and then moving the bleed to where it belongs" Sam Hocking
[Video](#)

Bleeding of claps in vocals

- Reverse polarity and/or remove DC offset of the input file
- [KaraFan](#) (for general drum artefacts, but it doesn't work well for inverts, try out modded preset 5 [here](#))
- Remove drums with e.g. demucs_ft first, then separate the drumless mixture from inversion
- [Settings](#) for VR Colab
- Kim Vocal 2 (but it has a cutoff and creates a lot of noise in the output)
- Denoise model with 0.1-0.2 aggressiveness
- [Sam Hocking method](#)

Bleeding of guitars/winds/synths in vocals

- BVE (Karaoke) models

Overlapped/misrecognized stems

- Spectralayer's 9/+ [Cast & Mold](#)

Low-end rumble

- Spectral Editing:
 - a) RX Editor's brush ([video](#) by Bas)
 - b) Audacity ([image](#)) - "you can, just barely"

Potential alternatives for spectral painting:

Free: [ISSE](#), [Ampter](#), [Filter-Artist](#), [AudioPaint](#)

Paid: RipX, SpectraLayers, Melodyne, prob. Revoice Pro 5

Bleeding of instruments in vocals

- Denoise model with 0.1-0.2 aggressiveness

Cleaning the white noise/sizzle inside the vocals

(from e.g. Roformer models)

- MDX23C model (e.g. the latest on MVSEP or HQ in UVR)

[Debleeding guide by Bas Curtis](#) (other methods, e.g. Audacity)

[Denoising](#) and [dereverberation/apps](#) later below.

See also “[Vinyl noise/white noise](#)” from the end of the list.

How to check whether a model in UVR5 GUI is vocal or instrumental?

- Read carefully the models list above - they're categorized
- If you want to experiment with other models:

The moment you see "Instrumental" on top (and "Vocal" below) in the list where GPU conversion is mentioned, you know it's an instrumental model.

When it flips the sequence, so Vocal on top, you know it's a vocal model.

Same happens for MDX and VR archs.

- “Be aware that MDX23C/MDXv3 models can be multisource - it depends on the training, so it can be only vocals, or only instrumental, or vocals+instrumental, or vocals+drums+bass+other (like baseline models are), or whatever else.
- You can know it looking at the config file of the model, for example InstVocHQ, https://github.com/Anjok07/ultimatevocalremovergui/blob/master/models/MDX_Net_Models/model_data/mdx_c_configs/model_2_stem_full_band_8k.yaml

Seeing by the instruments line above, D1581 and InstVocHQ models are instrumental+vocal.

Config for the rest of the models:

[\(decoded hashes\)](https://github.com/Anjok07/ultimatevocalremovergui/blob/master/models/MDX_Net_Models/model_data/model_data.json)

Keeping only **backing vocals** in a song (lead vocal extractor):

>Karaoke

You might want to use a good [vocal](#) model as a preprocessor to use with the models below (if MVSEP/x-minus don't do it already), but sometimes it can degrade the quality, so experiment both ways.

Optionally you may also [de-reverb](#) vocals with a good model/plugin first before proceeding, but only if the specific model/method doesn't filter out a lot of BGVs in your vocals.

- Anvuew's Karaoke BS-Roformer [model](#) | [metrics](#) | incompatible with UVR RTX 5000 patch | [MSST](#) | MVSEP | [Colab](#)

(Lead vocals/backing with instrumental)

"extracts lead vocals a bit better than karaoke becruily frazer, and in some parts, the lead vocals from karaoke anvuew still sound brighter compared to karaoke becruily frazer, which sounds a bit more compressed. oh, and for some reason, the becruily frazer model doesn't detect vocals with radio effects, while anvuew's model handles them just fine" - neoculture "lead vocals leak into instrumental (...) Mel Becruily and Frazer's BS don't have this problem" In that case, maybe "isolate the acapella first in almost all cases of using a karaoke model" or use the model below instead.

- BS-Roformer Karaoke [model](#) by becruily & frazer | [metrics](#) | MVSEP | uvrone

(Lead vocals/backing with instrumental)

Make sure you don't have the option "Vocals only" checked in UVR.

8GB VRAM users of AMD and Intel ARC GPUs need to use 160000 chunk_size, or the separation will be very slow.

"After dozens of tests I can tell this (...) is the best (better harmony detection, better differentiation between LVs and BVs, sounds fuller, less background Roformer bleed, better uncommon panning handling etc) (...) I noticed "lead vocal panning" works really well" - dca "It also can detect the double vocals" - black_as_night

It works the best for some previously difficult songs. Aufr33 and viperx model seems more consistent, but the new BS is still the best in overall - Musicalman

"My OG Mel also catches some of the FX/drums, I guess quite a difficult one due to how it's mixed" - Becruily

"It does do better on mono than previous, sometimes confuses which voice should be the lead, but all models do that on mono in the exact use-case I normally test" - Dry Paint Dealer Undr

"[In] no way inferior to the ViperX (Play da Segunda)" - fabio5284

"The new karaoke model doesn't actually differentiate between lvs & bvs and there's some lead vocal bleeding in the instrumental stem" - scdxtheresolution

VS the newer BS-Roformer MVSEP team model: "sound isn't as clear, but it does an infinitely better job at telling lead/bgv apart"

Becruily:

"I want to remind something regarding my (and the frazer) models

they're made to separate true lead vocals, meaning either all of the main singer's vocals, or if it's multiple singers - theirs too

this means if the main singer has stuff like adlibs on top of the main vocals, these are considered lead vocals too - they go together

if there are multiple singers singing on top of each other, including harmonise each other, and if there are additional background vocals behind those - all the singers will be separated as one main lead vocal, leaving only the true background vocals"

"Noticed it a couple days ago. I've had fantastic results with it so far. Much MUCH better at holding the 'S' & 'T' sounds than the Rofo oke (for backing vox). Generally seems to provide fuller results... but also the typical 'ghost' residue from the main vox can end up in the backing vox sometimes, but it's usually not enough to be an issue. I won't go so far as to say that it's replacing the other backing vox models for me entirely... but it feels like the best of both worlds that Rofo and UVR2 provide." - CC Karaoke

Tips for the model

- "I had success by setting the BS Rofo Karaoke model to 100% Right, and then taking the 'Other' result and reprocessing it at 100% Left to get the backing vocals cleanly out.

Curious note on something I've never tried or had to do before but it has worked wonderfully; I'm isolating the backing vocals on Radiohead's Let Down." - CC Karaoke

- If you use e.g. vocfv7beta1 as preprocessor for the model, you may get some quieter backing vocals better - Rainboom Dash

- MVSEP's BS Roformer by MVsep Team (SDR: 10.41)

under option "MVSep MelBand Karaoke (lead/back vocals)", [metrics](#). Might be a fine-tune. Use the option extract vocals first.

("In contrast with other Karaoke models, it returns 3 stems: "lead", "back" and "instrumental".)

"If I had to compare it to any of the models, it is similar to the frazer and becruily models. Sometimes it does not detect the lead vocals especially if there's some heavy hard panning, but when it does, there is almost no bleed, and it works very well with heavy harmonies in mono from what I tested." - smilewasfound

"becruly & frazer is better a little when the main voice is stereo" - daylightgay

"On tracks I tested, harmony preservation was better in becruily & frazer (...) the new model isn't worse, I ended up finding examples like Chan Chan by Buena Vista Social Club or The Way I Are by Timbaland where it is better than the previous kar model. The thing is, with the Kar models, it's just track per track. Difficult to find a model for batch processing as it's really different from one track to another" - dca100fb8

As for MVsep Team: "It's the only model that combines the lead vocal doubles with the lead vocals stem. It's far more useful for dissecting harmonies on songs with vocal doubles like Backstreet Boys" - heuheu

"I also found the new model to not keep some BGVs, mainly mono/low octave ones, despite higher SDR" - becruily

"I think I've found a solution for people who don't like the new model.

If you put an audio file through the karaoke model and then put the lead vocal result through that, it usually picks up doubles.

Which you can then put in your BGV stem if you'd like" - dynamic64

- Ensemble of Mel v1e + BS Karaoke MVSep Team with extract vocals first option, Max Spec (using BS 2025.07 as reference and 200/200 for the values)

"It's very aggressive values cuz v1e is noisy, and it works quite well", the best ensemble for now (dca100fb8)

- Ensemble of BS Roformer Karaoke by anvew + BS Resurrection Inst (aka "unwa high instrum fullness" on mvsep) + phase fix (using BS 2025.07 as reference), older, more crossbleeding - (dca100fb8)

- "Ensemble of 3 models "MVSep + Gabox + frazer/becruily" gives 10.6 SDR on the leaderboard. I didn't upload it yet, but I had local testing." - ZFTurbo

- Ensemble of ([metrics](#)):

BS-Roformer Karaoke Frazer & Becruily + BS-Roformer Karaoke Anvew (avg/avg)

- MVSEP fusion model of Gabox Karaoke and Aufr33/viperx
(tend to confuse BV/LV more than single models)

- Gonzaluigi Karaoke fusion models ([standard](#) | [aggressive](#) | [yaml](#)) -
also confuses BV/LV more

- MVSep MelBand Karaoke (lead/back vocals) SCNet XL IHF by becruily (SDR: 9.53)
Worse SDR than the top performing Roformers, but works best in busy mix scenarios, and when Mel-Roformer models fail, generally bleedier arch. To fix the bleed in the back-instrum stem, use "Extract vocals first", but "I noticed a pattern that if you hear the lead vocals in the back-instrum track already (SCNet bleed), don't try to use Extract vocals first because there will be even more lead vocal bleed" - dca (iirc it uses the biggest SDR BS-Roformer vocal model as preprocessor).

"Separates lead vocals better than Mel-Roformer karaoke becruily. It's not perfectly clean, sometimes a bit of the backing vocals slips through, but for now, scnet karaoke model still the most reliable for lead vocals separation (imo)" - neoculture

- (x-minus/uvronline) Lead and Backing vocal separator (in "Extract Backing
vocals>Mel-Roformer Lead/Back)

~~It uses big beta 5e model as preprocessor for becruily Mel Karaoke model~~

"In fact, the big beta 5e model is run after becruily Mel Karaoke" Aufr33 (so you don't need the additional step to use this separator), plus it also allows controlling option for lead vocal panning like for BVE v2 (It's to "tell" the AI where the main vocals are located (how they are mixed)). "Doesn't even need Lead vocal panning a lot of the time, [the] ability to recognize what is LV and what is BV [is] impressive" - dca).

"The new separator is available in the free version, however, due to its resource intensity, only the first minute of the song will be processed." if you don't have a premium.

The difference from using single becruily Kar model is that, here, “you get the third track, backing vocals.”.

- Becruily: “Probably too resource-intensive, but you could try adding demudders to each step

- 1) Karaoke model + demudding
- 2) Separate vocals of BGV + demudidng

But not sure how much noise this will bring

(Or even a 50:50 ensemble with BVE OG)”

- Mel-Roformer Karaoke by becruily [model file](#) | [Colab](#) | MVSEP | x-minus
(back/main vox/instrumental - 3 stems)

Use voc_fv4 vocal model before running it (less bleeding than voc_fv6) or:
“first extract the vocals with a fullness model [it was Mel becruily vocal back then] and combine the results with a fullness instrumental model.” - becruily

“It's a dual model, trained for both vocals and instrumental. It sounds fuller + understands better what is lead and background vocal,
and to me, it is better than any other karaoke model.

Important note: This is not a duet or male/female model. If 2 singers are singing simultaneously + background vocals, it will count both singers as lead vocals. The model strictly keeps only actual background vocals. The same goes for “adlibs” such as high notes or other overlapping lead vocals.

The model is not foolproof. Some songs might not sound that much improved compared to others. It's very hard to find a dataset for this kind of task.

Compared to Aufr33’s Melband model below, it can achieve e.g. cleaner pronunciation in some songs ([examples](#)).

“Better than Mel Kar, UVR BVE v2, lalal.ai, Dango...”

“For sure better than [the] older Karaoke (aufr’s model) for harmonies. Though I can say Dango can remain useful in certain situations” - dca

On x-minus there’s lead vocal panning setting added for Mel-RoFormer Kar by becruily model. It’s to “to “tell” the AI where the main vocals are located (how they are mixed).”. “doesn’t even need Lead vocal panning a lot of the time, [the] ability to recognize what is LV and what is BV [is] impressive” - dca

“Sometimes struggles when the backing vocals are the same notes as the lead vocals” - isling. Seems like xminus panning can’t solve such issues either.

"Had a similar issue as you however with the Chase Atlantic vocal, MDX Kar V2 with stereo 80% then Chain or Max mag extracts the leftovers works very very well not perfect (at least works for most CA song) but It's enough for me to do an edit" - cali_tay98"

It seems demudder shouldn't be used when Lead vocal panning is set to something different than center, I noticed it brings back the lead vocals in the inst w/BVs as it was before changing LV panning" - dca

- Mel-Roformer Karaoke (by aufr33 & viperx) on [x-minus.pro](#) / [uvronline.app](#) / [mvsep model file](#) (UVR [instruction](#)) - online version above might work better, not sure about preprocessor model, maybe even old voc_ft, but not necessarily.
This Mel may extract more than BVE V2, if "extract directly from mixture" on MVSEP doesn't detect the BVs (x-minus behavior for this model), the chances are "extract from vocals part" on MVSEP (which uses BS-Roformer 2024.08 for it) will detect more BVs (although with possible cross bleed between lead/back in inst+bv)
- If you ensemble the model above with Unwa v1e (Max) it removes all the muddiness of Mel Kar (dca100fb8)

"You can do it via:

Choose Process Method --> Audio Tools --> Manual Ensemble --> Max Spec

Q: How do you select both inputs?

A: Via Select Input, but if you want to batch process Manual Ensemble it's not possible yet"
(dca100fb8)

Ensemble algorithms like ~“min_spec in the direct ensemble are only available if the selected type of stems in the yaml corresponds with the target in UVR.

Example:

If the target in the yaml is:

- vocals
- other

then you can't use it in the Vocal/Instrumental selection because it's not written that way in the yaml." (mesk)

- Gabox Mel [KaraokeGabox](#) model (uses Aufr's [config](#)) | [Colab](#) - finetune of becruiy model
"The lead vocals are good and clean!
While the backing tracks are lossy for this model, [it still] provide[s] great convenient for those who need LdV"
"The model doesn't keep the backing vocals below the main vocals, sometimes the backing vocals will be lost even though there are backing vocals there."
- BVE v2 model on x-minus.pro/uvronline.app for premium users | [model](#) (uses "4band_v4_ms_fullband" stock [config](#)) by Aufr33

Place the model file in Ultimate Vocal Remover\models\VR_Models and config file in lib_v5\vr_network\modelparams (if doesn't exist already). Then pick "4band_v4_ms_fullband.json" and BV when asked to recognize the model (it has the same checksum as in modelparams folder if it's there already). Seems like it works with "VR 5.1 model" checked (and probably without it too).

"Note that this model should be used with a rebalanced mix.

The recommended music level is no more than 25% or -12 dB.

If you use this model in your project, please credit me."

(it's v1 version is also added in UVR's "Download More Models", but also without stereo width feature which can fix some issues when BVs are confused with other vocals).

On x-minus "When you select stereo, it applies a stereo narrower before AI processing."

It used to be one of the best models for this purpose. On x-minus at certain point it used voc_ft for all vocals as a preprocessor already (not sure if it got changed).

"BVE sounds good for now but being an (u)vr model the vocals are soft (it doesn't extract hard sounds like K, T, S etc. very well)"

"Seems to begin a phrase with a bit of confusion between lead and backing, but then kicks in with better separation later in the phrase."

"If something struggles to separate on bve v2 I change the lv panning option to either 50 or 80% [stereo or center], and it separates it amazingly.

It even allows me to separate backing backing vocals from backing vocals"

- For UVR BVE v2 LV bleed in Song without LV - download the BV track and add it to inst v1e model result - no vocal bleed/residues (introC).

"I tried it ensembled with Gabox's model [kar_gabox]" and they are amazing together. Yes you have to make the primary stem of aufr33's model "Instrumental" if you're ensembling" - AG89

- Newer Gabox experimental [Karaoke model](#) (June 2025). It's one stem target so keep extract_instrumental enabled for the rest stem.

"really hard to tell the difference between this and becruily's karaoke model" minus the latter has more target stems.

- Chain ensemble mode for B.V. models (available on [x-minus.pro](#) for premium users, added in UVR beta 5.5/9.15 beta [patch](#) already):

It is possible to recreate this approach using non-BVE v2 models in UVR by processing the output of one Karaoke model by another (possibly VR model as the latter) with Settings>Additional Settings>Vocal Split Mode option (so it separates using the main model for all vocals, then it uses the result as input for the next model).

So you might experiment with using voc_ft or Kim Vocal 2 or 1296 as the main vocal model in the main UVR window, and in Vocal Split Mode use HP5 or HP6 or BVE model, so you won't have to make the process in 2 steps manually, so separating the result with another model once the first separation is done. Although Vocal Split Mode was designed mainly for

BVE models, so in case of any problems with HP5/6 or Karaoke, you can test out also Settings>Choose Advanced Menu>[model arch]>Secondary model instead.

Don't forget reading [vocals](#) to find the best separation method for your song to use it for separation with Karaoke/BVE models.

Recommended ensemble settings for Karaoke in UVR 5 GUI (instrumentals with backing vocals):

- 5_HP-Karaoke-UVR, 6_HP-Karaoke-UVR, UVR-MDX-NET Karaoke 2 (Max Spec) (in e.g. "min/max" the latter is for instrumental)

- Alternatively, use Manual Ensemble with UVR with Max Spec using x-minus' [UVR BVE v2](#) result and the UVR ensemble result from the above.

Or single model:

- HP_KAROKEE-MSB2-3BAND-3090 (a.k.a. VR's 6HP-Karaoke-UVR)
- UVR BV v2 on x-minus (and download "Song without L.V.". Better solution, newer, different model)
- 5HP can be sometimes better than 6HP

([UVR5 GUI](#) / [x-minus.pro](#) / [Colab](#)) - you might want to use Kim Vocal 2 or voc_ft or 1296 or MDX23C first for better results.

- UVR-BVE-4B_SN-44100-1

Q: What are the differences between Mel-Roformer Karaoke and the last model?

A: If the vocals don't contain harmonies, this model (Mel) is better. In other cases, it is better to use the MDX+UVR Chain ensemble for now.

- Gabox denoise/debleed Mel-Roformer | [model](#) | [yaml](#) | [Colab](#)
"better results than kar v2"
- Gabox kar v2 | [model](#) | [yaml](#)
- De-echo VR model in UVR5 GUI set to maximum aggression
- [MedleyVox](#) with our trained model (more coherent results than current BV models)

Or ensemble in UVR:

"The karaoke ensemble works best with isolated vocals rather than the full track itself"

- VR Arc: 6HP-Karaoke-UVR
- MDX-Net: UVR-MDX-NET Karaoke 2
- Demucs: v4 | htdemucs_ft

Or:

- VR Arc: 5HP-Karaoke-UVR
- VR Arc: 6HP-Karaoke-UVR
- MDX-Net: UVR-MDX-NET Karaoke 2
(Max Spec, aggression 0, high-end process)

Or:

- VR arc: 5_HP-Karaoke
- MDX-Net: UVR-MDX Karaoke 1
- MDX-Net: UVR-MDX Karaoke 2
(you might want to turn off high-end process and post process)

Or:

- VR Arc: 5HP-Karaoke-UVR
- VR Arc: 6HP-Karaoke-UVR
- MDX-Net: UVR-MDX-NET Karaoke 1
- MDX-Net: UVR-MDX-NET Karaoke 2
(Min/Min Spec, Window Size 512, Aggression 100, TTA On)

If your main vocals are confused with backing vocals, use X-Minus and set "Lead vocal placement" to center (not in UVR5 at the moment).

Or [Mateus Contini's](#) method.

[How to extract backing vocals X-Minus Guide](#) (can be executed in UVR5 as well)

Vinctekan Q&A

Q: Which BVE aggression settings (for VR model, e.g. uvr-bve-4b-sn-44100-1) is good for backing removal?

A: "I recommend starting from exactly from 0 and working from there to either - or +.

0 is the baseline for BVE that are almost perfectly center.

If it's off to the left or right a little bit, I would start from 50"

Q: How do I tell what side BVs are panned or if they are Stereo 50 % or 80 % without extracting them?

A: "It's more about listening to the track. The way I used to it is to invert the left channel with the right channel. In most cases this should only leave the reverb of the vocals in place, but if there is backing vocals that is panned either left or right, then it should be a bit louder than the reverb. Audacity's [Vocal Reduction and Isolation>Analyze] feature usually can give a rough estimates as to how related the two channels are, but that does not tell where the backing vocal actually is. I would only recommend doing the above with a vocal output, though."

Q: Does anyone know how to tell what side BV's (backing Vocals) are panned similar to [this](#)? Like, is there a way to tell using RipX? Or another tool. In my case I think mine might be Stereo 20 30 percent or lower

A: "Your ears [probably the least effective]

If you have Audacity, select your entire track, and select [Vocal reduction and Isolation] and select the [Analyze] but it won't tell you which direction the panning is in.

Or use it to isolate the sides, and just take a look at the output levels of each channel.

Spectralayers's [[Unmix>Multichannel Content](#)] tab can measure the output of frequencies in the spectrogram and can tell you when certain elements are not equal in loudness, which you can restore."

- Dango.ai has also a good BVE model (expensive) - at least sometimes it gives better results than uvr bve v2 to get songs without lead vocals. "meant for separating melody from harmony, not separating singer from singer, so you'll hear both [singers in one] stem" if present

Later a new Advanced repair tool was added to "fix any backing-vocals errors"

- AudiosourceRE Demix Pro has BVE/lead vocals model

- lalal.ai has a new decent lead and backing vocals model

If bve or mel karaoke model CAN do it, then they'll do it better, but if they CAN'T do it, then lalal will do it better. "I have seen lalal work better on mono audio than bve model." ~Isling

dca100fb8: "For Instrumentals with Backing Vocals:

If Mel Kar doesn't work, it's likely Dango Backing Vocal Keeper will not too, although it's not always the case, and still worth trying out - separating left and right channel with Dango Backing Vocal Keeper once fixed the issue.

For back vocals/lead vocals model

If neither Mel Kar or UVR BVE v2 work, it's likely lalal.ai Lead & Back Vocal Splitter will work instead. Its deep Extraction seems to provide better results than Clear Cut"

- Advanced chain processing chart ([image](#))

It's a method utilizing old models, and e.g. Kim Vocals 2 can be potentially replaced by unwa's BS/Mel-Roformer models in [beta UVR](#) (or other good method for [vocals](#)) or ensembles mentioned in this document. Check the best current methods for vocals in one stem to find what works the best for your song to get all vocals before splitting to other stems using this diagram.

htdemucs v4 above can be replaced by htdemucs_ft, as it's the fine-tuned version of the model (or [MDX23 Colab](#)). Even better, you can use some of the methods for [4 stems](#) in this GDoc (like drums on x-minus).

De-echo and reverb models can be potentially replaced by some better paid plugins like: DeVerberate by Acon Digital, Accentize DeRoom Pro (more in the [de-reverb](#) section).

UVR Denoise can be potentially replaced by less aggressive Aufr33 model on x-minus.pro (used when aggressiveness is set to minimum), and there's also newer Mel-Roformer (read [de-reverb](#) section).

As for [Karaoke](#) models, there's e.g. a Mel-Roformer model on x-minus.pro for premium users or MVSEP/jarredeou inference [Colab](#).

"If the vocals don't contain harmonies, this model (Mel) is better. In other cases, it is better to use the MDX+UVR Chain ensemble for now.". It is possible to recreate to some extent this approach while not using BVE v2 models, by processing the output of main vocal model by one of Karaoke/BVE models in UVR (possibly VR model as the latter) using Settings>Additional Settings>Vocal Splitter Options, so it separates using one model, then it uses the result as input for the next model (see the Karaoke section). MedleyVox (not available in UVR) will be useful in the end in cases when everything else fails after you obtain all vocals in one stem, as it's very narrowband. But you can use AudioSR on it afterwards.

>Keeping only **lead** vocals in a song

Sometimes the same model might work for lead, sometimes for back vocals depending on a song

- (x-minus/uvronline) Lead and Backing vocal separator (in "Extract Backing vocals>Mel-Roformer Lead/Back")

~~It uses big beta 5e model as preprocessor for becruily Mel Karaoke model~~

"In fact, the big beta 5e model is run after becruily Mel Karaoke" Aufr33 (so you don't need the additional step to use this separator), plus it also allows controlling option for lead vocal panning like for BVE v2 (It's to "tell" the AI where the main vocals are located (how they are mixed).". "Doesn't even need Lead vocal panning a lot of the time, [the] ability to recognize what is LV and what is BV [is] impressive" - dca).

"The new separator is available in the free version, however, due to its resource intensity, only the first minute of the song will be processed." if you don't have a premium.

The difference from using single becruily Kar model is that, here, "you get the third track, backing vocals.".

- Mel-Roformer Karaoke by becruily [model file](#) | [Colab](#) | MVSEP | x-minus

"It's a dual model, trained for both vocals and instrumental. It sounds fuller + understands better what is lead and background vocal

- Gabox Mel [KaraokeGabox](#) model (uses Aufr's [config](#)) | [Colab](#)

"The lead vocals are good and clean!

While the backing tracks are lossy for this model, [it still] provide[s] great convenient for those who need LdV"

"The model doesn't keep the backing vocals below the main vocals, sometimes the backing vocals will be lost even though there are backing vocals there."

- Newer Gabox experimental [Karaoke model](#) (June 2025). It's one stem target so keep extract_instrumental enabled for the rest stem.

"really hard to tell the difference between this and becruily's karaoke model" minus the latter has more target stems.

- BVE v2 model on x-minus.pro/uvronline.app for premium users | [model](#)

("4band_v4_ms_fullband" stock config) by Aufr33

Place the model file in Ultimate Vocal Remover\models\VR_Models and [config](#) file in lib_v5\vr_network\modelparams. Then pick "4band_v4_ms_fullband.json" and BV when asked to recognize the model (it has the same checksum as in modelparams folder if it's there already). Also, I think it's not a VR 5.1 model.

"Note that this model should be used with a rebalanced mix.

The recommended music level is no more than 25% or -12 dB.

If you use this model in your project, please credit me."

(it's v1 version is also added in UVR's "Download More Models", but also without the stereo width feature which can fix some issues when BVs are confused with other vocals).

On x-minus "When you select stereo, it applies a stereo narrower before AI processing." (sometimes it might work for lead, sometimes for back vocals)

It used to be one of the best models so far. On x-minus it used voc_ft for all vocals as a preprocessor already.

"Seems to begin a phrase with a bit of confusion between lead and backing, but then kicks in with better separation later in the phrase."

"If something struggles to separate on bve v2 I change the lv panning option to either 50 or 80% [stereo or center], and it separates it amazingly.

It even allows me to separate backing backing vocals from backing vocals"

"BVE sounds good for now but being an (U)VR model the vocals are soft (it doesn't extract hard sounds like K, T, S etc. very well)"

- For UVR BVE v2 LV bleed in Song without LV - download the BV track and add it to inst v1e model result - no vocal bleed/residues (introC).

- Mel-Roformer Karaoke (by aufr33 & viperx) on [x-minus.pro](#) / [uvronline.app](#) / [mvsep model file](#) (UVR [instruction](#)) - online version above might work better

(may extract more than BVE V2), if "extract directly from mixture" on MVSEP doesn't detect the BVs (x-minus behavior for this model), the chances are "extract from vocals part" on MVSEP (which uses BS-Roformer 2024.08 for it) will detect more BVs (although with possible cross bleed between lead/back in inst+bv)

- Gabox Mel [KaraokeGabox](#) model (uses Aufr's [config](#)) | [Colab](#)

"The lead vocals are good and clean!

While the backing tracks are lossy for this model, [it still] provide[s] great convenient for those who need LdV"

"The model doesn't keep the backing vocals below the main vocals, sometimes the backing vocals will be lost even though there are backing vocals there."

- Gabox Mel [KaraokeGabox](#) model (uses Aufr's [config](#)) | [Colab](#)

"The lead vocals are good and clean!

While the backing tracks are lossy for this model, [it still] provide[s] great convenient for those who need LdV"

"The model doesn't keep the backing vocals below the main vocals, sometimes the backing vocals will be lost even though there are backing vocals there."

- anview [dereverb](#) Mel-Former model v2 ("on some songs I tried it worked better than karaoke models") not for duets, it debleeds too, "cleaner backing vocals [than the below], can sometimes mistake main vocals/delay for backing vocals more often than [the below]"
- karokee_4band_v2_sn a.k.a. UVR-MDX NET Karaoke 2 ([MVSEP](#) [MDX B (Karaoke)] / [Colab](#) / [UVR5 GUI](#) / [x-minus.pro](#)) - "best for keeping lead vocal detail" on its own, "cleaner main vocals, has significantly less bleeding than the MVSep counterpart", removes backing vocals from a track, but when we use min_mag_k it can return similar results to:
- Demix Pro (paid, "keeps more backing vocals [than Karaoke 2] (and somehow the lead vocals are also better most of the time, with fuller sound"

"Demix is better for keeping background vocals yes, but for the lead ones they tend to sound weaker (the spectrum isn't as full and has more holes than karaoke 2, but this isn't always a bad thing because the lead vocals themselves are cleaner, the mdx karaoke 2 might produce fuller lead vocals, but you will most certainly have some background vocals left too")

- Fusion model of Gabox Karaoke and Aufr33/viperx model on MVSEP (tend to confuse BV/LV more than single models)
- Gonzaluigi Karaoke fusion models ([standard](#) | [aggressive](#) | [yaml](#)) - also confuses BV/LV more
- MDX B Karaoke on mvsep.com (exclusive) - good, but as an alternative you could use MDX Karaoke 2 in UVR 5 (they are different)
"I personally wouldn't recommend 5/6_hp karaoke, except for using 5_hp karaoke as a last resort, you could also use the x minus bve model in uvr which sometimes is good with lead vocals"
- UVR-BVE-4B_SN-44100-1
- [Center extraction](#) model
- Melodyne [guide](#)
- RipX

(doesn't work for everyone)

- MDX-UVR Inst HQ_3 - new, best in removing background vocals from a song (e.g. from Kim Vocal 2)

Or consecutive models processing:

- Vocals (good vocal stem from e.g. voc_ft or 1296 or MDX23C single models or ensembles of MDX23C / MDX23 2.2 / UVR top/near top SDR / Ensemble of only vocal models: Kim 1, 2, voc_ft, MDX23C_D1581, eventually with demucs_ft
>The vocal result separated with->

Karaoke model -> Lead_Voc & Backing_Voc

[Tutorial](#)

(+ experimentally split stereo channels and separate them on their own, then join channels back)

- [arigato78 method](#)

"Karaoke 2 really won't pick up any chorus lead vocals EXCEPT for ad-libs

6-HP will pick up the melody, although it's usually muffled as hell"

"Q: is mdx karaoke 2 still the best for lead and back vocals' separation?

A: I'm finding it's the best for "fullness" but 6-HP picks up chorus melody while K2 only usually picks up ad-libs

I personally like mixing K2, 6-HP (and sometimes 5-HP if 6-HP sounds very thin) together also, let's say a verse has back vocals that are just the melody behind the lead vocal (instead of harmonies) for a doubling effect, sometimes K2 will still pick up both the lead and double."

AG89 avg ensemble:

UVR-5-1_4band_v4_ms_fullband_BVE_V2,
Karaoke_GaboxV2,
mel-band_karaoke_fusion_standard

Harmonies

For more layers from the above (e.g. starting with voc_ft/1296/Mel/BS [or other good model]>Karaoke 2 [or other])

- becruily & frazer BS-Roformer Karaoke (or [more](#))
- Stereo width feature for uvr bve v2 by setting it to 80% (it might use voc_ft as preprocessor already) - on x-minus.pro/uvronline.app
- [Medley Vox](#) (free, 24kHz SR model trained by Cyrus, [Colab](#), local installation [tutorial](#), more [info](#), use e.g. [AudioSR](#) afterwards; "help[ed] me separate a few layers harmonies on one song, though I had to keep mixing certain ones back together and run them through the same model to get a cleaner result")
- Sacial Mel-Roformer dereverb/echo model #3 called "fused": [model](#) | [yaml](#)
- [Melodyne](#) (paid, 30 days trial) - "the best way to ensure it's the correct voice", or
- Hit 'n' Mix [RipX](#) DAW Pro 7 (paid/trial)

In Melodyne it is "harder to do but can be cleaner since you can more easily deal with the incorrect harmonics than RipX sometimes chooses"

"every time I'd run a song through RipX I was only able to separate 4-5" harmonies (or also)

- (prob.) Revoice Pro 5
 - Mel-Roformer Karaoke (by becruily) [model file](#) (better than aufr's model)
 - Dango.ai (paid, might be still useful in certain situations)
- Choral Quartets F0 Extractor - "midi outputs, but it works"

For research

<https://c4dm.eecs.qmul.ac.uk/ChoralSep/>

<https://c4dm.eecs.qmul.ac.uk/EnsembleSet/> (similar results to MedleyVox)

For two singers in a duet from one song

(use on already separated [vocals](#))

- becruily & frazer BS Karaoke (sometimes can separate even 3 singers if the vocals aren't completely glued into one, especially if it's male and female - maxerv19,

"It's getting better than MedleyVox" - ryanz48)

- Becruly Mel Karaoke

- [MedleyVox](#) (trained by Cyrus, 24kHz SR - use e.g. [AudioSR](#) afterwards, MVSEP, [Colab](#), local installation [tutorial](#) [use vocals 238 model], more [info](#))

"best at it, but not guaranteed to work always. 5% chances of perfectly separating duets on MedleyVox, or else it always false detects and switches back and forth" "also does a pretty good job on other solo instruments"

- MVSEP Multispeaker model (Experimental section at the bottom)

Works well for rap overlapped with singing in one already separated vocal stem.

"Seems very picky with audio, most of the songs/files I tried didn't work

MedleyVox works on the other side (regardless that it's of lower quality)" - becruily

- MVSEP Male/Female:

a) Mel-Roformer Male/Female separation model 13.03 SDR

a) SCNet XL Male/Female separation model on MVSEP (same model base)

SDR on the same dataset: 11.8346 vs 6.5259 (Sucial)

Sometimes the old Sucial model might still do a better job at times, so feel free to experiment.

- Aufr33 BS-Roformer Male/Female beta [model](#) | [config](#) | [Colab](#) | x-minus (uses Kim-Mel-Band-Roformer-FT2 as preprocessor) | MVSEP

(based on BS-RoFormer Chorus Male Female by Sucial) SDR 8.18

- Male/female BS-Roformer [model](#) by Sucial | [config](#) for [UVR](#) | tensor match error [fix](#)

If they sing at intervals [one by one - not together] they cannot be separated. | MVSEP

- Mel-Roformer Karaoke on x-minus.pro (model files in [Karaoke](#))

- MDX-UVR Karaoke models

- VR's 5_HP or ev. 6_HP in UVR

- BVE v2 on x-minus (already uses voc_ft as preprocessor for separating vocals)

It might be still not enough, then continue and/or look for Dolby Atmos [rip](#) and retry; works for several backing vocals when lead vocal panning is set to center, "then running the bgv through bve v2 again, but this time set lead vocal panning to 80% but be aware the lead vocal quality will not be that good with this model" - Isling)

- Dry Paint Dealer Undr's Melband Roformer and Demucs Lead and Rhythm guitar [models](#)

- [duet-svs-diffusion](#) ("mono/16kHz, 24kHz sample rate, and quality is lower than MedleyVox models")

- RipX (paid)

- Melodyne (paid, "with polyphonic mode, with a lot of manual finetuning in the detection tab, and this can only work if the voices are not on same pitch").

- SpectraLayers 11 (but it's mainly dedicated for voice, not singing)

Spectral painting:

- [ISSE](#) (free, you can figure out which voice is who's just by frequencies alone; use on e.g. separated vocals too)
- RX Editor's brush ([video](#) by Bas)
- Audacity ([image](#)) less effective
- [Ampter](#)
- [Filter-Artist](#)
- [AudioPaint](#)

Notes

If artists sing the same notes, Karaoke models will rather not work in this case.

If BVs are heard in the center, don't use the MDX karaoke model but the VR karaoke model instead.

Use the chain algorithm with mdx (kar) v2 on x-minus which will use uvr (kar) v2 to solve the issue. It will be available after you process the song with MDX. (Aufr33/dca)

“The MDX models seem to have a cleaner separation between lead and backing/background vocals, but they often don't do any actual separation, meanwhile the VR models are less clean, but they seem to be better at detecting lead and background”

“MDX models basically require the lead to be completely center and the BV to be stereo whereas VR ones don't really care as much about stereo placement”

You could also ask [playdasegunda](#)/play da primeira/viperx for separation, as he has some decent private method/models for double vocals better than becruily model (although the latter can be still close), although newer frazer & becruily model is “no way inferior to the ViperX (Play da Segunda). The rumour says, viperx model on Play da Segunda was trained on 40GB dataset allegedly (it would be small), and the model can be bought for 500\$ when you contact via email. It's actually a set of models used for the final inference.
For the record, the open-sourced model by the duo costed 600\$ on compute and probably used a bigger dataset, achieved a bit smaller SDR, but it's a single model.

For research:

“These archs are [...] really promising for multiple speakers separation, and should be working for multiple singers separation if trained on singing voice:

<https://github.com/dmlguq456/SepReformer> (current SOTA)

<https://github.com/JusperLee/TDANet>

<https://github.com/alibabasglab/MossFormer2>”

> Separating two main vocals

E.g. one panned about 30% left and the other right

- “use bve v2 and click the “lead vocal panning” button” on x-minus premium

For vocals with vocoder

- voc_ft

Alternatively, you can use:

- 5HP Karaoke (e.g. with aggression settings raised up) or
- Karaoke 2 model (UVR5 or Colabs). Try out separating the result obtained with voc_ft as well.
- BS-Roformer model ver. 2024.04 on MVSEP (better on vocoder than the viperx' model).

"If you have a track with 3 different vocal layers at different parts, it's better to only isolate the parts with 'two voices at once' so to speak"

Various speakers' isolation (from e.g. podcast or movie)

- MVSEP Male/Female SCNet model
- MVSEP Male/Female MelRoformer model
- Aufr33 BS-Roformer Male/Female beta [model](#) | [config](#) | [Colab](#) (based on the model below)
- Male/female BS-Roformer [model](#) by Sacial | [config](#) for UVR | tensor match error [fix](#)
(if they sing at intervals [one by one] they cannot be separated)
- Multispeaker model on MVSEP
- BS-Roformer becruily & frazer Karaoke model

-
- [Guide and script for WhisperX](#)

- <https://github.com/alexlnkp/Easy-Audio-Diarisation>
- [Spectralayers](#) 11's unmix multiple voices option

(for further research) - some of these tools might get useful:

<https://github.com/dmlguq456/SepReformer> (SOTA for 2 speakers)

<https://paperswithcode.com/task/speaker-separation/latest>

<https://arxiv.org/abs/2301.13341>

<https://paperswithcode.com/task/multi-speaker-source-separation/latest>

> 4-6 stems (drums, bass, others, vocals + opt. **guitar, piano**):

- Currently when used on AI-generated music, usually hihats will be left behind.
- You might want to use the *already well-sounding instrumental*, possessed with 2 stem model in the section above first, and then separate using the following models.
- Furthermore, you can slow down your song by x0.75 speed - the result can be - more elements in other stem and better snaps and human claps using 4 stems.

Read the [Tips to enhance separation #4](#) for more.

[Logic Pro \(May 2025 update\)](#) / BS-Roformer SW 6 stem | MVSEP | uvronline

SDR bass 14.57, drums 14.05, piano 7.79, guitar 9.00, other 8.66, vocals 11.27

Currently, the best SDR for all stems but vocals, and drums have lower fullness than MVSep SCNet XL drums 14.26 vs 21.21). Excellent guitar and piano.

“guitar model sounds better than demucs, mvsep, and moises” - Sausum

“it's not a fullness emphasis or anything, but it's shockingly good at understanding different types of instruments and keeping them consistent sounding” - becruily
vocals doesn't have the biggest metric, but are good for deep voices.

Drums lacks some fullness but “I got better drums/bass separation with that model than with any others when input is some live/rehearsal recordings with shitty sound” - jarredou

Although, compared with Mel-Roformer drums on uvronline:

“separates far far better when it's programmed instruments compared to actual recorded ones” - isling

“Roformer SW is putting finger snaps and foot taps as vocals and in the vocal stems.” - GodzFire

“gets not just drums but anything percussive/non-melodic. Which I personally don't mind, but yeah it does cause problems with [drumsep](#) models because they're only expecting standard drums.”

Bass can be occasionally worse vs demucs_ft as bass stem Demucs “considers not only spectrograms but also waveforms”.

“can't differ an electric bass with pedal effect from an electric guitar” - qraiqu

“better than LalalAI by a long shot too” - nowarrantywarren

As for MVSep “just as good a job on vocals as the paid version's ensemble preset.”

In UVR it takes two hours for overlap 11 for 4 minute file to separate on 6700 XT. Keep it at 2 (the fastest) - it will take also 2 hours, but on only i3-7100u (GPU Conversion disabled).

Highest SDR with 882000 chunk_size.

- MDX23 v.2.5 by ZFTurbo, fork by jarredou ([Colab](#); 4 stems when it's enabled)

Multisong dataset SDR bass: 12.58, drums: 11.97, other: 7.28, vocals: 11.10 (v2.4)

It's weighted ensemble of various 4 stem Demucs models with weighted ensemble of 2 stem models for 4 stem input, so the metrics for RAW 4 stems output (without getting instrumental from ensemble first) will be a bit lower, and more for other stem - even by 1.38+, and 0.24+ for bass, and 0.02+ for drums ([read more](#)).

~"compared to this, demucs_ft drums sound like compressed"

- Ensemble 2/4/8 stems ([MVSEP](#), paid) - similar or better results with newer single stem models combined, various ensembles to choose from, freedom to experiment.

[Read](#) all the *ensembles metrics sorted by instrumental bleedless*.

Chained separation order

With single stem models below, feel free to experiment with different orders of sequential stem separation:

#1

1) Instrumental model first 2) then drums or bass 3) piano or guitar 4) strings or horns
If your song has “weird percussion that don't get picked up by drum models and if it's piano-heavy, I would go for piano first, but it sometimes leaves piano behind” - lsling

#2

1) Instrumental model first 2) drums 3) piano 4) strings or horns 5) bass 6) guitars
This way once someone “ended up with a decent 5th-stage “other” stem that seemed to contain some unknown synth sounds + possible orchestra hits” vs when bass was after drums when “gave me a messy “other” with stray piano & strings content” SilSinn9821

#3

ZFTurbo: “After each remove, you also remove some useful parts for next instruments. So I'd propose to first use the models with the best quality. Anyway, in the end others can be very muddy”.

#4

(dynamic64)

“I think this is my new default [order]:

Bass Ensemble (SCNet XL + BS Roformer + HTDemucs4), Drums MVSep SCNet XL, Piano MVSep SCNet Large, Organ MelRoformer, Saxophone MelRoformer, MVSep Wind Ensemble (SCNet + Mel), MVSep Guitar Ensemble (BSRoformer+ MelRoformer), MVSep Strings”.

Example: “Personally, I like putting the instrumental through mvsep bass model (above), then putting the other stem through mvsep drums (specifically SCNet XL), then putting the other stem of that through the 6 stem model [a.k.a. SW]

The 6 stem model is best for piano and guitar, and separating out the drums and bass beforehand helps that model not have to work as hard.

And it also allows you to manually add anything that the first 2 models missed, because its a 6 stem model, and re-separates any missed drum and bass.

Creates close to studio results. As close to studio as I've ever heard”

#5

In case of ensembling various models of the same stem, if the top model in terms of SDR doesn't have significantly lower metric, sometimes it's better to use it instead of ensemble [esp. if it doesn't have any crossbleeding] (thx dynamic64).

Just be aware that it might vary from song to song

#6*

“Drums model often bleeds the bass and 808s into the drum track, [using bass model] prevents this issue from happening”

Single **drums** models

- Ensemble: MVSEP Drums Mel + SCNet XL ~“Usually works the best” - dynamic64, but you might want to save intermediate files and split it into different fragments to get the best of both. Consider using instrumental model result from e.g. bctruiuy model as input.

- Drums only/other SCNet XL model by ZFTurbo on MVSEP, SDR 13.72

“Very hit or miss. When they’re good they’re really good, but when they’re bad there’s nothing you can do other than use a different model” - dynamic64

- Drums only/other Mel-Roformer model by viperx on x-minus.pro (occasionally might work only with [this](#) link), 12.54 SDR

“compared to demucs_ft it was too muddy and had too much bleed at the same time both mvsep and xminus” - isling

“I got some bad results (that could ruin the ensemble mode). On these tracks, uvrone [x-minus] melband drums model was giving better results” - jarredou

Previously the best

- Drums only/other SCNet Large (“x-minus’ Mel band drums model is better” - drypaintdealerundr)

- Drums only/other Mel-Roformer model by ZFTurbo on mvsep.com, 12.76 SDR
Older ZF’s model

- 1053 BS-Roformer drums/bass model by viperx in UVR Roformer beta or [Colab](#).

Very good drums with bass in one stem model - use instrumental as input to avoid vocal residues)

More [metrics](#) for drums models.

- MVSep Percussion

- MVSep Tambourine

- MVSep Timpani

- MVSep Congas

Single **bass** models

- MVSEP Bass SCNet XL (the best 13.81 SDR, “It passes Food Mart - Tomodachi Life test. That’s the first model to”,

- Ensemble of SCNet XL, BS and HTDemucs4 models (SDR 14.07); SCNet can be sometimes worse than Demucs which “considers not only spectrograms but also waveforms” - Unwa)

After separation, you might want to then apply Mel-RoFormer De-noise to remove the high noise, and finish with Apollo Universal by Lew ([model](#) | [Colab](#)) to get more clarity (Tobias51).

- MVSEP Bass BS Roformer 12.49 (integrated to 2 stem and multi/All-in stem ensemble too, worse other stem vs the below - more “empty” than below, problems with getting even results when bass contains a low pass filter with high resonances, picks up more “actual bass guitar than x-minus” model - drypaintdealerundr)
- x-minus.pro BS (much better at treble-heavy bass tones than demucs, better other stem than above, “catches higher end and synth basses. It makes it sound cleaner” although might sound “weird and muddy” compared to demucs_ft at times, and often when it does not capture synth bass, the mvsep bass will do - isling)
- MVSEP Bass BS Roformer SW - might be worse than the above, at least the ensemble with SW model is not better than the 14.07 - isling)
- <https://twoshot.app/model/548> (paid)

- MVSep **Double Bass** model
“The BS-Roformer SW bass model should probably be used first to extract the double bass. Creates a better sound” - dynamic64

- MVSep **Synth** (also, it can sometimes pick up bass in places where regular model’s can’t)

4 stems in one model

- Demucs_ft (4 stem) - the best single Demucs’ model ([Colab](#) / [MVSEP](#) / [UVR5 GUI](#))
Multisong dataset SDR 9.48: bass: 12.24, drums: 11.41, other: 5.84, vocals: 8.43
(shifts=1, overlap=0.95)
Better drums and vocals than in Demucs 6 stem model, decent **acoustic guitar** results in 6s. Good bass stem as Demucs “considers not only spectrograms but also waveforms”. For 4 stems alternatively check MDX_extra, generally Demucs 6 stem model is worse than MDX-B (a.k.a. Leaderboard B) 4 stem model released with [MDX-Net arch](#) from MDX21 competition (kuielab_b_x.onnx in this [Colab](#)), and is also faster than Demucs 6s.
For Demucs use overlap 0.1 if you have instrumental instead of mixture mixed with vocals as input (at least it works with ft model) and shifts 10 or higher. For normal use case (not instrumentals input) it will give more vocal residues, overlap 0.75 is max reasonable speed-wise, as a last resort 0.95, with shifts 10-20 max.

- [SCNet-large_starrytong](#) model (4 stems) ([Colab](#) or [MSST-GUI](#))
Multisong dataset SDR 9.29: bass: 11.28, drums: 11.24, other: 5.58, vocals: 9.06 (overlap: 4)
It’s 3x faster than Demucs (Nvidia GPU or CPU-only) and sounds better for some people, except for bass. [SDR](#)-wise, vocals are better than in demucs_ft (which is low vs single vocal/inst models anyway). Better SDR than starrytong’s MUSDB18 and Mamba models.

- Ableton 12.3 update’s high quality option separation (4 stems) - slow, works only on CPU, probably utilizes BS-Roformer. Better bleedless than ZFTurbo SCNet XL undertrained public model below, and better SDR. Might take 20 minutes for 2 minutes separation on slower CPUs (iirc mobile Sandy/Ivy). The default separation mode has very low metrics.

- [SCNet XL IHF](#) model (4 stems) by ZFTurbo

Multisong dataset SDR 9.93: bass: 11.94, drums: 11.58, other: 6.49, vocals: 9.69

- [SCNet XL](#) model (4 stems) by ZFTurbo ([Colab](#) | [MSST-GUI](#) | [MSST](#) | [UVR beta patch](#))

Multisong dataset SDR 9.72: bass: 11.87, drums: 11.49, other: 6.19, vocals: 9.32

Better metrics than the starrytong model but “downgrade to the Large model since it produces a *** ton of buzzing” due to undertraining.

Only bass is better in Demucs_ft - 12.24, although drums might be still better in demucs_ft.
2 stem model on MVSEP is further trained iirc.

- [BS-Roformer](#) model (4 stems) by ZFTurbo | [MSST-GUI](#)

Multisong dataset SDR 9.38: bass: 11.08, drums: 11.29, other: 5.96, vocals: 9.19

Trained on MUSDB18HQ

- Mel-Roformer [models](#) (4 stems) by Aname

a) Large (4GB): multisong dataset SDR drums: 9.72, bass: 9.40, other: 5.11

b) XL (7GB):

Despite lower AVG SDR on musdb18 dataset (8.54 vs 9), it seems to outperform demucs_ft model (only other stem has better SDR in demucs_ft - all other metrics are better in SCNet/XL/BS-Roformer).

XL is “heavy and slow, without giving a quality boost compared to existing public 4 stems models trained on musdb18 by ZFTurbo and starrytong (BsRofo and SCNet large/XL [above])” - jarredou (maybe minus buzzing in the public XL model).

“Drums are sounding really good in particular, tested a couple songs with the large model after using unwa's v1e+ for instrumental” “drums are absolutely the standout”.

“The bass stem is definitely the weakest one from this new model. Very, very muddy and inconsistent.” - santilli_

“Large [variant] works in like 99% use case” “Large split sounds amazing so far tho”

XL “result would take so much longer, but the large results sounded better IMO” - 5B

XL model won't work with default settings on Colab, and very slow on e.g. RTX 3060, “on 4070 Super it took like 6 mins on XL 4 stems compared to 30 seconds on Large 4 stems”

- [SCNet Tran](#) a.k.a. small model (4 stems) by ZFTurbo

Multisong dataset SDR bass: 10.99, drums: 10.87, other: 5.63, vocals: 8.42

Outperformed by the above models at least SDR-wise. Cannot be used in UVR.

- [KUIELab-MDXNET23C](#) (4 stems) - its first scores were probably from ensemble of its five models, and in that configuration it had better SDR than demucs_ft on its own, and drums had better SDR than “SCNet-large_starrytong” above (so single models' score of any of these MDX23C models is probably lower than in demucs_ft).

- Lighter “model1” drums sounds surprisingly better than htodemucs non_ft v4 on previously separated instrumental. It handles trap really well and preserves hi-hats correctly, but in cost of other stem bleeding. v4 model can be used to clean it a bit further, but at least using GPU Conversion on AMD and older directml.dll for some GPUs, it adds more noise/artefacts, so use CPU in that case (tested on Roformer as preprocessor for instrumental). It's relatively

fast, but not as mdx_extra (which sounds rather lo-fi in that case).

- Bigger “model2” is heavier and doesn’t work on at least AMD 4GB VRAM GPUs on Beta Roformer patch #2 (before the introduction of the new overlap code).

To run model1/2 in UVR “You must change the model names in "mdx_C" from "ckpt" [name] to model1.ckpt, model2.ckpt, & model3.ckpt. [so simply add the name to the extension]” and then copy the ckpts to models\MDX-Net without yaml. There are actually 3 mdx23c models there (and 2 demucs), but model3 seems to be only for vocals (and with low SDR). So the two of three most important were explained above.

OG KUIELab’s [repo](#).

- [model mdx23c ep 168 sdr 7.0207](#) (4 stems)

Multisong dataset SDR bass: 8.40, drums: 7.73, other: 4.57, vocals: 7.36

4 stems, also trained on MUSDB18HQ, but by ZFTurbo, it’s different from the above, similar size to “model2”.

- Aname 4 stem BS-Roformer [model](#) | [yaml](#)

Multisong dataset SDR bass: 9.79, drums: 10.21, other: 5.27, vocals: 9.13

It has better SDR than the 7.0207 above (as in the SDR metrics link below), but worse than demucs_ft and BS-Roformer 4 stem ZFTurbo model above.

- BS-RoFormer 4 stems model by yukunelatyh / SYH99999 added on x-minus

<https://uvronline.app/ai?discordtest>

Multisong dataset SDR bass: 8.68, drums: 10.37, other: 5.05, vocals: 8.57

Some people like it more than Demucs, but “it’s like demucs v4 but worse, I think.

The vocals have a ton of bleed, the bass is disappointing tbh.

The other stem has a ton of bgv and adlib bleed in it” Isling

It has [SDR metrics](#) for all stems worse than 4 stem BS-Roformer by ZFTurbo and demuicks_ft.

- v2 of it was added on site with lower metrics for all stems in later period.

Smaller public 4 stem models and all metrics:

https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/docs/pretrained_models.md#multi-stem-models

- [GSEP AI](#) (2-4-6 stems, sonically it used to have the best other stem vs Demucs, also piano in Demucs is worse, and it picks up e-piano more frequently, GSEP **electric guitar** model doesn’t include acoustic, it’s only electric). In general, it used to have a very good **piano** model before many alternatives existed

- [Ripple](#) (defunct; and used to be for US iOS, at one point best SDR for all-in one single 4 stem (besides the other stem), but bad, bloody other stem. Back then could be the best bass stem, and/or kick in drums, but not the best drums in overall vs demucs_ft, “you need something to get the rest of the drums out of the ‘other’ stem and at that point might as well use a proper drum model”, good vocals. You can minimize residues in Ripple by providing an already well separated instrumental from the section above, and/or minimizing volume of the input file by 3/6dB.

- [Bandlab Splitter](#) (4-6 stem - guitar, piano, web and iOS/Android app) - previously could be used e.g. for cleaning stems from other services, 48kHz output, stems can be misaligned, the quality got worse since one of the updates (rather not worth using anymore)

- [Audioshake](#) (paid, only non-copyrighted music, or slowed down songs [see workaround in "paid" above]) - sometimes better results than Demucs ft model.

- Spectralayers 10 - mainly for bass and drum separation -
"I think I've got some really comparable samples out of jarredou's MDX23 Colab fork", but for vocals and instrumentals it's mediocre [in Spectralayers 10].

- music.ai - "Bass was a fair bit better than Demucs HT, Drums about the same. Guitars were very good though. Vocal was almost the same as my cleaned up work. (...) I'd say a little clearer than MVSEP 4 ensemble. It seems to get the instrument bleed out quite well, (...) An engineer I've worked with demixed to almost the same results, it took me a few hours and achieve it [in] 39 seconds" Sam Hocking

- dango.ai (<https://tuanziai.com/en-US>) - also has 4 or more stems separation (expensive)

- (old) MDX23 1.0 by ZFTurbo 4 stems (Colab, desktop app, as above, much cleaner vs demucs_ft, less aggressive, but in 1.0 more low volume vocal residues in completely quiet places in instrumentals vs e.g. HQ_3, instrumentals as input should sound similar to the current v. 2.4 fork as only 4 stem separation code didn't change much since then)

- MVSEP has also single **piano, guitar and bass** models (in many cases, guitar model can pick up piano better than piano model;
"works great for songs with grand piano, but only grand piano, since that's what it was trained on.
Same with guitar, which catches more piano than piano model does, ironically").

- x-minus/uvronline.app has also single acoustic and guitar models by viperx
- viperx' piano model is also on x-minus/uvronline.app.
(More on piano and guitar models later below)

To enhance 4 stem results, you can use good instrumental obtained from other source as input for the above (before instrumental Roformers it could be e.g. [KaraFan](#), and its different presets ensembled in UVR5 app with Audio Tools>Manual Ensemble)

For the best results for piano or guitar models, use other stem from 4 stems from e.g. "Ensemble 8 models" or MDX23 Colab or htodemucs_ft as input.

Moises.ai - although drums might be better using e.g. "MVSep Drums" already, probably vs Mel-Roformer on MVSEP or x-minus (not sure) [Moises] can give "better results (...) if the input material is for example cassette-tape sourced or post-FM).

FL Studio - rather nothing better than the solutions above

Older 4 stem models in UVR (for some specific songs, e.g. while trying to fix bleeding across stems):

htdemucs
htdemucs_mmi
mdx_extra
kuielab_b

>Sep. parts of drums

(kick/hi-hat/snare/toms/...)

For drums stem from e.g. MDX23 Colab/instrumental model>Demucs_ft/GSEP or drums model (e.g. on x-minus/MVSEP)

- MVSEP 8 stems ensemble of all the 4 drumsep models below (so besides Demucs model by Imagoy) [metrics](#)
- MVSEP's SCNet 4 stem (kick, snare, toms, cymbals) best SDR for kick and similar to 6s below for toms: -0.01 SDR difference)
- MVSEP's SCNet 5 stem (cymbals, hi-hat, kick, snare, toms)
- MVSEP's SCNet 6 stem model (ride, crash, hi-hat, kick, snare, toms) worse snare SDR
- [jarredou/Aufr33 MDX23C model](#) (kick/hi-hat/snare/toms/ride and crash stems); worse overall SDR, but it's a public model usable in UVR or inference Colab
- SpectraLayers 11 (Unmix Drums)>OG [drumsep](#) by Imagoy/>[FactorSynth](#) (depending on how far you want to unmix the drums) >Regroover>UnMixingStation (all the last three paid)> Virtual DJ (Stems 2.0, barely, or doesn't pick those instruments at all).
- [LarsNet](#) (vs OG drumsep, it also allows separating hi-hats and cymbals, toms might be better)
- RipX (paid)
- SpectraLayers 10 (paid, sometimes worse, sometimes better than OG Drumsep. IDK if it was added in update or main version) "drumsep works a f* ton better when separating on this one song I've tested with the pitch shifted down 2"
- FADR.com (in paid subscription)
- Moises.ai (only for pro)

Compared to OG drumsep, Regroover allows more separations, especially when used multiple times, so allows removing parts of kicks, parts of snares etc, noises etc. More deep control. Plus, it nulls easily. But drumsep sounds better on its own, especially with higher parameters like e.g. shifts 20 and overlap 0.75-0.98. Now it can be replaced by the public MDX23C model.

Strings

- Dango.ai (e.g. violin, erhu; paid, free 30 seconds fragments) - "impressive results" for at least violin

- Music.ai (paid, “Dango.ai and Music.ai [previously] the best strings models [Dango sounds fuller meanwhile Music.ai has more accurate recognition of strings but sounds a bit too filtered]” - from before BS Strings release)
- MVSEP Bowed Strings BS-Roformer (“doesn’t disappoint”, SDR 5.41)
- MVSep Plucked Strings
- Moises.ai (paid, not bad)
- Audioshake

- MVSEP Violin (sometimes does better than the strings model for strings and has also bigger SDR on strings dataset)
- MVSEP Strings MDX23C (it’s “weak”, SDR 3.84)
- x-minus.pro/Uvronline.app Mel-Roformer model by viperx (SDR 2.87) (sometimes with this [link](#) or [this link](#))

- [Demix Pro](#) (paid, free trial)
- [RipX DeepRemix](#) (once was told to be the best bass model, but it doesn’t score that good SDR-wise, probably it’s Demucs 3 (demucs_extra) and is worse than Demucs_ft and rather also vs MDX23 above; could have been updated) (paid)

- Sometimes Wind model in UVR5 GUI picks up strings

- MVSEP Harp
- MVSep Mandolin
- MVSep Banjo
- MVSep Sitar
- MVSep Ukulele
- MVSep Dobro

Violin

- MVSep Violin BS Roformer

“it can even separate violin quartets from cellos, so cool.” - smilewasfound

“Very neat model. (...) Sometimes the model does seem to pick up more than just violins imo, but yeah for separating high strings in particular it is really cool.” - Musicalman
- Dango.ai “impressive”

- MVSEP Viola

- MVSEP Chello

Electric guitar

“For better results you might try first removing vocals.”

[Audioshake](#)>[RipX](#)>[Demix Pro](#)>[lalal.ai](#) (e.g. lead guitars; the model got better by the time) (they're paid ones)/

Logic Pro>GSEP>Demucs 6s (free)<[Moises.ai](#) (paid “holy shit better [vs demucs, but] still pretty bad”)

[Dango.ai](#) (paid)

[Music.ai](#) (paid, free trial)

Logic Pro (paid, May 2025 update) / BS-Roformer 6 stems a.k.a. MVSEP Guitar SW (“really on point. So far it separated super well, also didn’t confuse organs for guitars and certain piano sounds as well.” - Tobias51

“guitar model sounds better than Demcus, MVsep, and Moises” - Sausum

“guitar in particular was amazing. All other models I tried had trouble with it” - Musicalman)

|||||

MVSep Electric Guitar (“really neat. One thing I noticed is that it seems to be better than other models at picking up midi/synth lead guitars (...) also gets tripped up a bit more by weird FX and synth sounds being partially flagged as guitar” - Musicalman)

[Becruily Melband guitar | yaml](#) (“Not SOTA, but much more efficient and comparable to existing guitar models, and for some songs it might work better because it picks up more guitars [though it can also pick some other instruments].”)

[Mvsep.com](#) (Mel-Roformer model and the previous - MDX23C one. Mel is “pretty good but suffers some dropouts where MDX23C doesn’t”)

uvronline.app (Mel-Roformer viperx' model, it is not flawless either)

[uvronline.app](#) (HQ_5 beta/[paid users](#) - places guitars in vocal stem pretty well)

“Rebalance volume of chans before processing” if you have better separation results processing L and R channel separately.

Consider using Apollo Universal by Lew ([model](#) | [UVR](#) | Colab in [HTMYOR](#)) to get more clarity after separation.

Acoustic guitar

- [dango.ai](#) (paid, probably the best for now, better in at least some songs than the SW model)

- MVSep Acoustic Guitar (strong competitor, outperforms moises “like crazy”)

- uvronline.app (viperx' model for premium users - does a good job too)

- BS-Roformer SW

- Demucs 6s - sometimes, when it picks it up

- [GSEP](#) - when the guitar model works at all (it usually grabs the electric), the remaining ‘other’ stem often is a great way to hear acoustic guitar layers that are otherwise hidden.”.

- [lalal.ai](#) (both paid)>[moises.ai](#) (It picks up acoustic and electric guitar together)

- Audioshake (both electric and acoustic)
- moises.ai

Separating electric and acoustic guitar

- Use a model from one of two the categories above, e.g.
 - MVSep Acoustic Guitar ("it's separating acoustic from electric very well, even in fuzzy, lo-fi recordings" - Input Output)
 - "To separate electric and acoustic guitar, you can run a song [e.g. other stem] through the Demucs guitar model and then process the guitar stem with GSEP [or MVSEP model instead of one of these]."
- GSEP only can separate electric guitar so far, so the acoustic one will stay in the "other" stem."
- "[medley-vox](#) main vs rest model has worked for me to separate two guitars before"
 - moises.ai "it's not perfect, it's good when the solo guitar for example is loud then it can be isolated but when it comes in a balanced lead and rhythm guitar, it can't isolate it"
 - MDX23C [phantom center](#) model
 - [moises.ai](#) (it has electric, acoustic, rhythmic, solo)

Lead and rhythm guitar

- moises.ai (paid)
- MVSep Lead/Rhythm Guitar (1 stage, and 2 stage variant)
- MVSEP guitar models

"I can isolate both guitars with the different models that MVSEP has, especially in rock tracks where the lead guitar is in the center channel and the rhythm guitar is on the right - left side of a stereo track, good results are not always obtained, especially when the lead guitar has long delay effects, tons of reverb or when these effects go from one channel to another, but it also depends on how the song was mixed." - edreamer 7

- MDX23C [phantom centre](#) extraction by wesleyr36 model
- "First, isolate guitar, then (...) use phantom centre extraction by wesleyr36 model. Here I can find the rytmn and the lead guitars, as I told before, results can vary" - edreamer 7
- Dry Paint Dealer Undr's Melband Roformer and Demucs Lead and Rhythm guitar [models](#).
- "my own very mediocre model for it that I never shared. it does work but has issues that I imagine any better executed model won't."
- lalal.ai ("it sucks" - isling, Oct 25)

Wind instruments and wind noises

(*trumpet/saxophone/brass/woodwinds/flute/trombone/horn/clarinet/oboe/harmonica/bagpipe/s/bassoon/tuba/kazoo/piccolo/fluge/horn/ocarina/shakuhachi/melodica/reeds/didgeridoo/mussette/gaida/farts*)

- MVSep Wind BS Roforomer (2025.09) 9.77 SDR (+2.64 SDR)
- MVSep Wind BS Roformer (2025.08) (more robust and cleaner than the Mel and detects instruments better, +2.5 SDR)
- Wind BS-Roformer on x-minus.pro by viperx (big step forward vs the old UVR model)
- MVSEP Wind SCNet

- MVSEP Wind Mel-Roformer
- MVSEP Trumpet (“so clean”)

“after testing [Wind 9.77] on a song where trumpet and sax play in unison, doing the trumpet model is cleaner than doing the sax model” - dynamic64
- MVSep Trombone
- MVSep Oboe
- MVSep Clarinet
- MVSep Harmonica (“Hit or miss” - musicbybrooks)
- MVSep French Horn
- MVSep Tuba
- MVSep Bassoon
- MVSep Accordion
- MVSep Brass
- MVSep Woodwind

Older

- "Wind" model on UVR5 (Download Center -> VR Models -> select model 17)
(You might have to use it on instrumental separation first, e.g. with HQ_4 or Kim Inst)
- Audioshake
- Music.ai
- Adobe Podcast
- Karaoke 4band_v2_sn on e.g. MVSEP (worse than Wind model in UVR)
- Probably someone had some success with one de-crowd model for wind noises
- Lot of instrumental/vocal models confuses wind instruments with vocals

MVSEP Saxophone

- SCNet XL (SDR saxophone: 6.15, other: 18.87)
- MelBand Roformer (SDR saxophone: 6.97, other 19.70)
- Ensemble Mel + SCNet (SDR saxophone: 7.13, other 19.77)

Piano

Consider using *Unwa BS-Roformer Resurrection inst* a.k.a. “*unwa high fullness inst*” on MVSEP as preprocessor - *rainboomdash*

- Logic Pro (paid; May 2025 update, SDR 7.79) / BS-Roformer 6 stems / MVSEP Piano SW (“1000 times more efficient than the Lalal.ai piano model”)
- Lalal.ai (paid; no other stem with piano stem attached)
- Demix Pro (paid; “I often combine the two” - Mixman, but it was before the 6 stems above)
- MVSep Piano Ensemble (Mel-Roformer + SCNet Large piano models; SDR 6.21)
(Mel is viperx' iirc; “a [tiny] bit more bleed during the choruses and whatnot” vs x-minus, “works well maybe 7 times out of 10”; SCNet, “has a less watery sound, but more bleed” vs Mel)
- x-minus.pro (for paid users; cheap subscription; “more consistent than MVSep piano and demucs_6s” it knows well what piano is, but it sounds the best for other stem of piano separation, but e.g. on Carpenters - Yesterday Once More “while not terrible, the dropouts, underwater ‘gurgles’, and general lack of piano punch/presence remains noticeable” - Chris_tang1, while MVSEP Piano Ensemble: SCNet + Mel, SDR: 6.21, was much better in that case - might vary on a song)
- Music.ai (paid)
- Dango.ai (paid)
- GSEP (formerly best, paid)
- Moises (separate models for piano and keys)
- MVSep Piano MDX23C 2024 & 2023
- htdemucs_6s (not too good)
- MVSep Digital Piano (much better for epiano, and sometimes also for real or synthesiser when it's picked vs the SW model)
- MVSep Keys
- MVSep Harpsichord

Synths

- MVSep Synth (it can also pick some bass which some bass models failed to pick up)
- MVSep Organ
- Piano or guitar models might work (if the song doesn't have piano or guitar already) depends on a song
- [Zero Shot](#)
- lalal.ai (hit or miss, sometimes might not work)
- Some voc/inst models might treat synths as vocals (then you could separate them using better inst/voc model)

Organs

- MVSEP Organ (surprisingly good, and since then SDR doubled since the first version, eliminating some bleed issues or e.g. Hammond organs not being picked in some places)

Idiophones

- MVSep Marimba
- MVSep Glockenspiel
- MVSep Triangle
- MVSep Bells (tubular bells or chimes - for sleigh bells use drums model)
- MVSep Wind Chimes

Crowd

- UVR-MDX-NET Crowd HQ 1 (UVR/x-minus.pro/[model](#)/conf in UVR) (can be more effective than MVSEP's sometimes; e.g. good for live shows)
 - Mel-Roformer De-Crowd by ZFTurbo (MVSEP/[files](#)/UVR)
To use it in UVR, Go to UVR\models folder, and paste [that](#) folder there.
Then change "dim_t" value to 801 at the very bottom of:
"model_mel_band_roformer_crowd.yaml" file in "mdx_c_configs" subfolder. Don't use overlap above 4.
 - Mel-Roformer De-Crowd by Aufr33/viperx (x-minus.pro/UVR/[DL](#)/[conf](#))
For UVR, change the model name to the one from the attached yaml, copy chkpt to models\MDX_Net_Models, and yaml to model_data subfolder, then set overlap 2 or use ZFTurbo inference [script](#)] - more effective than MDX below at times)
 - MDX23C De-crowd v1/v2 (MVSEP)
 - Older MVSEP model (applause, clapping, whistling, noise)
 - Aufr33's Mel-Roformer Denoise average variant ([link](#) | [yaml](#) | [Colab](#)) can be also used as crowd removal
-
- [AudioSep](#)
 - [USS Bytedance](#)
 - [Zero Shot Audio Source Separation](#)
 - GSEP (sometimes), and e.g. drums stem is able to remove applauses
 - [Chant model](#) (by HV, VR arch, e.g. works for applauses; may leave some echo to separate with other models or tools below) for Colab usage - you need to copy that model to models/v5 and then use 1 band 44100 param, turn off auto-detect arch and set it to "default". In UVR pick one of 44100 1 band parameter, possibly 512.

"For really difficult live songs (where the crowd is overwhelmingly loud to the point where you can't hear the band properly) sometimes filtering vocals with mel roformer on xminus THEN running the vocals stem through the mdx decrowd model sometimes helps"

SFX

- "You do need to first get an instrumental with a different model, because this isn't really trained to remove vocals. Just SFX" or speech.
- SFX models can be more aggressive than regular vocal models for [speech](#). Sometimes, some of the regular vocal models may turn out to be better suited for your task, so try out those for speech too.
- BS-Roformer SW drums - "really good to remove some SFX and foley, way better than DnR v3" - erosunica
- MVSEP DNnR v3:
 - a) SCNet
 - b) MelBand(better metrics than Bandit v2)
- Bandit v2 (MVSEP | OG [weights](#) | [yaml](#)) multilingual model (multi) or single ones for EN/GER/FR/SPA/CH/FAR language.
All [models](#) converted ([yaml](#)) for ZFTurbo inference | [Colab](#) |
in [UVR](#)>MDX-Net>Download More Models>Bandit v2/Plus
(v2 is better on speech vs Bandit Plus, not always on SFX - experiment).
"Multilingual model is most of the time giving better results than French model for French content, so I would start with it" - jarredou

"Co-developed by Netflix and Georgia Institute of Technology. The [paper](#) is titled "A Generalized Bandsplit Neural Network for Cinematic Audio Source Separation"

Older models

- Bandit Plus (may work good for TV shows and movies: MVSEP, jarredou [Colab](#) | [UVR](#) "trained on mono audio, so it's dual mono (...) when content is heavily panned left/right, it's where issues start to say "hello !"" but other than that it might still handle stereo good enough or even better than others depending more on a song)

- "My suggestions from those who want best results, i suggest using either: ressurection inst or instfv8 to extract Music and effects audio track" - killdubo
- jazzpear94 Mel-RoFormer model (ability to separate specific SFX groups - Ambiance, Foley, Explosions, Toon, Footsteps, Fighting and General - for all in one stem | MVSEP, fixed newer [Colab](#), [files](#), [instruction](#), prob. broken Colabs: [1](#), [2](#), [3](#), [4](#))
- joowon bandit model: <https://github.com/karnwatcharasupat/bandit>

(better SDR for Cinematic Audio Source Separation (dialogue, effect, music) than DNR Demucs 4 below (SDR 10.16>11.47) - [Colab](#) / MVSEP)

- GAudio (a.k.a. GSEP) announced their SFX (DnR) model in their API: "DME Separation (Dialogue, Music, Effects)"

So far it's not available for everyone on their regular site:

<https://studio.gaudiolab.io/>

But the link on their Discord redirects to the site with a form to write an inquiry:

<https://www.gaudiolab.com/developers>

Shortly after entering the one or both of the links and logging on the first, you might get an email that \$20 of free credits to access their API have been added to your account

- [USS-ByteDance](#) (for providing any, at least, proper sample)

- [Zero Shot](#) (currently worse for SFX vs Bytedance)

- [Audiosep](#)

- custom stem separation on Dango (paid, 10 seconds for free)

- DNR Demucs 4 model (repo: [CDX23](#) repo, MVSEP, [Colab](#)) - it used to output fake stereo (at least training dataset was mono).

"I noticed [it] doesn't do well and doesn't detect water sounds, and fire sounds"

Can be used in UVR (the three stems will be labelled wrong, SFX will be bass).

>For UVR, download the [model files](#), put them in the Ultimate Vocal

Remover\models\Demucs_Models\v3_v4_repo, Delete "97d170e1-" from all the three file names, copy this [yaml](#) alongside the model files (it won't work on AMD 4GB VRAM GPUs).

>The Colab might run occasionally on CPU, and then it might be slow to the point that it might take 2.5h for a 15 min audio track (maybe change Google account or retry), and then it might take 2 mins for a similar length once it uses GPU.

- jazzpear's MDX23C model ([files](#)) - rename the config to .yaml as UVR GUI doesn't read .yml. You put config in UVR's models\mdx_net_models\model_data\mdx_c_configs. Then when you use it in UVR it'll ask you for params, so you locate the newly placed config file.

- Aufr33 Mel-Former denoise average "27.9768" model - dedicated for footsteps, crunches, rustling, sound of cars, helicopters

If it's not available for paid users of [uvronline.app](#), use [this](#) link | MVSEP | model files:

[Less aggressive](#) & [More aggressive](#) | [yaml file](#) | UVR Roformer [patch](#) | [Colab](#)

- myxt.com (uses Audioshake)

- [AudioSep](#) (you can try it to get e.g. birds SFX and then use as a source to debleed or maybe try to invert phase and cancel out)

- Moises.AI (the rumour says it's better than Bandit v2, but it's expensive "Dialogue, Soundtrack, Effects")

- Older DNR model on MVSEP from '22

I think the most commonly used recent SFX models discussed in the server before DnR v3 ones are DNR Demucs 4 model and Bandit v2, but I haven't seen any settlement in the community on which model is the best, hence it might simply depend on a song.

- voc_ft - sometimes it can be better than Demucs DNR model (although still not perfect)
- [jazzpear94 model](#) (VR-arch) - put the .pth file to: Ultimate Vocal Remover\models\VR_Models. On UVR start set config: 1band sr44100 hl 1024, stem name: SFX, Do NOT check inverse stem in UVR5, 5.1 disabled. Or put that [file](#) to model_data subfolder.
- ([dl](#)) [source](#) by Forte (VR) (probably setting to: instrumental/1band_44100_hl1024 is the proper config) Might work in Colab "I tried it with the SFX models, and I just uploaded them in the models folder and then placed the model name, and it processed them" and may even work in UVR.
- Or [GSEP](#) (sometimes) esp. the new "Vocal Remover" model

Any other stem/instrument/sample if not listed above

- Zero Shot Audio Source Separation
- Bytedance-USS (might be worse for instruments, but better for SFX)
- Dango.ai custom stem separation (paid, free 10 seconds preview)
- [Audiosep](#) (separate everything you describe; Colab has unpickling issue)
- *Spectral removers (software or VST):*
Quick Quack MashTactic (VST), Peel (VST, they say it's worse alternative of MT), Bitwig (DAW), RipX (app), iZotope Iris (VST/app), SpectraLayers (app, "Problem with RX [Editor's spectral editing] is it doesn't support working in layers non-destructively."), R-Mix (old 32 bit 2010 Sonar plugin), free [ISSE](#) (app, [showcase](#)), [FactorSynth](#), Zplane Copycat "but MashTactic also has a dynamics parameter that is really useful (you can isolate attack from longer sounds, or the opposite, coupled with the stereo placement and EQ isolation)" RipX is "not as good as UVR5 for actual separation, but RipX is very good if you need to edit what's already separated more musically. SpectraLayers is a nicer spectral editor, RipX spectral editor is not as usable"

Consecutive multi-AI separation for not listed instruments

- Extract all other instruments "one by one" using other models in the chain (e.g. remove vocals with voc_ft or now e.g. inst Mel Kim derivative, use what's left to remove drums/bass with htdemucs_ft/MDX23/MVSEP ensemble, use what's left to remove guitars/piano with GSEP/demucs_6s or now any better purpose model, then use what's left to remove e.g. wind instruments (if present) with UVR wind model, or any any other purpose model applicable, even SFX, till you're left with the instrument of your choice, or as few instruments, as possible, for potentially easier work with spectral editor listed above)
- [Drumsep](#) - "Using DrumSep on melodic stems can help separate instruments easier if you plan on sampling/editing them, but they are separated based on range rather than actual instrument. Low instruments will often be on the kick/tom stems, mid instruments will be on snare and/or tom, and higher instruments will be on the cymbals."

De-reverb

- Mel-Roformer de-reverb by anvuew v2 (a.k.a. 19.1729 SDR) | [DL](#) | [config](#) | [Colab](#)

Probably the best de-reverb for now.

- and RX11's dialogue isolate for de-echo.

"anvuew's models can remove reverb effect only from vocals, "captures early reflections a little". Old FoxJoy's model works with full track."

"sort of reminds of RX11 dialogue dereverb results but doesn't destroy singing voices"

"perfect for rvc"

Both BS and Mel variant "will also remove harmonies or vocal effects that are not in the center channel."

"it works sort of like [that](#) phantom center model, removing sides basically"

Sometimes "noreverb" stem might get empty (e.g. on MVSEP, but similar issues was fixed already once there).

"reminds me of the equivalent dereverb mdx model (...) cleaner in some ways, though slightly more filtered and aggressive."

"it's EXTREMELY aggressive, like very aggressive, it seems kinda muddy at a lot of parts, almost NO reverb bleed, it also caught so many effects and removed them (good thing) which is actually insane!! i also noticed that when it gets breathy or like it has falsetto, it seems to remove a lot of it, it's very weird at the breathy-ish parts of it lol, will be using this mainly if there are heavy vocal effects i want removing" - isling

In fact, it was "fine-tuned from kim's mel" - anvuew.

To make it work with UVR, delete "linear_transformer_depth: 0" from the YAML file, copy the model to MDX_Net_Models and config to model_data\mdx_c_configs.

- Dango Reverb Remover - [click](#) ("it's very similar to [RX11] dialogue isolate good/real-time set to 5. Yeah, it's like listening to the same inference files" John; probably also works in mono, you can get 30 seconds for free) but for other purposes, even older FoxyJoy's models can give better results

- anvuew BS-Roformer Dereverb Room [model](#) | [Colab](#) | MVSEP

(doesn't work in UVR, use [MSST](#), If you have stereo errors using MSST on stereo files, update MSST [git clone and git pull commands] or see below.

"specifically for mono vocal room reverb." as most are recorded in mono.

Not that long inference compared to other Roformers.

"Really liking the fullness in the noreverb stem. Virtually all dereverb roformers I've tried sound muddy, but this one is just the opposite. (...) Other noises may interfere, and in my experience, makes the model underestimate the reverb. [The previous anvuew's mono model] is way different [from] this one in every way. So, like I say, worth a shot." - Musicalman

Do the below to fix stereo error using that model, it might work with your current MSST version instead of the linked repo too, but in a different line.

“Edit inference.py from my [repo](#) line 59:

Replace :

```
# Convert mono to stereo if needed
if len(mix.shape) == 1:
    mix = np.stack([mix, mix], axis=0)
```

by :

```
# If mono audio we must adjust it depending on model
if len(mix.shape) == 1:
    mix = np.expand_dims(mix, axis=0)
if 'num_channels' in config.audio:
    if config.audio['num_channels'] == 2:
        print(f'Convert mono track to stereo...')
        mix = np.concatenate([mix, mix], axis=0)"
```

- jarredou

- anvew [dereverb_mel_band_roformer_mono_anvew_sdr_20.4029](#) model | [yaml](#) | x-minus
“supports mono, but ability to remove bleed and BV is decreased” “separates reverb better than v2”

- Sacial Mel-Roformer dereverb/echo model #3 called “fused”: [model](#) | [yaml](#)
“more effective in removing large reverb”

“Specifically targeting large reverb removal. After training, I combined these two models with my v2 model through a blending process, to better handle all scenarios. At this stage, I am still unsure whether my new models outperform the anvew’s v2 model overall [besides large reverbs].” [More](#)

- avew v2 “less aggressive” variant - a bit lower SDR 18.81 | [DL](#) | [config](#)

- Gabox Lead Vocal Mel-Roformer de-reverb | [DL](#) | [config](#) | [Colab](#)
“just use it on the mixture”

Older

VR [models](#)

(added to UVR 5 (GUI), works in NotEddy’s [Colab](#))

- UVR-DeEcho-DeReverb (213 MB) - “it removes reverb not echo” but you could try it if everything above fails

“use an aggression of 3.0 -5.0 and nothing more than that.” e.g. 4 (0.4 in some CLI code Colabs).

“Results have a frequency ceiling around 17540 Hz and a very high pitched noise above

22000 Hz, you might want to upscale your results with HQNizer or Apollo Model" - [more](#). Or use [point #4](#) to slow down the audio and revert it back.

(below the old ones, which might only work with vocal-remover 5.0.2 by tsurumeso's default arch settings, [maybe 1band_sr44100_hi1024 or [512?](#) and his [nets and layers](#)])

- VR dereverb - only works on tracks with stereo reverb (j48qny.pth, 56,5MB) ([dl](#)) ([source](#))
- VR reverb and echo removal model (j48qny.pth, 56,5MB) ([dl](#)), works with mono/stereo)

MDX models (less aggressive than those at the top)

"I use it when there's not so much reverb, but if it's more intense I will choose VR-Arch DeEcho"

> FoxyJoy's dereverb V2 - works only with stereo (available in UVR's download center and Colab (eventually via [this](#) dl link); it can spoil singing in acapellas or sometimes removes delay too). "I do think [that] MDX is noticeably more accurate [vs VR DeEcho-DeReverb]"

"(the model is also on X-minus) Note that this model works differently from the UVR GUI. I use the rate change (but unlike Soprano mode only by 1 semitone). This extends the frequency response and shifts the MDX noise to a higher frequency range." It's 11/12 of the speed so x 0.917, but actually something else goes on here:

(Anjok)

"The input audio is stretched to 106%, and lowered by 1 semitone using resampling. After AI processing, the speed and pitch of the result are restored."

You'll find slowing down method explained further in "tips to enhance separation"

"De-echo is superior to de-reverb in every way in my experience"

"VR DeEcho DeReverb model removes both echo and reverb and can also remove mono reverb while MDX reverb model can only remove stereo reverb"

"You have to switch [main stem/pair] to other/no other instead of vocal/inst" in order to ensemble de-echo and de-reverb models.

Newer

- UVR Dereverb model by Aufr33 & jarredou for uvronline.app premium | [Model files](#) | [settings](#) (PS: Dry, Bal: 0, VR 5.1, Out: 32/128, Param: 4band_v4_ms_fulband)
Copy model file to Ultimate Vocal Remover\models\VR_Models and json config is probably already present lib_v5\vr_network\modelparams (and has the same checksum).

- MDX23C UVR Dereverb model for uvronline.app premium | [Model files](#) | [config](#)
(by Aufr33 & jarredou)

Seems to pick up room reverb. Previous Foxy's model sometimes cut "way too much" than this model.

Copy model file to Ultimate Vocal Remover\models\MDX_Net_Models and yaml config to \model_data\mdx_c_configs subfolder.

Bas Curtiz' conclusion on both:

- MDX23C "seems to be cleaner, takes the reverb away, also between the words, whereas (U)VR leaves a little reverb"
- VR "seems to sound more natural, maybe therefore actually."
- MDX23C tends to 'pinch' some stuff away to the background, which sounds unnatural.

"This is just based on my experience with 3 songs/comparisons, but both points are a pattern.

Overall, they're both great when u compare them against the original reverbed/untouched vocals." showcase [video](#).

- You can find older avuew Mel versions [here](#)

- BS-Roformer anvew variant (a.k.a. 8/256/8) | [DL](#) - a bit higher SDR than Mel v1 (8/26/6) posted firstly in ZFTurbo repo.

"not good is all people say" but it might depend on a use case or a song.

Mel-Roformer might turn out to be better more often "works weirdly and leaves some echo for some reason" "very usable for single singing voices and speech, cus it's very precise in eliminating echo and reverb,

but if you have a choir singing or vocals with backing vocals in it, then it'll probably ruin it a bit. For such vocals it's better to use aufr33/jarredou model or dereverb deecho" John UVR

To fix issues with BS variant of the model in UVR "change stft_hop_length: 512 to stft_hop_length: 441 so it matches the hop_length above" in the yaml file (thx lew), plus delete linear_transformer line in the config like above too.

- 8 384 dim 10 depth BS variant | [dl](#) | [config](#)

- #2 Sacial Mel-Roformer dereverb/echo model ([model](#) | MVSEP).

Fine-tune with more training data.

- #1 Sacial Mel-Roformer dereverb/echo model ([model](#) | MVSEP).

It's good but doesn't seem to be better than the anvew's Mel v2 model above ([models list](#)).

Still, might depend on a use case.

- older V1 de-reverb HQ MDX model by FoxyJoy ([dl](#)) ([source](#)) (also decent results, but most likely worse).

("It uses the default older architecture with the fft size of 6144"

"After separation, UVR cuts off the frequencies at 15 kHz, so I found that to fix that is to invert the "Vocals" and mix that with the original audio file."

Demonstration: [Original](#) | [Dereverbed](#) | [Detected reverb](#))

- To enhance the result if necessary, you can use more layers of models to dereverb vocals, e.g.:

Demucs + karaoke model + De-reverb HQ (by FoxyJoy)

"works wonders on some of this stuff".

"Originally I inverted with instrumentals then I ran through deecho dereverb at 10 aggression then demucs_ft then kim vocal 2 then uvr 6_ at 10 aggression and finally deecho normal" (isling)

- For room reverb check out:

Reverb HQ

then

De-echo models (J2)

"from my experience, De-Reverb HQ specifically only really works when the sound is panned in the center of the stereo field perfectly with no phase differences or effects or anything that could cause the sound to be out of phase in certain frequencies.

If the sound doesn't fit that criteria, it only accurately produces the output of whatever's in the mid"

"I noticed that in some cases the DeEcho normal worked better than the aggressive, which was weird. That's why I ran through both, so to remove as much as possible."

- For removing reverb bleed left over in the left and right channels of a 5.1 mix from TV shows/movies check out:

Melband Roformer on MVSEP

Free apps/VSTs for de-reverb/de-echo/denoise

- Accusonus ERA (was good, but discontinued when Facebook bought them, can be found on archive.org from when they gaveaway it without DRM)

- [Voicefixer](#) (CML, only for voice, [online](#))

- [RemFX](#) (de: chorus, delay, distortion, dynamic range compression, and reverb or custom)

- [Noise Suppression for Voice](#) (a.k.a. RNNNoise, worse, various plugin types, available in OBS; now also RNNNoise 0.2/1.10 available)

- [Krisp](#) app (paid, free 60 minutes per day) better (same for RTX voice) - free on Discord

- [Adobe Podcast](#) (online, a.k.a. Adobe Podcast Enhance Speech, only for narration, changes the tone of voice, so you might want to use only frequencies from it above 16kHz)

- [AI-Coustics](#) (speech enhancement, 30 minutes/5 files for free per month)

- [CrystalSound.AI](#) (app)

- [Noise Blocker](#) (paid, 60 minutes free per day)

- [Steelseries GG](#) (app, classic noise gate with EQ and optional paid AI module, activating by voice in noisy environment may not always work correctly)

- RTX Voice (in NVIDIA Broadcast app, currently for any GTX or RTX GPU)

- AMD Noise Suppression (for RX 6000 series cards, or for older ones using unofficial Amernime Drivers)
- Elgato Wave Link 3.0 - Voice Focus feature (now free for everyone, standalone VST3/AU version is paid 50\$)
- [AI SWB Noise Suppression](#) (free, currently they give away that Mac/Windows driver only on [email](#) requests)
- Audio Magic Eraser shipped with new Google Pixel phones (separate [options](#) for cancellation of: noise, wind, crowd, speech, music)

The best paid de-reverb plugins for vocal tracks/stems/separations:

- RX 11 Dialogue Isolate (RX Editor/VST, paid) - some people like it more than DeVerberate 3. In RX Advanced variant, there's additionally "multi-band processing and a high-quality mode as an offline process", good companion for de-echo along with anvew v2 model for dereverb
- DeVerberate 3 by Acon Digital (someone while comparing said it might be even better than RX10) "I find it's useful to take the reverb only track and unreverbed track and mix them to a nice level" "Acon is probably best if you can tweak to each stem separated. RX is imo too rough." [Comparison](#)
- Accentize DeRoom Pro ("great" but expensive, available in DxRevive Pro, now 1.1.0)
- prime:vocal (multitool with also dereverb and other vocal enhancers)
- DxRevive Pro 1.1.0 - complete dialogue restoration tool; noise removal, reverb suppression, restoration of absent frequencies, elimination of Codec Artifacts
- Izotope RX <?8-10 Dialogue De-Reverb (RX Editor/VST) for voice and mixtures (more possible free [solutions](#)). Good results not only for room reflections, but also regular reverb in vocals. It picks reverb where even FoxyJoy's model fails ("De-reverb" and "Dialogue de-reverb" options). It's destructive for mixing raw vocals, but can just work.
- Clear by Supertone "equally good compared to RX10 imho. Smoother imho. It's only good on vocals though" Simple 3 knob plugin - "the cleverest / least-manual to get good results and is AI-based." previously known as Supertone Voice Clarity and defunct free GOYO.AI) Also destructive for mixing raw vocals, but can just work.
- Waves Clarity Vx DeReverb - it cannot perform de-echoing, so you need UVR De-echo (17.7kHz cutoff) or RX Dialogue Isolate for it, simpler than RX (paid; models updated in 12/17/2023 build) - same. Maybe you could even mix the two plugins using less aggressive settings in both.

Others:

- SPL De-Verb Plus
- Audio Damage Deverb
- Zynaptiq UnVeil
- Zynaptiq Intensity
- Thimeo Stereo Tool (one of its modules)
- Acon Dialogue:Extract 2 (dereverb, denoise)

If you want to use some of these DAW plugins for your microphone in real-time, you can use Equalizer APO.

Go to "Recording devices" -> "Recording" -> "Properties" of the target mic -> "Advanced".

To enable a plugin in Equalizer APO select "Plugins" -> "VST Plugin" and specify the plugin dll. AFAIK, VST3 is unsupported.

To run a plugin for a microphone in a simple app and send it to any output device, alternatively, you can download [savihost3x64](#), then edit downloaded exe name to the name of your plugin you want to use, placed nearby, and run the app. Now go to settings and set input and output device (can be virtual card, maybe not necessarily). Contrary to Equalizer APO (irc) it supports VST3 plugins too. Of course, you can also use DAWs for the same purpose (Reaper, Cakewalk etc. - but not Audacity irc)

AI (paid):

<https://twoshot.app/model/36>

Free:

<https://github.com/FORARTfe/HyMPS/blob/main/Audio/AI-Enhancing.md#dereverbers->

De-echo

- UVR-De-Echo-Aggressive (121 MB)
- UVR-De-Echo-Normal (121 MB)
- UVR-DeEcho-DeReverb (213 MB)

(now added in UVR and MVSEP, won't be in Colab for now, but the first two are on [HuggingFace](#))

- [delay_v2_nf2048_hl512.pth](#) (by FoxyJoy, all VR arch, [source](#), can't remember if it was one of the above), decent results.

"works in UVR 5 too. Just need to select the 1band_sr44100_hl512.json when the GUI asks for the parameters"

"You [also] can use this command to run it: python inference.py -P
models\delay_v2_nf2048_hl512.pth --n_fft 2048 --hop_length 512 --input audio.wav --tta
--gpu 0"

They're also on X-Minus now:

"The "minimum" and "average" aggressiveness settings use the Normal version of the model. The Aggressive one is used only at the "maximum" aggressiveness."

"What's crazy is maximum aggressiveness sometimes does better at removing bgvox than actual karaoke models"

Denoising (vinyl noise/white noise/general)

- Denoise standard in UVR for MDX noise (like in HQ_1-5, iirc it uses HV [code](#); explanation how it works: it separates "twice, with the second try inverted, after separation reinverted, to

amplify the result, but remove the noise introduced by MDX, and then deamplified by 6dB, so it still the same volume, just without MDX noise.”

- Denoise model in UVR (it's using VR's UVR-DeNoise-Lite, 20kHz cutoff)

(Options>Choose Advanced Menu>Advanced MDX-Net Options>Denoise output)

for filtering noise existing in almost all MDX-Net models in silent or quiet parts, but potentially also for more applications

- Min Spec ensemble of *denoise model* and *denoise disabled* results in Advanced MDX-Net Options (Audio Tools>Manual ensemble>Min Spec)

Filters more MDX noise in quieter parts than denoise standard and denoise model.

- Mel-Roformer Denoise by Aufr33 | [Colab](#) | MVSEP | links below

a) minimum aggressiveness model called “27.9959” a.k.a. “1”

- good for white noise/static noise

b) average “27.9768” a.k.a. “2” model

- works for footsteps, crunches, rustling, sound of cars, helicopters

Some people like to use overlap 10 with these models.

minimum - removes fewer effects such as thunder rolls, scratching or sweeping surfaces. It's not as good at removing louder MDX noise when using AMD GPU instead of CPU on older system's DirectML.dll in UVR

average a.k.a. aggressive - usually removes more noise than minimum, and also occasionally slight reverb/echo from room in vocals

“The Mel-RoFormer denoise model is amazing at removing 78 RPM record crackle”

It's much better at higher frequencies than the model below (it doesn't damage them that bad).

Incredibly useful as mixing tools, can pull all kinds of hum out of raw vocals, guitars, room mic's, bass, etc before mixing with zero artifacts left over. I would absolutely love to see where he takes these.

If it's not available for paid users of [uvronline.app](#), use [this](#) link | model files:

[Less aggressive](#) & [More aggressive](#) | [yaml file](#) | works with UVR Roformer [patch](#) | [MSST](#)

For UVR - use Install model option in MDX-Net, or copy ckpt files to models\MDX-Net folder and yaml to model_data\mdx_c_configs subfolder. Choose the new model, press yes to set parameters, enable Roformer option, pick the config file corresponding with the copied yaml name. In case of “use_amp” error (e.g. in MSST), add “use_amp: true” in the yaml under optimizer and other_fix lines.

“From most aggressive to least:

VR Denoise

VR Denoise Lite

Mel-Rofo Denoise Aggr(essive)

Mel-Rofo Denoise”

- Bas Curtiz

Rather RX11 Spectral Denoise can be more aggressive than all of them at certain settings.

- Gabox [denoise/debleed](#) model | [yaml](#) | [Colab](#) - for noise from fullness models (tested on v5n) - it can't remove the vocal residues - try out denoising on mixture first, then use fullness model.

"It can preserve slightly more high frequency content in speech [than Aufr33 Mel model]" - Musicalman. "quite a bit slower". Impressive "ability to clean up noisy vinyl or cassettes" (padybu), might work better than Aufr33's model (pipedream)

- Apollo Lew Uni model - tends to smooth out some even consistent noise in e.g. higher frequencies, making the spectrum more even there

- UVR De-Noise by aufr33 "minimum aggressiveness" on [x-minus.pro/uvronline.app](#) (for premium or using [this](#) link)

(less aggressive than denoise model in UVR,

"The (...) model is designed mainly to remove hiss, such as preamp noise. For vocals that have pops or clipping crackles or other audio irregularities, use the old denoise model".

Grabs "sound effects in old recordings (radio drama)", might make "soft voices sound weak").

- UVR De-Noise by aufr33 "medium aggressiveness" on x-minus (same as default for free users) - it seems to be even less aggressive than UVR-DeNoise-Lite in UVR

- Mel-Roformer De-Crowd by Aufr33/viperx ([x-minus.pro/UVR/DL/yaml](#))

("to remove background noise when denoise models were failing [not sure if it was rain or wind", can remove vinyl noises])

For UVR, change the model name to the one from the attached yaml, copy chkpt to models\MDX_Net_Models, and yaml to model_data subfolder, then set overlap 2 or use ZFTurbo inference [script](#)] - more effective than MDX below at times)

- yxllic's harmonic noise separation VR [model](#) (v. 6 or 5.x; unsure)

Vocal models as denoisers

- Unwa BigBeta 5e and 6 (5e "good when your mic/pc makes a lot of noise. All the denoise models are a bit too harsh for ASMR" - giliaan, both "for denoising a conversation, was better than: UVR Denoise, MDX23cInstVocHQ, HQ5, KimVocal2, VocFT, Apollo, MelRof-aufr33-Denoise, GaboxDenoise, BanditV2cinematic, ViperX-BSrof-1297" - mixamillion) | [Colab](#) | [MVSEP](#) | [Colab](#) | [MSST-GUI](#) | [UVR instruction](#) | [Model](#) | yaml: big_beta5e.yaml | [fixed](#) yaml for AttributeError in UVR

- BS-Roformer *viperx 1296 / MVSEP* BS-Roformer 04.24 / *Gabox BS_ResurrectioN* (denoising and derumbling working the most efficiently on vocals here too)
x-minus/[Colab](#)/UVR/MVSEP

- Kim Mel-Roformer (works for denoising and debleeding vocals well)
 - Vocal model like Voc_FT or Unwa's ft2 bleedless ("it can sometimes isolate the vocals without the noise. And has a better result than a normal denoiser model" - Kashi)
 - Mel-Roformer Karaoke (by aufr33 & viperx) (to remove noise from a dialogue, mostly rustling in the background)
- [x-minus.pro](#) / [uvronline.app](#) / [mvsep model file](#) (UVR [instruction](#))
- Mel-Roformer Duality model (excellent for pops and clicks in mixture to get clean vocals out of 45 RPM vinyl mixture - bratmix)

Other tools

- [resemble-enhance](#) (available on x-minus, but only as denoiser for voice/vocals, and on [HuggingFace](#), [site](#); works good for wind/outside noise)
- <https://tape.it/denoiser> - ("great tool for removing tape hiss. Seems to be free without limitation at this point in time, though it seems to have issues with very large files [20 mins etc])"
- <https://crowdunmix.org/try-rokuon/>
- <https://github.com/eloimoliner/denoising-historical-recordings> (mono, old 78rpm vinyls, fixed [Colab](#), sometimes deletes SFX, but not as much as UVR De-Noise by aufr33 in old recordings)
- [audio.ai](#)
- <https://github.com/sp-uhh/avgen>
- <https://github.com/Rikorose/DeepFilterNet> | [Huggingface](#) (for speech)
- <https://studio.gaudiolab.io> (new Noise Reduction feature)
- possibly [USS-Bytedance](#) (when similar sample provided)
- [Various AI tools](#) - list by FORARTfe/HyMPS | [#2](#)
- [Free apps](#)

Older models

- [UVR-DeNoise](#) (trained by FoxJoy) - DeNoise-lite above is less aggressive
You can use negative values in UVR for that model. -20/-25 - for cleaning vocals
-10/-15 - when some vocals are gone - Gabox
"It's decent, but it needs a little work compared to" RX 10 spectral denoise.
- voc_ft - works as a good denoiser for old vocal recordings
- GSEP 4-6 stem ("noise reduction is too damn good. It's on by default, but it's the best I've heard every other noise reduction algorithm makes the overall sound mushier", it's also good when GSEP gives too noisy instrumentals with 2 stem option, it can even cancel some louder vocal residues completely)
- UVR-MDX-NET Crowd HQ 1 (UVR/x-minus)
- [This](#) VR ensemble in [Colab](#) (for breaking sounds, process your separation output more than once till you get there)

Plugins (different types of noise)

Free

- Guide for classic denoiser tools in DAW, e.g. for debleeding (Bas Curtiz):
<https://docs.google.com/spreadsheets/d/1XIbyHwzTrbs6LbShEO-MeC36Z2scu-7qjLb-NiVt09I/edit?usp=sharing>
- [Bertom Denoiser Classic](#) (or paid [Pro](#))
- [Accusonus ERA 6](#) (released for free after FB acquisition) - bundle with also de-esser, voice auto-EQ, voice leveller (better than soothe2 for de-essing for some people), deplosive, declipper and more

Paid

- Izotope RX 10 Spectral De-noise (“I think RX 10’s Spectral De-noise is better at removing the noise MDX [model] makes”)
Actually, the new UVR De-noise model is really good when you combine it with RX 10’s Spectral De Noise”, better than Lab 4 and current models, also more tweakable, but takes more time to set (now also RX 11 available - should be even a step forward)
- Acon Restoration Suite 2’s DeNoise “is decent if you can build a good noise profile with the Learn option, I like to have a few in series set to do -3dB of NR.” - theophilus3711
- SOUND FORGE Audio Cleaning Lab 4 (formerly Magix Audio & Music Lab Premium 22 [2016/2017] or MAGIX Video Sound Cleaning Lab - basically the same stock plugin across all of these versions)
- Unchirp VST (for musical noise, artefacts of lossy compression)
- Izotope Dialogue Dereverb (it is also denoiser)
- Izotope Dialogue Isolate in RX11
- Waves Clarity Vx / Pro (designed mainly for vocals)
- Brusfri by Klevgrand
- prime:vocal (multitool with also dereverb and other vocal enhancers)
- DxRevive Pro (mainly for dialogue: denoiser, declipper, dereverb, enhancer, codecs artefacts removal)
- Acon Dialogue:Extract 2 (dereverb, denoise)

Visit also [Debleeding/cleaning](#) e.g. inverts

Bird sounds

- [Google’s bird_mixit](#) (code & checkpoint for their bird sound separation algo. [More](#))
- De-reverb models, e.g.:
 - UVR-DeEcho-DeReverb (doesn’t work for all songs)
- Vocal models, e.g.:
 - MVSEP BS-Roformer 2025.07 (if you already have birds in a vocal stem, as most vocal models do iirc, that may do the trick)
- SFX models
 - Zero shot solutions:

(you can try them to get e.g. birds SFX and then use as a source to debleed or maybe try to invert phase and cancel it out)

- [AudioSep](#)
- [USS-ByteDance](#) (for providing any, at least, proper sample)
- [Zero Shot](#) (currently worse for SFX vs Bytedance)

- custom stem separation on Dango (paid, 10 seconds for free)

Technically, if bird noises are in vocals, then equally:

- RTX Voice,
- AMD Noise Suppression or even
- Krisp and
- Adobe Podcast

might get rid of them, but at least the last changes the tone of voice, and the previous may work good only with voice instead of vocals.

Spectral editing

De-clipping/de-limiter/de-compression of dynamics (for loud or brickwalled songs with overly used compressor/clipper/limiter/distortion - transients/peaks recovery)

Free declipper plugins:

- ReLife 1.42 by Terry West (works best for stereo tracks divided into mono, newer versions are paid)
- [ERA 6](#) declipper (released in bundle for free after they were bought by Meta)
- Airwindows AQuickVoiceClip - mainly for streamers yelling into the microphone “It’s not a ‘un-clipper’ but it tames the distortion a bit.”

AI tools (not plugins):

- [RemFX](#) (contains model to get rid of distortion and compression; “mostly for singular sounds, it won’t work for whole mixes like songs”)
- [Rukai](#) (for speech and instrumentals)
- [Amis](#) (mainly for speech)
- [stet-stet's DDD](#) (speech, req. decent CML knowledge to setup)
- [jeonchangbin49's De-limiter](#) (“if you have any squished tracks that apollo doesn’t handle well, try passing it through that AI de-limiter first” - macularguide
“parallel mix - “define[s] how the normalized input and the de-limited inference will blend together. Where 0 is 100% normalized and 1 means 100% de-limited” - santilli_)
- [Neutone FX>Clipper](#), actually an AI plugin ([instruction](#))

More GH repos - [HyMPS](#) list | #2

Paid: Ozone’s 12 Delimiter, ProAudioDeclipper, Declipper in Thimeo Stereo Tool (a.k.a. Perfect Declipper - standalone; both free for Winamp), iZotope RX De-clip (in RX Editor or as plugin), DxRevive Pro (mainly for dialogue, also denoiser, dereverb, enhancer, codecs artefacts removal), Declipper in Magix/Sound Forge Cleaning Lab, Adobe Audition’s

Declipper, sometimes even Fabfilter Pro-MB multiband compressor might be useful

Clippers (the opposite, but useful in the whole mastering chain, sometimes in a tandem with the above in the whole chain):

Free

- KClip Zero
- FreeClip (sometimes you can use both in the same session for interesting results)
- GClip
- Limiter6 by vladg (Clipper module)

- Initial Clipper
- Airwindows Hypersoft - “a more extreme form of soft-clipper”
- Airwindows OneCornerClip - compared to OG ADClip, it retains the character of sound
- Airwindows ADClip8 - “loudenator/biggenator”
- Airwindows ClipOnly - “2-buss safety clipper at -0.2dB with powerful anti-glare processing.”
- Hornet Magnus Lite - clipper and limiter modules

Paid: Orange Clip 3 (multiband mode), Gold Clip (widely praised lately), Gold Clip Track, Soundtheory Kraftur, KClip 3, SIR Standard Clip (popular, though KClip 3 may give better results), Izotope Trash 2, DMG Tracklimit, TR5 Classic Clipper (great for a kick), KNOCK (hard & soft clipper), Boz Little Clipper 2, Flatline (clipper), Newfangled/Eventide Saturate (spectral clipper), JST Clip, Brainworx Clipper, Elysia Alpha Mastering Compressor (soft clip module), soft clipper in Cubase

De-expliciter (removes explicit lyrics from songs)

<https://github.com/tejasramdas/CleanBeats> (more recent fork)

De-breath

- Sacial de-breath VR v1/2 [models](#)
- Accusonus ERA Bundle (free/gave away after FB acquisition) [download](#)
- Izotope RX11 breath control (paid; VST/Audio Editor)
("The “remove breaths” preset they have on it Usually works about 95% of the time for me”
-5b)
- DNR v3 (Sometimes ...) (without vocal help), the grunts and breathing will be in the SFX, and the dialogue in the speech, while both will be in the music) - fal_2067

Manipulate various [MDX settings](#) and [VR Settings](#) to get better results

Final resort - specific [tips to enhance separation](#) if you still fail in certain fragments or tracks

Get VIP models in UVR5 GUI (optional donation) - it's if you can't find some of the listed above or in top ensembles chart:

<https://www.buymeacoffee.com/uvr5/vip-model-download-instructions>

(dead links)

List of VR models in UVR5 when VIP code is entered (w/o two denoise by FoxyJoy yet):

<https://cdn.discordapp.com/attachments/708595418400817162/1104424304927592568/VR-Arch.png>

List of MDX models when VIP Code is entered (w/o HQ_3 and voc_ft yet and MDX23C):

<https://cdn.discordapp.com/attachments/708595418400817162/1103830880839008296/AO5jKyQ.png>

More updated list can be found in that UI:

https://huggingface.co/spaces/TheStinger/UVR5_UI

(some models might be not from Download Center/VIP code)

Models repository backup of all UVR5 models in separate links

https://github.com/TRvlvr/model_repo/releases/tag/all_public_uvr_models

Some models might be not available in the repository above, as e.g. 427 model which is available only after entering VIP code.

(just in case, here's the link for 427:

https://drive.google.com/drive/folders/16sEox9Z_rGTngFUtJceQ63O5S9hhjjDk?usp=drive_link

Copy it to UVR folder\models~MDX folder and rename the model name to:

UVR-MDX-NET_Main_427)

Q: "Hello, we are now getting very good results in turning music that includes human voice into only instrumental. Sometimes there are vocal leaks that we can call just crumbs or whispers, but this is not that important. But now we have another important problem. People who do not want to listen to vocals, that is, who only want to listen to the music that remains when the vocals are deleted, encounter a problem. Sometimes there are big gaps in the songs. Because not every song is arranged in a way that continuous instrumental music is heard, and when the vocal part is deleted, a perception of silence or emptiness can occur. It is as if the music does not have continuity, and everything is cut off in some parts of the song. The reason for this is that when the vocal is deleted, the vocal melody is also destroyed. Although it seems like a good idea at first, when we listen to music that is only instrumental with the vocals deleted, that song loses a lot of its identity. As a result, I want to learn how we can preserve the vocal melody after deleting the vocal. What I mean is, can we divide the song into instrumental and vocal and then turn the melody of the vocal part into an

instrument such as piano, bass guitar, flute, etc. Then, I want to combine this vocal melody with the instrumental result." - sweetlittlebrownbat

A: You could try out some older, less aggressive models than Roformers. Even GSEP. They can sometimes leave some melody from vocals (in fact, some quiet harmonies), so the song is not so "dead" after separation. Actually, you could try to separate vocals into separate stems to look for something useful to mix with the instrumental quietly. Check Vocal models, then separate further with BV/Karaoke models or alternatively check GSEP, MDX-Net and maybe even VR models. Open document outline of this document and there you have all the interesting sections.

Also, you can use:

"<https://audimée.com/>

split instrumental from vocal

use vocal as input

convert it into piano, bass, flute, whatever they offer

merge

profit" Bas Curtiz

Mixing/mastering

If you already did your best in separating your track, tried out [ensembles](#) or [manual weighting](#), also read [tips to enhance separation](#), but if it lacks original track clarity, you can use:

- Demudder added in the beta [Roformer patch](#) #14 in UVR (if it won't increase vocal residues too much; won't work with even small chuk_size in AMD/Intel 4GB VRAM GPUs with Roformers)
- [AI Mastering services](#) (mainly for instrumentals)
- For improving vocals' clarity, you could even "train a RVC model out of clean [artists] audio clips and then inference this audio with the model you made. It takes some time, but the results are worth it" John UVR ([examples](#)). Workflow explained later below.
- Aufr33's expander template for Reaper 7.05 ([DL](#)) fixing ducking in instrumentals (explained later below)

Mixing track from scratch using various AIs/models

Now if you're not afraid of mixing, and e.g. if you have clear instrumental already or whole track to remaster, I used for such a task:

- v. quiet mixture (original file with mixed vocals)
- stems from demucs_ft or BS-Roformer SW (both [MDX23](#) Colab or Ensemble of various models on MVSEP can be even better than Demucs, and vs SW esp. for bass - check out [4 stems](#) section) mixed with also:

- [drumsep](#) MDX23C free model result (but you can also test out stems from the old drumsep and LarsNet [although they have worse SDR], or newer MVSEP drumsep models)
- GSEP result for piano or guitars (MVSEP models can be handy too, now the SW model for those stems are much better, and previously the only downloadable decent guitar model released recently, and demucs_6s is mediocre, now we have SW)
- for bass both GSEP and Demucs ft/MDX23 aligned and mixed together (or simply from MVSEP ensemble or MDX23 Colab) or see [bass models](#) for more recent list
- I think "other" stem could have been paired that way too (but drums remained only from e.g. Demucs_ft - they were cleaner than GSEP and good enough)
- Actually in one of those guitars weren't recognized in guitar stem, but were in other stem, so I mixed that all together (it wasn't busy mix)
- If it's not instrumental, probably mixing more than one vocal model might do the job, check various [vocal ensembles](#) (but it's essentially what MDX23 and ensembles on MVSEP do, but the latter with private models, it's not exactly the same - you can add different effects for every of such tracks, having fuller sound and change their volume manually).

The all above gave me an opportunity for a very clean mix and instruments using various plugins while setting correct volume proportions vs mastering just instrumental separation result or plain 3 stems from Demucs.

For example, demucs_ft or other single or incorporated drums model provides much higher quality of drums than the old Demucs drumsep during mixing, so in such case you won't use its stems on its own, but you will use drumsep more to overdub the specific parts of instruments more (e.g. snares - that's the most useful part of using drumsep as normally it's easy to bury snare in a busy mix when hi hats kick in overly in a heavily processed instrumental stem or drums stem - not you won't have to push drums stems from demucs_ft or MDX23 so drastically).

Sam Hocking's method for enhancing separated instrumentals from a mixture (song containing instrumental and vocals):

"I think looking at spectrally significant things like snares can work. We can already do it manually by isolating the transient audio/snare pattern as MIDI and then triggering a sample from the track itself to reinforce, but it's time-consuming and requires a lot of sound engineering to make it sound invisible."

You can probably use Cableguys Snapback plugin for that, or maybe UVI Drum replacer. Sam's method will work the best in songs with samples instead of live recordings (if the same sounds repeat across the whole beat). [More](#) of those plugins.

PS. In the late 2025 we received an info about Apple Music rejecting Atmos mixes of some legacy music made with separation models (ensembles, and then probably some for 4-6 stems), even though the mixes sounded good, and were accepted by labels and artists. Also, we know that these separation methods worked fine in the past, at least for some other engineers. We suspect that they might use automated tools catching specific artefacts

usually seen in separation models on spectrograms of extracted channels from the whole Atmos mix.

"I don't think there's a need of really advanced and expensive method to detect source separated stems, most of the time, just looking at the background noise is enough to tell, original stem vs separated one [\[click\]](#)

+ kind of "aliasing" artifacts and/or dither residues popping here and there...

There are lots of patterns than can make separated audio stems identifiable, I don't think it's hard to develop a model to spot them with quite good accuracy (even if not really audible to human ears)" - jarredou

To sum up:

Tips to [enhance separation](#)

[Demudder](#) in UVR/x-minus
(increases vocal residues)

AI audio upscalers [list](#)

AI mastering [services](#)

[Blending with RVC model](#)

[Make your own remaster:](#)

More clarity/better quality/general audio restoration of separated stem(s)

Have complete freedom over the result, using (among others) spectral restoration plugins to demud the results of separations freely with plugins. Then you can use the result further with e.g. AI upscaler or in reverse.

E.g. from plugins, you can start by using [Thimeo Stereo Tool](#) which has a fantastic re/mastering chain feasible for spectral restoration useful for instrumentals sounding too filtered from vocals and lacking clarity. Also use [Unchirp](#) which states great complement to Thimeo Stereo Tool, although focuses more on already existing spectrum.

You can also play with free Airwindows [Energy/Energy2](#) and [Air/Air2](#) (or [Air3](#), [MIA Thin](#)) plugins for restoration, and furthermore some compressors or other plugins and effects mentioned in the link above.

If you're not afraid of learning a new DAW, [Sound Forge Cleaning Lab 4](#) has great and easy built-in restoration plugins too (Brilliance, Sound Clone>Brighten Internet Sources) with complete mastering chain to push even further what you already got with Unchirp and Stereo Tool.

Izotope RX Editor and its Spectral Recovery may turn out to be just not enough, but the rest of RX plugins also available as VST can become handy, although Cleaning Lab has lots of substitutes for filtering various kinds of noise. Working comfortably in real-time with all the

plugins opened simultaneously while combined is more comfortable than RX Editor workflow. But you can use some plugins from RX Editor as separate VSTs in other DAWs including Lab 4. Ozone Advanced might turn out useful too.

Actually, once you finish using the plugins above, now you can try out some of the mastering services and not in the opposite way (although you might want to meet some basic requirements of AI mastering services to get the best results first, e.g. in terms of volume).

Q: AI vocal remover did not "normalize" (I don't think it's the right word) the track on the moment where the vocal was removed, so it's noticeable, especially on instrument-heavy moments.

I make things better by created backup echo track by combining stereo tracks with inverted ones and adding this to the main track with -5db, but it's still not good enough. Are there any technics that separate track with not noticeable effects or maybe there is some good restoration algorithm that I can use

A: If vocals are cancelled by AI, such a moment stands out from the instrumental parts of the song.

Sometimes you can rearrange your track in a way that it will use instrumental parts of the song when there are no vocals, instead of leaving AI separated fragments. Sometimes it's not possible, because it will lack some fragments (then you can use only filtered moments at times), and even then, you will need to take care about coherence of the final result in the matter of sound as you said.

At times, even fade outs at the ends of tracks can have decent amounts of instrumentals which you can normalize and then use in rearrangement of the track. E.g. you normalize every snare or kick and everything later in fade out, and then till the end, so it will sound completely clean.

Generally it's all time-consuming, not always possible, and then you really have to be creative using normal mastering chain to fit filtered fragments to regular unfiltered fragments of the track.

You can also try out layering, e.g. specific snare found in a good quality in the track. May work easier for tracks made with quantization, so when the pattern of drums is consistent throughout the track. Also, you can use 4 stem Demucs ft or MDX23 and overlap drums from a fragment where you don't hear vocals yet, so drums are still crispy there.

Ducking effect eliminator

You can also check Aufr33 Reaper 7.05 project aimed at alleviating this issue:

"the music volume is reduced where there are vocals". Instruction:

"Just place two stems: vocals and music. Adjust the Expander if necessary."

"It's just an expander side-chained to vocals. You can replicate this in any other DAW."

[Src](#) | [mirror](#)

- Nice [chart](#) (>moved to “Advanced chain processing chart” at the bottom of Karaoke section (use search)
describing process for creating AI cover (replace kim vocal with voc ft there, or MDX23 vocals/UVR top ensemble/Roformers).

Blending with RVC model

(by Gabox & dubpluris a.k.a. Mark | Avalaunch - text)

“My use Case:

Restoring older, lower-quality vocal recordings (e.g., camcorder recordings from the 90s) using RVC models trained on clean studio vocals from the same artist.

Practical Workflow for Using RVC in Vocal Restoration

The idea is not to replace old performances entirely, but to enhance them. A few key points came out of the discussion:

- Blending, not replacing: Using only the RVC output will usually sound artificial. The better approach is to run the old vocal stems through the trained RVC model and then blend the AI-generated stem with the original. This preserves natural performance qualities while adding clarity.
- Input quality matters: Even “decent but rough” camcorder audio can work. Extremely degraded sources, however, will still produce artifacts (“bad input = bad output”).

Complementary tools:

FlashSR – an audio super-resolution method that restores high frequencies and improves fidelity before running RVC. (https://mvsep.com/en/demo?algorithm_id=60)

[AudioSR might potentially give better results, but it’s much slower;

“Imo much better candidates are: [AP-BWE \(Colab | new repo \[old\]\)](#) and [Clearer-Voice-Studio's Clear Voice](#) (my favorite is the 2nd one - codename0; more simplified version by codename0 - [DL](#)”]

Matchering – matches EQ/tonal balance of rough recordings to studio references, either standalone or integrated into UVR5. Using a clean studio version of the artist as the reference and the old performance as the target is recommended.

(<https://sergree.github.io/matchering/>) - Available on UVR

[You can also try out <https://masterknecht.klangknecht.com/>]

General workflow:

1. (Optional) Pre-process low-quality audio with FlashSR.
2. Train RVC on clean studio stems.

3. Run inference on the old stems with the trained model (i.e., feed the cleaned original vocal through the trained RVC model to get a converted stem.)
4. Blend, align and mix original + RVC stem (RVC as enhancement, not replacement) until it feels natural.
5. Use Matchering or other mastering techniques to polish.

For comprehensive remastering workflow, see [How to make your own remaster](#).

The overall takeaway: RVC can be used for restoration, but it works best as part of a chain of tools (super-resolution, EQ matching, mastering), with the human performance always kept at the center through blending rather than full replacement.”

**More descriptions of models
and AIs, with troubleshooting and tips**
(most models here are dated as it lacks Roformers)

(Instruction moved to [Reading advice](#))

Older models descriptions

- Inst fullband (fb) HQ_3/4/5 x-minus, MVSEP, Colabs

HQ_4 vs 3 has some problems with fadeouts when occasionally it can leave some vocal residues

HQ_3 generally has problems with strings. mdx_extra from Demucs 3/4 had better result with strings here, sometimes 6s model can be good compensation in ensemble for these lost instruments, but HQ_3 gives some extra details compared to those.

HQ_3/4 are generally muddy models at times, but with not much of vocal residues (near Gsep at times, but more than BS-Roformer v2).

For more clarity, use MDX23C HQ model (HQ_2 can have less vocal residues at times).

Another possibly problematic instruments are those wind ones (flute, trumpet etc.)

- use Kim inst or inst 3 then

HQ3 has worse SDR vs:

- voc_ft, but given that HQ_3 is an instrumental model, the latter can leave less vocal residues at times.

https://mvsep.com/quality_checker/leaderboard2.php?id=4029

https://mvsep.com/quality_checker/leaderboard2.php?id=3710

These are SDR results from the same patch, so the voc_ft vs HQ_3 comparison is valid.

- MDX23C_D1581 (narrowband) - usually worse results than voc_ft and probably worse SDR if evaluation for both models was made on the same patch

Can be a bit better for instrumentals

“The new model is very promising

although having noise, seems to pick vocals more accurately and the instrumentals don't have that much of the filtering effect (where entire frequencies are being muted).”

While others say it's worse than demucs_ft

- [GSEP AI](#) an online closed source service (cannot be installed on your computer or your own site). mp3 only, 20kHz cutoff.

Decent results in some cases, click on the link above to read more about GSEP in the specific section below. This [SDR leaderboard](#) underestimates it very much, probably due to some kind of post-processing used in GSEP [probably noise gate and/or slight reverb or chunking]. As a last resort, you can use 4-6 stems option and perform mixdown without vocal stem in e.g. Audacity or other DAW. 4-6 stem option has additional noise cancellation vs 2 stem.

GSEP is good with some tracks with not busy mix or acoustic songs where everything else simply fails, or you're forced to use the RX10 De-bleed feature.

- GSEP is also better than MDX-UVR instrumental models on at least tracks with **flute** and possibly duduk/clarinet or oriental tracks, and possibly tracks with only piano, as it has a decent dedicated piano model.

- To address the issue with flute using MDX-UVR, use the following ensemble: Kim_Inst, HQ1, HQ2, INST 3, Max Spec/Max Spec (Anjok).

- Sometimes kim inst and inst3 models are less vulnerable to the issue (not in all cases).

- Also, main 406 vocal model keeps most of these trumpets/saxes or other similar instruments

- Passing through a Karaoke model may help a bit with this issue (Mateus Contini [method](#)).

- inst HQ_1 (450)/HQ_2 (498)/HQ_3 MDX-UVR fullband models in Download center of UVR5 - great high quality models to use in most cases. The latter a bit better SDR, possibly a bit less vocal residues. Not so few like inst3 or kim ft other in specific cases, but a good point to start.

What you need to know about MDX-UVR models is that they're divided into instrumental and vocal models and that instrumental models will always leave some instrumental residues in vocals and vice versa - vocal models will more likely to leave some vocal residues in instrumentals. But you can still encounter specific cases of songs when breaking that rule will benefit you - that might depend on the specific song. Usually, instrumental model should give better instrumental if you're fighting with vocal residues.

Also, MDX-UVR models can sometimes pick up sound midi effects which won't be recovered.

- kim inst (a.k.a. ft other) - cutoff, cleaner results and better SDR than inst3/464 but tends to be more noisy than inst3 at times. Use:
- inst3/464 - to get more muddy, but less noisy results, although it all depends on a song, and sometimes HQ_1/2/3 models provides generally less vocal residues (or more detestable).
- MDX23 by ZFTurbo v1 - the third place in the newest MDX challenge. 4 stem. Already much better SDR than Demucs ft (4) model. More vocal residues than e.g. HQ_2 or Kim inst, but very clean results, if not the cleanest among all at the time. Jarredou in his fork fixed lots of those issues and further enhanced the SDR so it's comparable with Ensemble on MVSEP, which was also further enhanced since the first version of the code released in 2023, and also has newer models and various enhancements.
- [Demucs 4](#) (especially ft 4 stem model; UVR5, Colab, MVSEP, 6s available) - Demucs models don't have so aggressive noise cancellation and missing instruments issue like in GSEP. Check it out too in some cases (but it tends to have more vocal bleeding than GSEP and MDX-UVR inst3/464 and HQ_3 (not always, though), and 6 stem has more bleeding than 4 stem, but not so much like the old mdx_extra 4 stem model).
- [Models ensemble](#) in UVR5 GUI (**one of the best results** so far for both instrumentals and vocals SDR-wise). Decent Nvidia GPU required, or brace for 4 hours processing on 2/4 Sandy Bridge per whole ensemble of one song. How to set up ensemble [video](#). General video [guide](#) about UVR5.

"UVR-MDX still struggles with acoustic songs (with a lot of pianos, guitars, soft drums etc.)" so in this case use e.g. GSEP instead.

Description of vocal models by Erosunica

"That's my list of useful MDX-NET models (vocal primary), best to worst:

- MDX23C-8KFFT-InstVoc_HQ (Attenuates some non-verbal vocalizations: short low-level and/or high-frequency sounds)
 - Kim Vocal 2
 - UVR-MDX-NET-Voc_FT
 - Kim Vocal 1
 - Main (Attenuates some low level non-verbal vocalizations)
 - Main_340 (Attenuates some non-verbal vocalizations)
 - Main_406 (Attenuates some non-verbal vocalizations)
 - Kim Inst (Attenuates some non-verbal vocalizations)
 - Inst_HQ_3 (Attenuates some non-verbal vocalizations)
 - MDXNET_2_9682 (Attenuates some non-verbal vocalizations)"
- and it's also worth to check HQ_4.

"UVR BVE v2 model [currently on x-minus] is actually full band. There is, however, a small nuance. This model uses MDX VocFT preprocessing, which is not full band. MDX VocFT model is rebalancing the song. The music is slightly mixed with the vocals (25% music + 100% vocals). This mix is then processed by the BVE model. A small amount of music can help the model better understand the context (it's important for harmony separation). We train the model on a rebalanced dataset. It contains 25% of music." aufr33

All the tips moved to [Tips to enhance separation](#) section

[Screenshot and video showcase](#)

MDX settings & ens. explanations in UVR5
(and also Demucs/VR/MDX v2/23C inferencing parameters)

In one of the pre-5.6 UVR updates, the following min/avg/max features for single models got replaced by a better automated alternative, and you might still get cleaner results of e.g. voc_ft with max_mag on X-Minus or in [this](#) Colab still utilizing it (or downgrade your UVR version).

Now it's only applicable for Ensemble and Manual Ensemble in Audio Tools. Manual Ensemble is very fast, can be used on even old dual-core CPU, as it uses already separated files and simple code - not model.

Ensemble algorithm explanations

Ensemble - a way to use multiple models to potentially get better results.

Rules to be broken here, but:

Max Spec is generally for vocals

(is maximum result of each stem, e.g. in a vocal you'll get the heaviest weighted vocal from each model, and the same goes for instrumental, giving a bit cleaner results, but more artefacts)

Min Spec for instrumentals in most cases

(it leaves the similarity from the models)

Avg Spec is something in between

(gets the average of vocals/instrumentals)

E.g. following the above, we get the following setting:

"Max Spec / Min Spec"

Left side = about the Vocal stem/output

Right side = about the Instrumental stem/output

"Max takes the highest values between each separation to create the new one (fuller sounding, more bleed).

Min takes the lowest values between each separation to create the new one (filtered sounding, less bleed).

Avg is the average of each separation."

More

For ensemble, avg/avg got the highest SDR, then worse results for respectively max/max, min/max and min/min.

For single MDX model, min spec was the safest for instrumental models and gave the most consistent results with less vocal residues than others.

Max spec - is the cleanest - but can leave some artifacts (if you don't have them in your file, then Max Spec for your instrumental like now might be a good solution).

Avg - the best of the both worlds and the only possible to test SDR e.g. at least for ensembles, maybe even to this day if it wasn't patched

"Max Spec/Min Spec" option

For at least a single instrumental model, it's the safest approach for instrumentals and universal for vocals. E.g. Min Mag/Spec in kae Colab using the old codebase for MDX models gives me the only acceptable results with hip-hop. I usually separate using a single model, but I cannot guarantee that Min Spec in UVR and manual ensemble will necessarily work exactly like Min Mag in Colab for a single model. But the explanation remains the same. The best option might even depend on a song.

TL;DR

For vocals bleeding in instrumentals

You can use Spectral Inversion for alleviating problems with bleeding in instrumentals.
Max Spec/Min Spec is also useful in such scenario.

You want less bleed of Vocal in Instrumental stem?

Use Max-Min

For bleeding instruments in vocals

Phase Inversion enabled helps to get rid of transients of the kick which might be still hearable in vocals in some cases.

Set Ensemble Algorithm: Min/Avg when you still hear bleeding.

If still the same, try Min/Max instead of Avg/Avg when doing an ensemble with Vocals/Instrumental output.

Also, you can resign from ensemble setting, and simply use only one clean model on the models list if the result is still not satisfactory.

Further explanations

Why not always go for Min-Max when you want the best acapella?

Why not always go for Max-Min when you want the best Instrumental?

So far, I hear Max-Min on Instrumental sounds more 'muddy/muffled' compared to Avg-Avg.
I bet this will be the same for acapella, but it's less noticeable (I don't hear it).

Hence, I think the best approach would be always going with Avg-Avg.

Then based on the outcome - after reviewing, tweak it based on your desired outcome,
and process again with either Min-Max or Max-Min."

Min = less bleeding of the other side/stem (into this side/stem), but could get sound
muddy/muffled

Max = more full sound, but potential it will have more bleeding

Avg = average, so a bit of all models combined

Average/Average is currently the best for ensemble (the best SDR - compared with Min/Max,
Max/Min, Max/Max).

"Ensemble is not the same as chopping/cutting off and stitching, it blends/removes frequencies. If song 1 has high vocals in the chorus, and song 2 has deep vocals in the chorus, max will mash them together, so the final song will have both high and deep vocals while min will remove both vocals"

"If I ensembled with max, it would add a lot of noise and hiss, if I ensemble with min it would make the overall sound muted gsep."

Technical explanation on min/avg/max

Max - keeps the frequencies that are the same and adds the different ones

"Max spec tends to give more artifacts as it's always selecting the loudest spectrogram frequency bins in each stft frames. So if one of the inputs have artifacts when it should be silent, and even if all other inputs are silent at the same instant, max spec will select the artifacts, as it's the max loud part of spectrogram here." jarredou

Min - keeps the frequencies that are the same and removes any different ones

"if the phases of the frequencies are not similar enough min spec and max spec algorithms for ensembles will create noisy artifacts (IDK how to explain them, it just kinda sounds washy), so it's often safer to go with average"

by Vinctekan

"Min = Detects the common frequencies between outputs, and deletes the different ones, keeps the same ones.

Max = Detects the common frequencies between outputs, and adds the difference to them.

Now you would think that Max-Spec would be perfect since it should combine the all of the strengths of every model, therefore it's probably the best option

That would be the case if it wasn't for the fact that the algorithms that are used are not perfect, and I posted multiples tests to confirm this.

However, it still gives probably the cleanest results, however, there are a few issues with said Max_Spec:

1. Lot of instrumentals are going to be left within the output
2. If you are looking to measure quality by SDR, don't expect it to be better than avg/avg

The average algorithm, basically, combine all the outputs and averages them. Like the average function in Excel.

The reason why it works best is that it does not destroy the sound of any of the present outputs compared to Max_Spec and Min_Spec

The 2 algorithms still have potential for testing, though."

More on how the ensemble in UVR works

"Max takes the highest values between each separation to create the new one (fuller sounding, more bleed).

Min takes the lowest values between each separation to create the new one (filtered sounding, less bleed).

Avg is the average of each separation."

"[E.g.] HQ 1 would be better if the ensemble algorithm worked how I thought it did.

It was explained to me that [ensemble algorithm] tries to find common frequencies across all the outputs and combines them into the result, which to me doesn't actually seem to happen when HQ1 manages to bring vocals to the mix in an 8 model ensemble, how is it not like "okay A those are vocals, and B you're the only model bringing those frequencies to me trying to imply that they are not vocals" and discard them. I mean I am running max/max, but I swear all avg/avg and min/min do is lower the volumes [see [enemble in DAW](#)], It's hard to know without days of testing"

"If u try avg/avg it will get quite muddy on instr result than max/max. But some song if you put kim vocal 1 will get vocal residue on the result".

4-5 max ensemble models rule

Q: Why I shouldn't use more than 4-5 models for UVR ensemble (in most cases)

A: It's easier to get, when you separate the same song using some models. Get the best 4-5 models out of the most recommended currently, plus make some more separations, using some random ones. Then try to reflect avg spec from UVR by importing all of these results to your DAW.

You'll do it by decreasing volume by 3dB per one stem, so for a pair you need to decrease the volume of two stems by 6dB (possibly 6.02 as well). Decrease the volume by the same value further for more than a pair for all stems accordingly, so you'll get pretty much similar result like avg spec in UVR.

You can also maybe apply a limiter on the master. In the second variant, manipulate the volume of all stems by your taste instead of keeping the same volume. By this process, you can observe that the more results imported above 4-5 results, the worse result you have when you don't decrease volume of worse results. When you have control over the volume of single results, you'll end up decreasing the volume of bad results (or deleting them completely). You don't have this opportunity in UVR using avg spec - so like in the first variant in your DAW when you set the same volume for all results. The only way to not deteriorate the final result further, is to delete such worse results from the bag entirely, to not worsen the final outcome when you have too many models ensembled. Without the

possibility of decreasing volume of such a result when all volumes are equal, the more results you'll import to the bag of the 4-5 the best models, the worse final result you'll get. Because you cannot compensate for bad results in the bag by decreasing their volume like in avg spec - all tracks are equally loud in the bag of avg to the models with good results - hence, good models sound quieter if they are in minority and the final outcome is worse. The 4-5 max models ensemble rule is taken from long-conducted tests of SDR on MVSEP multisong leaderboard. When various ensembles were tested in UVR, most of these combinations didn't consist of more than 4-5 models, because above that, SDR was usually dropping. Usually due to all the reasons I mentioned.

Even using clever methods of using only certain frequencies of specific models, like in ZFTurbo, jarredou and Captain FLAM code from MDX23 (don't confuse with MDX23C arch) and its derivations, which minimize the influence of "diminishing returns" when using too many models I think they never used more than 4-5 in their bags, and they conducted impressive amount of testing, and jarredou even focused on SDR during developing his fork (actually OG ZFTurbo code too).

For vocal popping in instrumental issue, read about [chunks](#) or update UVR to use a better option used automatically (called batch mode) if you didn't update to 5.6/+ for a long time already, but the issue might still occur on GPUs with less than 11GB VRAM (and earlier patches doesn't have Roformers support).

MDX v2 parameters (e.g. HQ_1-5, Kim inst, Inst 1-3, NET, Crowd)
(self.n_fft / dim_f / dim_t inference parameters later below)

Segments 512 had better SDR than many higher values on various occasions (while 256 has lower SDR, and has almost the same separation time).

Segments 1024 and [0.5](#) overlap are the last options before processing time increases very much.

Don't exceed an overlap of 0.93 for MDX models, it's getting tremendously long with not much of a difference.

Overlap 0.7-0.8 might be a good choice as well.

Segments can also ditch the performance AF - segments 2560 and 2752 (for 6GB VRAM) might be still a high, but balanced value, although not fully justified SDR-wise, as 512 or 640 can be better than higher values for many songs.

In UVR and Not Eddy's Colabs you can change segment size from 512 to 32 in order to possibly get better results with some older models like e.g. 438 (it tremendously increases separation time).

Overlap: 0.93-0.95 (0.7-0.8 seems to be the best compromise for ensembles, with the biggest measured SDR for 0.99)

Best measured SDR on MVSEP leaderboard have currently following settings (but it was measured on 1-minute songs, so it can be potentially different for your song):

Segment Size: 4096

Overlap: 0.99

with 512/0.95 worse by a hair (0.001 SDR) and 0.9 overlap for as long, but still not tremendously long processing time (1h30m31s vs 0h46m22s for multisong dataset on GTX 1080 Ti).

Also, segments 12K performed worse than 4K SDR-wise (counterintuitively to what it is said, that higher means better result, but maybe diminishing returns at some point here, so too big values maybe cause SDR drop in some cases)

It seemed to be correlated with set overlap.

For overlap 0.75, segments 512 was better than 1024,
but for overlap 0.5, 1024 was better, but the best SDR out of these four results has 0.75/512 setting, although it's a bit slower than 1024, but for 0.99 overlap, 4096 segments were better than 512.

SDR difference between overlap 0.95 and 0.99 for voc_ft in UVR is 0.02.

Segment size 4096 with overlap 0.99 ([here](#)) vs 512/0.95 ([here](#)) showed only 0.001 SDR difference for voc_ft and vocals in favour of the first result.

Difference between segment size 512 with overlap 0.25 ([here](#)) vs 0.95 ([here](#)) is 0,1231 SDR for the latter.

The difference between default segment size 256 with overlap 0.25 ([here](#)) vs 512/0.95 ([here](#)) is 0,1948 SDR for vocals, and 0,1969 with denoiser on (standard, not model), and 0.95 is longer by triple.

1024/0.25 vs 256 has not much longer processing time (7 vs 6 mins) than default settings, and better SDR by 0.0865

For overlap [0.75](#), segments 512 were better than 1024 (at least on 1 minute audio).

Measurement is logarithmic, meaning that 1 SDR is 10x difference.

Be aware that increasing only overlap to e.g. 0.5 from default 0.25, when segments are still at default 256 will muddy the result a bit (might be more noticeable with denoise model enabled), while increasing segments (at least up to 480/512) suppose to add more clarity.

At least on the second beta Roformer patch, max supported segment size on 4GB AMD/Intel GPUs is 480 (at least for 4:58 and HQ 1-3 can sometimes only work with lower 448 - higher overlap and segment size crashes).

256/0.5 also works, at least with HQ 4 (but crashes with 480 segments)

480/0.38 works too, but you can settle on e.g. 0.31 if it's too muddy.

Try not to keep too many opened apps during separation, as drawing their interface also eats up VRAM on the GPU.

MDX-Net v2 max balanced settings:

Segment Size: 2752 (1024 if it's taking too long as it's the last value before processing time increases really much; at least SDR-wise, 512 is better in every case than default 256 unless overlap is increased, and still gets good SDR results)

Overlap: 0.7-/0.8

Denoising

Denoise option used to increase SDR for MDX-Net v2, but instrumentals get a bit muddier ([result](#)).

Denoise model has slightly lower SDR ([result](#)).

For MDX23C models it somehow changed and using standard denoiser doesn't change SDR.

Spectral Inversion

On bigger dataset like Multisong Leaderboard decreases SDR, but sometimes you can avoid some e.g. instrumental residues using it - can be helpful when you hear instruments in silent parts of vocals.

Explanation:

"When you turn on spectral inversion, the SDR algorithm is forced to invert the spectrum of the signal. This can cause the SDR to lose signal strength, because the inverse of a spectrum is not always a valid signal. The amount of signal loss depends on the quality of the signal and the algorithm used for spectral inversion."

In some cases, spectral inversion can actually improve the signal strength of the SDR. This is because the inverse of a spectrum can sometimes be a more accurate representation of the original signal than the original signal itself. However, this is not always the case, and it is important to experiment with different settings to find the best results.

Here are some tips for improving the signal strength of the SDR when using spectral inversion:

* Use a high-quality input. The better the quality of the signal, the less likely it is that the SDR will lose signal strength when the spectrum is inverted. (...)"

Further, there is also about picking a good inversion algorithm and experimenting with different ones, but UVR seems to have one to pick anyway.

Q: I noticed https://mvsep.com/quality_checker/leaderboard2.php?id=2967 has Spectral Inversion off for MDX but on for Demucs. The Spectral Inversion toggle seems to apply to both models, so should it be on or off?

A: Good catch.

Once u put it on for one or the other, both will be affected indeed.

I've enabled it (so for both, actually) [for this result].

MDX v3 parameters (e.g. MDX23C-InstVoc HQ and 2 and MDX23C_D1581)

(biggest measured SDR)

Segment Size: 512

Overlap: 16

(default)

Segment Size: 256

Overlap: 8

The “512/16 is slightly better for big cost of time” vs the default 256/8.

- On a GPU with lots of VRAM (e.g. 24GB), you can run two instances of UVR, so the processing will be faster. You only need to use 4096 segmentation instead of 8192.

It might be not fully correct to evaluate segment and overlap SDR-wise based on measurements done on multisong dataset, as every single file in the dataset is shorter than average normal track, and that might potentially lead to creating more segments and different overlaps than with normal tracks, so achieved results won't fully reflect normal separation use cases (if e.g. number of segments is dependent on input file). Potentially, the problem could be solved by increasing overlap and segments for a full length song to achieve the same SDR as with its fragment from multisong dataset.

Recommended balanced values for various archs between quality and time for 6GB graphic cards:

VR

Window Size: 320 (best measured SDR)

Faster value for slow PCs: 512

Slower, might give more artefacts: 272

Worse: 768, 1024

Read more in [VR settings](#)

Demucs

Segment: Default

Shifts: 2 (def)

Overlap: 0.5

(experimental: 0.75,
default: 0.25)

The best SDR for the least time for Demucs (more a compromise, as it takes much longer than default settings ofc - "best SDR is a hair more SDR and a sh*load of more time"):

Segments: Default

Shifts: 0

Overlap: 0.99 (max can be 0.999 or even more, but it's getting tremendously long)

Best results for instrumentals as input (tested in Colab):

Segments: Default

Shifts: 10 (20 is max possible)

Overlap: 0.1

"Overlap can reduce/remove artifacts at audio chunks/segments boundaries, and improve a little bit the results the same way the shift trick works (merging multiple passes with slightly different results, each with good and bad).

But it can't fix the model flaws or change its characteristics"

In case of Voc_FT it's more nuanced... there it seems to make a substantial difference SDR-wise.

The question is: how long do you wanna wait vs. quality (SDR-based quality, tho)"

In UVR and Not Eddy's Colabs you can change segment size from 512 to 32 in order to possibly get better results with some older models like e.g. D1581 (but it tremendously increases separation time).

For lack of spectrum above 14.7kHz

E.g. in such ensemble:

5_HP-Karaoke-UVR, 6_HP-Karaoke-UVR, UVR-MDX-NET Karaoke, UVR-MDX-NET Karaoke 2

Set Max Spec/Max Spec instead of Min Spec/Min Spec, and also hi-end process (both need to be enabled for fuller spectrum).

Karaoke models are not full band, even VR ones are 17.7kHz and MDX are 14.7kHz IRC. Setting Max Spec with hi-end process will give around 21kHz output in this case.

Cutoff with min spec in narrowband models is a feature introduced at some point in UVR5 GUI for even single MDX models in general, and doesn't exist in CLI version. It's to filter out some noise in e.g. instrumental from inversion. Cutoff then matches model training frequency (in CLI MDX, vocal model after inversion with mixture gives full band instrumental). Also, similar filtering/cutoff is done in ensemble with min spec.

More settings explanation

Leaving both shifts and overlap default vs shifts 10 decreases SDR by only 0.01 SDR in ensemble, but processing time is much faster - 1.7x for each shift. Also, 0.75 overlap increases SDR at least for a single model when even shift is set to 1)

It takes around 1 hour 36 minutes on a GTX 1080 Ti for 100 1-minute files.

"And 18 hours on i5-2410M @2.8 for 5:04 track.

Rating 1 Ensemble on a 7-min song to compare.

Time elapsed:

1080Ti = 5m45s = 345s = 100%

4070Ti = 4m49s = 289s = 83,8%

4070Ti = ~16% faster

1080Ti = ~€250 (2nd hand)

4070Ti = €909 (new)

Conclusion: for every 1% gain in performance, u pay €41 extra (€659 extra in total)." Bas

More min/max explanations moved to [MDX/Ensemble settings](#)

Compensation values for MDX v2
(no longer necessary since MDX23C)

"Volume compensation compensates the audio of the primary stems to allow for a better secondary stem."

For the last Kim's ft other instrumental model, 1.03 or auto seems to do the best job.
For Kim vocal 1 and NET-X (and probably other vocal models), 1.035 was the best, while 1.05 was once calculated to be the best for inst 3/464 model, but the values might slightly differ in the same branch (and compensation value in UVR5 only changes secondary stem - changing compensation value in at least UVR GUI for inst models doesn't change SDR of instruments metric)

self.n_fft / dim_f / dim_t parameters

These parameters directly correspond with how models were trained. In most cases they shouldn't be changed, and automatic parameter detection should be enabled.

- Fullband models:

`self.n_fft = 6144 dim_f = 3072 dim_t = 8`

- kim vocal 1/2, kim ft other (inst), inst 1-3 (415-464), 406, 427:

`self.n_fft = 7680 dim_f = 3072 dim_t = 8`

- 496, Karaoke, 9.X (NET-X)

`self.n_fft = 6144 dim_f = 2048 dim_t = 8` (and 9 kuielab_a_vocals only)

- Karaoke 2

`self.n_fft = 5120 dim_f = 2048 dim_t = 8`

- De-reverb by FoxyJoy

`self.n_fft = 7680 dim_f = 3072 dim_t = 9`

Roformers (located in MDX-Net menu;
only in UVR Roformer beta [patches](#))

chunk_size

"most of the time using higher chunk_size than the one used during training gives a bit better SDR score, until a peak value, and then quality degrades.

For Roformers trained with 8 sec chunk_size, 11 sec is giving best SDR (then it degrades with higher chunk size)

For MDX23C, when trained with ~6 sec chunks, iirc, peak SDR value was around 24 sec chunks (I think it was same for vit_large, you could make chunks 4 times longer)

How much chunk_size can be extended during inference seems to be arch dependant." - jarredou

Be aware that increasing chunk_size consumes much more VRAM, and for 4GB VRAM AMD/Intel GPUs, the max supported will be chunk_size = 112455 (2,55s), sometimes chunk_size = 132300 (3s). CUDA has garbage collector which might make VRAM usage more efficient.

"Conversion between dim_t and chunk_size [dim_t was used in the old Roformer beta 2 UVR patch]

dim_t = 801 is chunk_size = 352800 (8.00s) - maximum value working on AMD/Intel 8GB GPUs and 900MB, at least Mel models

dim_t = 1101 is chunk_size = 485100 (11.00s)

dim_t = 256 is chunk_size = 112455 (2,55s) - maximum value for AMD/Intel 4GB GPUs

dim_t = 1333 is chunk_size = 587412 (13,32s)

The formula is: chunk_size = (dim_t - 1) * hop_length" - jarredou

Unless you turn off Segment default in Options>Advanced MDX-Net>Multi Network Options, chunk_size is being read from the yaml of the model.

Inference mode

Can be found in the menu Multi Network Options menu above. Turning it off will fix the issue of silent separations on older GTX GPUs (iirc GTX 900 and older), but it might make separation slower for other, at least Nvidia GPUs.

It was implemented in one of the latest beta Roformer patches, so if you noticed any slowdowns since updating UVR, try enabling it (now it's disabled by default).

batch_size

Inference Colab by jarredou forces 1 (clicks with that setting were fixed in MSST later), and using above 2 might increase VRAM usage. In newer patches, Anjok started to use MSST inference code for Roformers and MDX23C, hence it might have inherited its usage.

Technical explanation how it works near the [end](#) of this document (scroll down a bit).

Overlap

4 is a balanced value in terms of speed/SDR according to [measurements](#) (since the beta patch #3 or later used above, overlap 16 is now the [slowest](#) (not overlap 2 anymore) and overlap 4 has a bigger SDR than overlap 2 now).

Some people still prefer using overlap 8, while for others it's already an overkill.

There's very little SDR improvement for overlap 32, and for 50 there's even a decrease to the level of overlap 4, and 999 was giving inferior results to overlap 16.

Compared to overlap 2, for 8 "I noticed a bit more consistency on 8 compared to 2 (less cut parts in the spectrogram)."

Instrumentals with overlap higher than 2 can get gradually muddier.

Calculations above were based on evaluations conducted on multisong dataset on MVSEP. Search for e.g. overlap 32 and overlap 16 below, and you will see the results to compare:

https://mvsep.com/quality_checker/multisong_leaderboard?algo_name_filter=kim

"overlap=1 means that the chunk will not overlap at all, so no crossfades are possible between them to alleviate the click at edges."

The setting in GUI overrides the one in model's yaml.

Refer to UVR Roformer beta [patch](#) section for more detailed information

[Tips to enhance separation results]

If you cannot achieve good separation, you can conduct the following experiments

1. *De-bass*

Turn down all the bass to stabilize the voice frequencies of your input song (example EQ curves: [1](#) and [2](#)).

Male setting: cut all below 100Hz + cut all above 8kHz.

Female setting: cut all below 350Hz + cut all above 17kHz.

This works, because jitter is reduced a lot.

2. *De-reverb*

You can also test out the de-reverb e.g. in RX Advanced 8-10 on your input song. One or both combined in some cases may help you get rid of some synth leftovers in vocals.

Alternatively (not tested for this purpose), you can also try out [this](#) or [this \(dl\)](#) (is in UVR's Download Center) de-reverb model (decent results). Currently, the VR dereverb/de-echo model in UVR5 GUI seems to give the best results out of the available models (but RX or others described in the models list section at the top can be more aggressive and effective with more customizable settings).

3. Unmix drums (mainly tested on instrumentals)

Separate an input song using 4 stem model, then mix the result tracks together without drums and separate the result using strong ensemble or single vocal or instrumental model (doesn't always give better results).

Alternatively, unmix bass as well. There's great bass+drums BS-Roformer model released for UVR (currently in beta)

4. Pitch it down/up (soprano/tenor voice trick + ensemble of both)

- You can use <https://github.com/JoeAllTrades/SpectraDownshift> for it (it's based on scipy, it's lossless, so fully reversible and nulls), or:
 - Already implemented option in newer versions of UVR under "Shift Conversion Pitch" in Settings>Choose Advanced Menu>Advanced [Arch] Options>
And there are positive and negative values when you scroll up and down (lossy, even more than soxr).

Negative value will slow down the track before separation, so e.g. model with cut-off will be compensated for its band lost a bit after speeding up again.

If you slow down the input file, it may allow you to separate more elements in the "other" stem of 4-6 stems separations of Demucs or GSEP (when it's done manually).

It works either when you need an improvement in such instruments like snaps, human claps, etc. The soprano feature on x-minus works similarly (or even the same), it's also good for high-pitched vocals.

Be aware that low deep male vocals might not get separated while using this method (then use tenor voice trick instead - so pitch it up instead of pitching it down).

Also, it serves the best for hard paned songs (e.g. 1970 and pre era, e.g. The Beatles, etc).

Also, it works great for drums. While evaluation on multisong dataset on MVSEP, it decreases SDR by around 1.

"Basically lossless speed conversion a.k.a. soprano voice trick done manually:

Do it in Audacity by changing sample rate of a track, and track only (track > rate), it won't resample, so there won't be any loss of quality, just remember to calculate your numbers
44100 > 33075 > 58800
48000 > 36000 > 64000
(both would result in x0.75 speed)
etc." (by BubbleG)

Q: Won't the result be sped up?

A: "No. Because when you first slow it down, after processing with said model it gets converted to 44100 again (only the sample rate, not the actual speed), so speeding it up brings the speed back to normal" becruily

Q: I don't quite get what I'm supposed to do though, just slow down the file to 0.75x and then export in 58800?

A: "change the sample rate to 33075 Hz,
then export at whatever sample rate
process then,
change the sample rate of the processed file to 58800 Hz
key word being change, not resample
like [this](#), click other and the pick the correct samplerate" Dry Paint Dealer

4b*. If you have a mix of soprano and baritone voices, you possibly can do:

- "1. Soprano mode (slow down sample rate), then bring back to normal
after that
2. Tenor mode (speed up sample rate), then bring back to normal
and finally combine the two with max algorithm"

Making an ensemble of such results can also increase the quality of separation.

5. Use 2 stem model result as input for better 4-6 stem separation

You may get better results in Demucs/GSEP/MDX23C Colab using previously separated good instrumental result from UVR5 or elsewhere (e.g. MDX HQ3 fullband or Kim inst narrowband in case of vocal residues, or BS-Roformer 1296)

6. Debleed

If you did your best, but you still get some bleeding here and there in instrumentals, check RX 10 Editor with its new De-bleed feature. [Showcase](#)

[More](#) methods of debleeding stems.

7. Vocal model>karaoke model

You might want to separate the vocal result achieved with a vocal model with MDX B Karaoke afterwards to get different vocals (old model).

8. The same goes for unsatisfactory result of instrumental model - you can use MDX-UVR Karaoke 2 model to clean up the result, or top ensemble or GSEP like for cleaning inverts (old models)

9. Mixdown of 4 stems with vocal volume decreased for final separation

An old trick of mine. Used in times of Spleeter to minimize vocal residues.

Process mixture to 4 stems and then mix stems in a way that vocal is still there, but quieter, so lower their volume, and set drums louder, then send the mixture from it to one good isolation model/ensemble, so in result drums after separation will be less muddy, and possible vocal residues will be less persistent.

But it was in times when there wasn't even Demucs (4) ft or MDX-UVR instrumental models, where such issues are much less prevalent.

10. If you use UVR5 GUI and 4GB, you may hear more vocal residues using GPU processing than e.g. while using 11GB GPU (tested on NVIDIA). In this case, use CPU processing instead.

11. *Fake stereo trick*

Aufr33: “process the left channel, then the right channel, then combine the two. [Hence] the backing vocals in the verses are removed” (it still may be poor, but better). “I’m having to process as L / R mono files otherwise I get about 3-5% bleed into each channel from the other channel, but processing individually, totally fixes that” -A5

On an example of Audacity: import your file, click on down arrow in track selection near its label, click [Split Stereo Track](#), go to Tracks>[Add New>Stereo Track](#).

Mark the whole channel, copy and paste on one of the tracks you divided before.

It will overlap the same mono track in stereo track, so the same across both channels.

Do the same for both L and R separately. Then separate with some model both results separately. Then import both files and join their separate channels by method above. Don’t confuse L and R channel while joining both.

12. *Turn on Spectral Inversion in UVR*

it can be helpful when you hear instruments in silent parts of vocals, and sometimes also using denoiser might help for it (although both can make your results slightly muddier)

13. *Chain separation*

For vocal residues in instrumental, you can experimentally separate it with e.g. Kim vocal (or inst 3) model first and then with instrumental model. You might want to perform additional steps to clean up the vocal from instrumental residues first, and invert it manually to get cleaner instrumental to separate with instrumental model to get rid of vocal residues. [Tutorial](#)

14. To not clean silences from instrumental residues in the vocal stem manually, you can use a noise gate in even Audacity. [Video](#)

In some cases, using noise reduction tool and picking noise profile might be necessary.

[Video](#)

15. *Choice of good models for ensemble*

Use only instrumental models for ensemble if you have some vocal residues (and possibly vice versa - use only vocal models for ensemble for vocals to get less instrumental residues) - mainly used in times when there was still strong division between vocal and instrumental models (before MDX23C release). Now it can narrow down to picking only models which doesn't have bleeding - listening all the separate models results carefully, and pick the best 2-5 results to make an ensemble.

16. *For vocals with vocoder*

You can use 5HP Karaoke (e.g. with aggression settings raised up) or Karaoke 2 model (UVR5 or Colabs). Try out separating the result as well (outdated models).

"If you have a track with 3 different vocal layers at different parts, it's better to only isolate the parts with 'two voices at once' so to speak"

Be aware that BS-Roformer model ver. 2024.04 on MVSEP is better on vocoder than the viperx' model.

17. *Find some leaked or official instrumental for inversion*

To get better vocals

If you're struggling hard getting some of the vocals:

"I used an instrumental that I don't remember where I found it (I'm assuming most likely somewhere on YouTube) and inverted it and then used MDX (KAR v2) on x-minus and then RX 10 after.

I Just tried the one-off Bandcamp and funny enough it didn't work with an invert as good as the remake that I used from YouTube, but I don't remember which remake it was I downloaded because it was a while ago"

18. *Fix for ~"ah ha hah ah" vocal residues*

Try out some L/R inverting, try out to separate multiple times to get rid of some vocal pop-ins like this

19. *Center channel extraction method*

by BubbleG using Adobe Audition:

"The idea is that you shift the track just enough where for example if you have a hip hop track, and the same instrumental tracks the drums will overlap again in rhythm, but they will be shifted in time so basically Center Extract will extract similar sounds. You can use that

similarity to further invert/clean tracks... It works on tracks where samples are not necessarily the same, too..."

>

Step-by-step guide by Vinctekan ([video](#))

1. You take your desired audio file
 2. Open it in Audacity
 3. Split Stereo to Mono
 4. Click the left speaker channel (now mono), and duplicate it with Ctrl+D.
*: If the original and duplicate is not beside eachother, move it so that it's next to eachother
 - 5: Select the original left speaker channel and it's duplicate, and click "Make Stereo Track"
 - 6: Solo it.
 7. Export it in Audacity, preferably in 44100hz since UVR doesn't output in higher frequencies. Format, and bit depth don't really matter, I prefer wav always.
 - 8: Do the same thing for the right speaker channel.
 - 9: Open UVR
 - 10: Navigate to Audio Tools>Manual Ensemble.
 - 11: Make sure to choose Min Spec (since that function is supposed to isolate the common frequencies of 2 outputs)
 - 12: Select the 2 exported fake stereo files of both the left and right speaker channels.
 - 13: Hit process
-

20. Q&A for the above

Q: For the right channel are you doing the same with the duplicate and moving the file next to the original or just duplicating and making that stereo?

A: Those 2 steps go hand in hand. These reason I mentioned it is because if you try to make a Stereo Track with those 2 (the left/right channel speaker, and it's duplicate mono) when there is a track between them, it doesn't work. Even if you select those 2 with Ctrl held down.

Take that 1 channel (left/right), Ctrl+C, Ctrl+V, now you have 2 of the exact same audio. Hold Ctrl select the 2, click "Make Stereo Track". Finally, export.

21. Passing through lot of models one by one

"I usually do ensemble to make an instrumental first, then demucs 4_ft... sometimes I do it once, then take that rendered file and pass it back through the algo a few more times, depends until it strips out artifacts."

It can be beneficial also in case of more vocal residues of MDX23 or Demucs ft model compared to current MDX models or their ensembles.

22. If you still have instrumental bleeding in vocals using `voc_ft`, process the result further with Kim vocal 2

23. *Rearrange cleaner parts*

When a verse starts, and you start having muddy drums and their pattern is consistent (e.g. some hip-hop), and you have cleaner drums from fragments before the verse starts, you can rearrange drums manually, using 4 stems model and paste that cleaner fragments throughout the track. Sometimes fade outs or intros can have clean loops without vocals, which can be rearranged without even the need of separation. Listen carefully to the track. Such moments can be even briefly in the middle of the song.

24. *arigato78 method for lead vocal acapella*

1) Try to make the best acapella (using mvsep.com site or using UVR GUI). I recommend the MDXB Voc FT model for this with an overlap setting set to at least 0.80 (I used 0.95 for this example). The overlap for this model at mvsep.com is set to 0.80. Speaking of the "segment size" parameter in UVR GUI - changing it from 320 to 1024 doesn't make much of a difference. It acts randomly, but we're working on a beta version of UVR GUI - remember that. (...)

I noticed all the "vocal-alike" instruments still remaining on the acapella track, but wait...

2) The second part is to process the acapella thru the mdx karaoke model (I did it using mvsep.com). I prefer the file with "vocalsaggr" in the name. It has more details than the file with "vocals" in it. The same goes to the background vocals in this case - I prefer the "instrumentalaggr" one.

One important thing - all (maybe almost) of the residue instrumental sounds were taken by mdx karaoke model to the backing vocals stem, leaving the lead vocal almost studio quality ("studio"). But - it may be helpful for all you guys trying to make good acapellas. I was just playing with all the models and parameters and I accidentally came across this. Please, let me know what you think about it. I'm gonna try this on some tracks with flutes, etc. And I realize that this method is not perfect - we get nice lead vocals, but the backing vocals are left with all that sh*tty residues.

So the track is called "Reward" by Polish singer Basia Trzetrzelewska from her 1989 album "London, Warsaw, New York".

—

25. *Uneven quality of separated vocals*

You can downmix your separated vocal result to mono and repeat the separation (works for e.g. BVE model on x-minus).

26. *Experimental vocal debleed with AI for voice*

Sometimes for instrumental residues in vocals, AIs for voice recorded with home microphone can be used (e.g. Goyo [now paid Supertone Clear], or even Krisp, RTX Voice, AMD Noise Suppression, Adobe Podcast as a last resort) it all depends on the type of vocals and how destructive the AI can get.

27. *Minimize vocal residues for very loud songs*

For very loud tracks between -2.5 and -4 iLUFS, try to decrease volume of your track before separation. E.g. for Ripple, -3dB for loud tracks is a good choice. If your track you're trying to separate is already quiet and around -3dB, then the step is not necessary.

27b. You could try out attenuate volume of the mixture before separation (-3/6 dB), but I can't remember whether current MSST uses normalization before anyway. UVR maybe not.

28. *Brief (old) models summary*

MDX-Net HQ_3 or 4 is a more aggressive model for instrumentals, with usually fewer amounts of residues vs MDX23C HQ models or sometimes even vs KaraFan or jarredou's MDX23 Colab v2.3. HQ_3 can give muddier results vs competition, though.

The most aggressive are BS-Roformer models, but they can sound filtered and even muddier at times, but cleaner. It's good to use them with ensemble with e.g. MDX23C model.

voc_ft is pretty universal for vocals (with residues in instrumental, but not less muddy results), while people also liked Ripple/Capcut, although they give more artefacts (use the released BS-Roformer models now for vocals instead). Consider using MDX23C HQ model(s) as well, but they tend to have more instrumental residues.

29. *Cleaning up bleeding between mics in multitracks* (by SeniorPositive)

"Demucs bleed "pro" tip that I figured out now, and I didn't see mentioned, that I will probably try to use every time I hear some bleed between. (...) I was cleaning multitrack from bleed between microphones in conga track, and used demucs for separation drums/rest pair, and [the] other [stem] had some of those bongos still, very very low, but it existed, and I heard it just enough.

- So I took rest signal, boosted it +20db (NOT NORMALISE! Other value but make note how much of it you boosted, go few dbs less to 0db threshold). If you do not boost it to sensible levels, the algorithm will skip it.
 - Do separation once again (this time I've done it using spectralayers one, but it's also demucs)
 - lower result -20dB add this result to first separation result
- [The] result [is -] better separation, fewer data in other/bleed and with proper proportions.

It looks like AI is not yet perfect with low volume information and, as seen in ripple Bas Curtiz discovery, too hot content also."

Showcase

30. For *clap leftovers in vocal stem*

Methods suggested in [debleeding](#)

31. (paraphrase of point 17)

Use traditional phase inversion method and then feed them to the UVR models if you had a chance finding any official instrumental or vocal, but it doesn't invert perfectly. This way, the models will have less noisy data to work with. But it sometimes happens that the official instrumental and the vocal version of tracks have slightly different phasing. This makes isolating vocals via phase inversion difficult, or even sometimes impossible ~Ryan_TTC
Sometimes only specific fragments of song will align, and in further parts of the track it will stop and require manual aligning. You may try to use Utagoe or possibly UVR with Aligning in Audio Tools as it shares some similar functionalities.

Why official stems don't invert?

"Very rarely will the vocal or instrumental fully invert out of the master. This is because of master bus processing and non-linear nature of that processing. I.e. part of the masters sound is the processing reacting to the vocal and instrumental passing through the same chain.

Sidechaining and many limiters are also looking ahead to the signal. Also, some processing is non-linear so even if you set it up identically re. settings, each bounce will be slightly different in nature. Stuff like saturation/distortion. Some reverbs, limiters and transient shapers etc are not outputting the same signal / samples every time you bounce, so instrumental bounce is not the same as the master bounce in terms of phase inversion." - Sam Hocking

32a. *Muddiness in instrumentals of some BS-Roformer models*

Invert (at best lossless) mixture (original song - instrumental mixed with vocals) with vocal result of separation. It might increase vocal residues outside busy mix parts.

Inverting vocals instead of mixture will result in less residues, but more artificial results in busy mix parts.

Similar trick might even increase SDR for MDX23C models irc.

How to perform inversion is explained somewhere in this [doc](#) by Bas Curtiz.

It might be unnecessary to use in UVR - it might use this trick for BS-Roformer models already, but for 2024.02 on MVSEP it was beneficial.

The trick is not necessary for 04.2024 BS-Roformer model (it sounds worse after inverting).

Furthermore, for some muddiness in this model, you can use the premium's feature - ensemble. The default output without intermediates should be enough (min_fft is very muddy, and max_fft very noisy). Strangely, the result from Roformer from intermediates might sound v. slightly better (maybe it was something random). The ensemble is kinda mimicked in jarredou's MDX23 v2.4 [Colab](#) and to some extend it can be mimicked in UVR by using 1296+1297+MDX23 HQ ensemble (or copy of 1296 result via Manual ensemble instead, for faster processing).

Now also x-minus has drums ensemble feature for Roformer models.

32b. Fixing *muddiness for MDX-Net* (on example of HQ_3 model) - *inverting trick*

It's less muddy when mixture is inverted and mixed with separated vocals in louder parts, but vs the instrumental stem, it's worse in silent parts with less busy mix - then it has more vocal residues than the instrumental stem.

When vocals were inverted instead of mixture, it was more muddy, but still more residues were present vs OG inst. stem, just a bit less. Can't tell how it's SDR-wise.

So you can combine various fragments for the best results.

33. *Descriptions of models, pt. 2*

Muddiness of instrumentals in specific archs

Beside changing min/avg/max spec for MDX ensembling (or in Colab for single models), plus aggression for VR models, or manipulating shifts and overlap for Demucs models, you need to know that some models or AIs sound usually less muddy than others. Like e.g. VR tends to have less muddiness vs MDX-Net v2 arch, but the first tends to have more vocal residues. Consider using HQ2/3/4/inst3/Kim inst for fewer residues than in VR arch or BS-Roformer.

For less muddiness than in MDX-Net, consider using MDX23 Colab 2.0/2.1 or 2.2 (more residues) or KaraFan (e.g. preset 5).

34. *Muddiness of 4/+ stem results after mixdown*

UVR5 supports even 64 bit output for Demucs, eventually you can use Colab or CML version for 32-bit float, but mvsep.com supports 32 bit output in MDX23 model when you choose WAV. It has better SDR vs Demucs, anyway, but sometimes more vocal residues.

Then, on MVSEP beside 4 stems, you have also instrumental - ready mixture of the three for instrumental in 32 bit provided, which is not bad, but you can go to extreme, and download e.g. Cakewalk, and 3 stems separately, and now in Cakewalk:

- 1) Don't use splash screen project creation tool, close it
- 2) Go to new
- 3) Pick 44100 and 64 bit

- 4) Make sure that double 64 bit precision is enabled in options
- 5) Import MDX23 3 stems (without vocals)
- 6) Go to file>Export
- 7) Pick WAV 64

Output files of 64 bit mixdown are huge, but that way you get the least amount of muddiness as possible. If only MDX23 model doesn't give you much more vocal residues vs MDX-UVR inst models or top ensemble which you wouldn't accept.

Be aware that 32-bit float vs 16 bit outputs can sound more muddy. Probably due to the fact that most sound cards/DACs don't have native 32-bit float output support in drivers and additional downsampling must be done in-fly during playback, probably even if some drivers allow using 32-bit output in Sound settings in Control Panel for the same device (while other version might not).

Spectrum-wise, instrumentals downloaded from MVSEP vs manual mixdowns are nearly identical. The only difference in one case I saw was in an instrumental intro in the song where the site's instrumental had more high end, maybe noise, but besides, spectrum looks identical at first glance without zooming it. Still, when I performed mixdown to anything lower than 64 bit, I didn't get comparable clarity to the site's instrumental. Maybe I'd need to change some settings, e.g. change project bit depth to the same 32 bits as stems and later perform mixdown to 64 bit. Haven't tested it yet.

35. Debleeding of drums in vocals by Sam Hocking

For drums, I usually try and do some kind of sidechained denoise using the demixed Drum stem itself as the signal to invert with. If you shape 'shape' the sidechained input using spectral tools/filters/transient tools etc, you can often null more of the drum out of the vocal. My favourite tool for this is Bitwig Spectral Split, but there's several FFT Spectral VSTs out there. The key is the tools has smoothing to extend the transients in time a bit so they null more.

Difficult to audibly hear on a video, but here's a vocal stem with a lot of residue I've exaggerated in a passage without singing. I turn on a sidechain bass, drums and other stem to phase invert them out the vocal a bit via the spectral transient split in Bitwig. I then take a spectral noiseprint in Acon Digital of what's left and that works as a mild denoiser, but only after the inversion has done its thing. Don't take the noise print until you're happy everything else is inverting out as much as you can get it, and it's not noticeable.

36. Manual MDX23 stems mixdown issues

It can happen that after importing three stems from MDX23 or other arch, into the same session, all combined they sound so loud that they clip on the master fader. I'd rather suggest that, in many cases it can be ignored, as after mixdown it will be fine in most cases

and better than with using limiter, but it also depends on a song loudness of how much clipping even the instrumental from single model will have:

37. Q: *Why sometimes separated instrumentals have clipping?*

A: "Mixture doesn't clip, but instrumental is clipping.

This is because where the instrumental is clipping in positive values, the vocals are in negative values, and so vocals are lowering instrumental peak value when mixed together.

If you separate a song peaking at 0 with high loudness, the instrumental will probably clip because of this (and the more loudness, the more chances this clipping can happen, as waveform is brickwalled toward boundaries values). It's the laws of physics, as that's because of these laws that audio phase/polarity inversion works.

That's why Demucs is using the "clamp" thing, or can also lower the volume of the separated stem to avoid that clipping.

- Most of the time, lowering your input by 3dB solves that issue.
 - Saving your audio to float32 can be a solution, as "clipped" audio data is not lost in this case" (jarredou)
- So theoretically in 32-bit float, the volume can be decreased after separation and still nothing is lost, and clipping should be fixed.

38. *Separated audio using MDX-Net arch has noise when mixture has no audio and is silent*

Use denoise standard (or denoise model) in Options>Choose Advanced Menu>Advanced MDX-Net Options>Denoise output

39. *MDX23C/BS-Roformer models ringing issue*

"It was reported that maybe DC offset can amplify it. Fixing it with RX before separation was said to alleviate the issue" See the [Screenshot](#) how to do it,
"Don't forget to use "mix" pasting mode" - jarredou

It serves to alleviate the issue of horizontal lines in specific frequencies across the whole track, cause most likely by bandsplitting neural network artifacts. Problem presented above.

Q: Mine is 0.047% for the DC offset, so I would just do 0.047 or 0.04

A: "0.047% is kind of normal value, it's even a great one. No need to fix that.

I don't know at what value it could be become problematic for source separation models.

On some raw instrument recordings, I have seen 20%~30% DC offset sometimes, which can become a real issue for mixing then, as it's reducing headroom" - jarredou

40. *Ensemble of pitch shifted results (point 4 continues)*

So you follow the point 4, and “change sample rate before each separation and restore it after for each, then ensemble them all”.

“on drums it was really working great, where sometimes you have sudden muffled snare because other [stem] masked it, the SRS ensemble [irc used in MDX23 2.x Colab and KaraFan] was helping a lot with that, making separation more coherent across the track.”

41. A5 method for clean separations

Consider the fake stereo trick fist from point 11, separate with BS-Roformer 1296, clean the residues in vocals manually, put the vocals back into mixture - so perform mixdown to have a mixture again, and then separate this mixture with demucs_ft (old models)

42. Using surround versions of songs

Sometimes you can get vocals separated easier from center channel from surround version of the song. Perhaps you might also get different separations of instrumentals from such versions, also with possibility of manipulating the volume of specific tracks before mixdown to 2.0/stereo file. It might be necessary anyway, because otherwise you might run into some errors on an attempt of separation of 5.1 file or with more channels.

Use Dolby Atmos/360 Reality Audio/5.1 version of the song

Multichannel mixes can give better results for separation. For more on Atmos [read](#).

Be aware that center may contain not only vocal, but also some effects.

Consider separating every channel separately, or one pair of channels at the time (rear, front, center, sides separately) or only separate center channel separately and all the rest separately.

Visit [this](#) section for more information.

43. Matchering as substitute of ensemble (UVR>Audio Tools)

If the result of some separation is too noisy, but it preserved the mood and clarity of the instrumental much better than some cleaner, but muddy result, you can use that noisy result as the reference for more muddy target file. E.g. voc_ft used as reference for GSEP 2 stem instrumental output.

44. Retry separation 2–3 times

At least for MDX23C models it happened for someone, that every separation made in UVR differed in terms of muddiness and residues, and someone received satisfactory result after the second or third attempt of separating the same song. Consider turning on Test mode in

UVR, so the few digits number will be added to the output file name, so the results won't be overwritten during the process, and you'll be able to listen and compare them.

45. Ensemble instrumental result with drums with max/max

Can help to fix muddiness of vocal BS-Roformer models, but drums can sound too loud in the end. Consider decreasing their volume before ensemble if necessary.

Drums can be obtained from e.g. demucs_ft (and mixture as input or from some less muddy model) or from MDX23 Colab/MVSEP (which already uses its own input from model ensemble for 4 stems)

46. Use EQ on your song before separation (e.g. for too weak "s" sounds in separated vocals)

It's an old method used in times when models didn't give good quality yet, might be no longer necessary. You can use EQ on a mixture to stress vocals in the mix more, so the separation might turn out to be better.

47. Bas Curtiz [video](#) tutorial for tips and tricks and [document](#)

48. [Aufr33's demudder](#) (more for Roformers than HQ 4)

49. Volume compensation finetuning for MDX-Net models

It can slightly enhance the result, helping fighting muddiness a bit.
It's no longer beneficial for MDX23C and Roformer models.

Volume compensation generally differ for every song. E.g. for HQ_3 model, sometimes 1.035 can be the best, but sometimes 1.022. By default, it affects only vocals, but when you switch primary stem in model settings, so vocals are labelled as instrumentals and vice versa (so how MDX kae Colab works), it can be used also to fine tune instrumental stems.

50. Picking correct models for ensemble (by dca100fb8)

"I'm seeing a certain pattern, if the Mel-Roformer model from x minus leaves faint vocals in the background during silent parts of the song, then it means MDX23 & Demucs 4
htdemucs_ft models should not be used for ensemble because vocals can be heard in the background too using these models, while MDXv2 models will not leave those vocals. So it's either UVR Mel/BS-Roformer 1296 + 1297 + MDXv2 or MDX23 + Demucs + Mel-Roformer X Minus + BS-Roformer 1296 + 1297.

I excluded VitLarge because it always leaves faint vocals"

51. Ensemble only extra higher frequencies

from e.g. HQ 3 model with narrowband inst 3 model - [guide](#)

52. Use some BV/Karaoke model first, to potentially get cleaner instrumental with dedicated model afterwards

53. Set vocals to center with stereo plugin (guide by Musicalman)

Trick [working] with the [now outdated] BS-Roformer karaoke model, though it may work on other karaoke models too (I suspect you might have some mileage with MDX for instance). Anyway, the trick has to do with separating one voice from other sounds. If the voice you want to separate is panned centrally, you're already in luck; the model should expertly separate it. If not, you can rotate the stereo field so that the voice is as close to the center as possible (I use the Reaper js stereo field manipulator plug in for this). Process the rotated sound with the karaoke model and the voice you're looking for will magically be separated, even from other voices! If you need the original stereo image back, simply perform the opposite rotation.

54. Method for cleaner vocals (by YAZKEN*)

“Basically you do 2 vocal extractions, invert the polarity of one of them and render it, after that you invert the rendered audio and choose one of the extractions you’ve made and listen what is cleaner”

55. Spectral editing in Audacity explained (by CC Karaoke)

I typically use Audacity .. But [...] RX11 has some nice shiny toys, so maybe try that. I do things the hard way.

Here's a great basic example when using the (Roformer) Karaoke model. It can really sometimes struggle with the hard consonants.

So for best results, you'll often need to isolate those in the main vocal stem by muting out all the surrounding sound, and then mixing them back in with the backing vocal stem. Of course by doing this the hard consonants will often be too loud, so you can de-amp the volume on them and then play back till you get a level that sounds like it blends properly.

<https://imgur.com/a/dyc9olh>

Slightly more complex example; Where the vocal lines are overlapping. I tried drawing green over one of the lines to show the difference. Might be a couple mistakes lol, as I haven't checked, but you get the idea. The previous sound is a carrying note, whereas the next line is a 'HA' kinda hard hit punching words, so it has a different shape to it. This kind of more obvious difference is easier than say... reverb...

<https://imgur.com/a/BFGqN7P>

jarredou's hint: That's a case where I would go SpectraLayers as while the 2 vocals are not on the same pitch, you can separate them manually (with harmonic selection tool). At least for that small part shown here.

In SpectraLayers, you can change FFT resolution, higher value will give you more defined freq "picture", and it can help when 2 parts are really close in pitch, like here.

The downside is that with high FFT values, you lose time resolution. So to use SpectraLayers manual selection efficiently, you often need to switch that FFT resolution value depending on the elements you are targeting, like you would zoom/dezoom in photoshop while editing a picture.

56. Sequential stem separation (by dynamic64/isling)

With single stem models, feel free to experiment with sequential stem separation - Instrumental model first, then drums or bass, piano or guitar, strings or horns. It depends on the song whether better results will give e.g. drums or bass when separated first, the same to piano vs guitar and strings vs horns first.

57. Advanced chain processing chart ([image](#))

It's a method utilizing old models, and e.g. Kim Vocals 2 can be potentially replaced by unwa's BS/Mel-Roformer models in [beta UVR](#) (or other good method for [vocals](#)) or ensembles mentioned in this document. Check the best current methods for vocals in one stem to find what works the best for your song to get all vocals before splitting to other stems using this diagram.

htdemucs v4 above can be replaced by htdemucs_ft, as it's the fine-tuned version of the model (or [MDX23 Colab](#)). Even better, you can use some of the methods for [4 stems](#) in this GDoc (like drums on x-minus).

De-echo and reverb models can be potentially replaced by some better paid plugins like: DeVerberate by Acon Digital, Accentize DeRoom Pro (more in the [de-reverb](#) section).

UVR Denoise can be potentially replaced by less aggressive Aufr33 model on x-minus.pro (used when aggressiveness is set to minimum), and there's also newer Mel-Roformer (read [de-reverb](#) section).

As for [Karaoke](#) models, there's e.g. a Mel-Roformer model on x-minus.pro for premium users or MVSEP/jarredeou inference [Colab](#).

"If the vocals don't contain harmonies, this model (Mel) is better. In other cases, it is better to use the MDX+UVR Chain ensemble for now.". It is possible to recreate to some extent this approach while not using BVE v2 models, by processing the output of main vocal model by one of Karaoke/BVE models in UVR (possibly VR model as the latter) using Settings>Additional Settings>Vocal Splitter Options, so it separates using one model, then it uses the result as input for the next model (see the Karaoke section).

MedleyVox (not available in UVR) will be useful in the end in cases when everything else fails after you obtain all vocals in one stem, as it's very narrowband. But you can use AudioSR on it afterwards.

58. See [here](#) for more on *cleaning/debleeding*
59. *Reverse polarity and/or remove DC offset* of the input file
60. Find *fragments* of instrumentals in your song and *overlap them inverted across the whole song* before separation (`heauxdontlast`)
60. *Method for better quality of instrumental leaks on YT by theamogusguy*

“I did something really odd. (...) since you can only rip max 128kbps I did something really odd to get a higher quality instrumental:

I inverted the 128kbps AAC YouTube rip into the original to get the acapella

I took the subtracted acapella and ran it through AI (Mel-Roformer 2024.10) to reduce the compression artifacts

I then inverted the isolated acapella and mixed it with the lossless to get an... unusual lossless instrumental file?

Also, the OPUS stream goes up to 20kHz, but I feel like the sample rate difference is going to cause issues, so I ended up ripping AAC (OPUS is 48khz while most music is 44.1kHz)”

61. *Join the best fragments from various models*

E.g. unwa inst models might be noisy at times, so you might want to use specific fragments of v1e/v1/v2 fitting across the song, or e.g. beta 4 vocal model in certain fragments where it's not enough, though it is more muddy, but less noisy than unwa's inst models. In some cases, if it's still not enough, you might want to use BS-Roformer models like unwa's Large or e.g. 24.10 on MVSEP. Just find which model on the list in this document has the least amounts of residues and experiment with the rest starting from models listed at the top.

62. *Lowpassing lossless file to 20kHz*

Sometimes it's a bit useful in getting rid of some constant faint noise/residues from vocals in instrumentals. It might muffle some unwanted parts of instrumental, but some more difficult fragments with more residues than usual might sound better that way. Tested on FLAC 16 compressed to mp3 320kbps, but it should work better with lowpassing using EQ instead of compressing. Other example values you might want to try out using are 19kHz (mp3 VBR V0 cutoff)/17.7kHz (cutoff of some narrowband models)/16kHz (cutoff of mp3 and AAC 128kbps)/14.7kHz (D1581 model cutoff).

A possible explanation of why it might sometimes work is: sometimes, e.g. more oldschool hip-hop beats might have less higher tones, or even none above 16kHz, so most of the information in this area might come from vocals in a mixture. You can recognize it especially if vocals lose much more clarity than beat in the mixture once you compress it to e.g. mp3 VBR V0 (19kHz cutoff) or lower.

63. Refrain from excessively stacking models (e.g. for RVC)

“Inst Voc, Kim Vocals, Denoise, ensemble mode, and so forth can introduce noises to your dataset as it rips away frequencies from your audio. This harms the model fidelity and quality.” [more](#)

64. Get cleaner vocals with vocal and instrumental model mixdown (e.g. of Mel becruily models) by [Havoc](#)/mrmason347

Separate with becruily Mel Vocal model and its instrumental model variant, then get vocals from the vocal model, and instrumental from instrumental model, import both stems for the DAW of your choice (can be Audacity) so you’ll get a file sounding like original file, then export - perform a mixdown of both stems, then separate it with vocal model

65. Less vocal bleed with dim_t 256 or corresponding [chunk_size](#) (cypha_sarin)

Small difference observed on 6GB NVIDIA GPU and Gabox instv5 model where “one little vocal glitching sound from the song that only gets picked up when the segment size is lower [256]”

66. If you set 24-bit output in UVR>Options>Additional settings (or ev. 64-bit) for e.g. demudder, the results might be slightly less muddy

67. *Clean loop of the instrumental used for Matchering and full separation*

You can use well sounding fragment of single instrumental model separation with high fullness metric as a reference for Matchering in UVR for [phase-fixed](#) muddy result set as target. It will have less bleeding than models with low bleedless metric, but still fuller than phase-fixed results.

68. chunk_size 112455 and overlap 50

To have the best SDR for Roformers, use chunks not lower than 11s, which is usually training chunks value (rarely higher). Although, at times people get better results with 2,55s chunks (called chunk_size 112455 since UVR Roformer patch #3). But be aware that e.g. using becruily Karaoke model, using low 2,55s chunk will lead to crossbleeding. dim_t to chunk_size conversion is later [here](#). Sometimes even go to extremes and use e.g. overlap 50 claiming that it was better with 112455 and becruily inst model (thx gustownis)

69. If you want smoother vocals from e.g. Beta 5e, use negative values of Shift Pitch Conversion in UVR Advanced MDX-Net settings (explained more thoroughly above). “tried it on a regular model (bigBeta5e) - the spectrogram looks a little more cut off at the high end than without the pitch adjust and overall the vocal sounds a little rounder and not quite as harsh (so the transients are not so nuclear)” - cristouk

70. Fixing missing sound after separation of multistem models

With certain at least 4 stem models, you might find out that the inversion of a mixdown of those 4 stems vs original mixture is different. So you might get an additional 5th stem that way - your own “other”. It might be useful if some instruments got missed, or simply for remastering purposes where not having any missed bits of audio is critical for your work.

71. Start separation in a different place of the song

Cut it manually. The result might resemble changing chunks setting a bit.

72. Use instrumental model result as pre-processor for vocal model

One of suggested RVC workflows

Get VIP models (optional donation)

<https://www.buymeacoffee.com/uvr5/vip-model-download-instructions>

If you still see some missing models in UVR5 GUI, which are mentioned in this document, get them from download center (or [here](#), expansion pack) and click refresh in model list if you don't see some models.

SDR leaderboard

Tested on multisong dataset

https://mvsep.com/quality_checker/leaderboard2.php?&sort=instrum

(some models/AIs/methods are not public, or only on MVSEP, all others you will find in UVR's and/or download center if you can't find some models, some only after using VIP code, or somewhere in this doc if it's public)

Older, “synth” dataset more of older models, a bit less reliable, no longer updated leaderboard by the results of new models

https://mvsep.com/quality_checker/leaderboard.php?sort=insrum

The biggest SDR doesn't automatically have to mean that your result will be the best for your song, and your use case (inst/voc/stem). Read the [list](#) of all the best models and methods at the top, and experiment.

Apart from bleedless/fullness metric, models with bigger SDR than others might pick up instruments better (e.g. less wind instruments recognized as voice).

Also, "The way I see high SDR is it indicates the lower frequencies will be more accurate to the original stem, and be more free of distortion or noise. And I also see it sometimes indicates better quality of fundamental frequencies (closer to the original gain/phase, more consistent separation), but I don't know much beyond that lol" - stephanie

- For specific songs, different ensemble configurations can give better results than for others.
- "Since the SDR [on MVSEP's] synth dataset is flawed from the get-go due to the dataset being used isn't really music, but sample-based, don't get your hopes up too much.". But it generally reflects in greater extent differences between models, e.g. used in Demixing Challenge 2021, so it's not totally bad and multisong dataset might be even better (and still not perfect) - just be aware that different settings can give you better results for your particular song rather than average best combination of models on the SDR chart.
- Bas Curtiz conducted some tests with commercial music as evaluation dataset, and it turned out that only models already close in SDR switched ranks, and most models kept the same. So in [conclusion](#), multisong dataset can be considered as still reliable (although bleedless and fullness metric is more suitable for our tasks now - more below).

About SDR evaluation on MVSEP and how important factor is that to the final result

It still depends on the specific song, what bag of models/ensemble or what specific models will come out the best in specific scenarios. Suggesting by SDR of at least multisong dataset can be misleading. For example, the metric doesn't really reflect the differences between e.g. HQ_3 and MDX23C fullband model in case of bleeding in instrumentals occurring in lots of contemporary songs. Although, the bleeding issue doesn't always occur, and then, HQ_3 results can be more muffled, so in this case, SDR metric would be more accurate to human listening scenario where MDX23C models gets better metric, so it can be misleading, because SDR can vary very much from song to song.

"The thing is that SDR evaluates at the same time how "full" the stem separation is and how much bleed there is in the separated stem. You can't know, only based on SDR score, which of "fullness" or "bleedless" is impacting the score the more" - jarredou

Also, according to some SDR evaluations conducted by Bas Curtiz, it turned out that permanent bleeding don't have more impact on SDR than occasional bursts of bleeding here and there.

Still, in some scenarios SDR metric of multisong dataset on MVSEP can be a safe approach, giving you some reassurance that the result in a strict test scenario will be at least decent in some respects, although you can (or even should when some instruments are missing) still experiment trying to get a better result, but it doesn't have to be reflected in SDR.

To sum up, SDR evaluation is only kind of averaging toward a specific dataset of songs, and it's unpredictable based on just SDR how certain model will behave on specific song, plus its algorithm is limited vs human ears too. For example, if you could measure SDR for a specific song by its official, perfectly inverting instrumental, then it may not get the best result by the settings of the best ensemble combination measured by SDR for the time being. Suggesting

by SDR means there's just higher chance to hit a good result in a certain spectrum of sonic changes - it's a good starting point to experiment further.

Based on 9.7 NET 1 models, MVSEP synth dataset usually gives ~0.7 higher scores than on [Demixing Challange 2021 leaderboard](#). Also, it favours Bas Curtiz FT model more than multisong dataset due to some characteristic features ZFTurbo pointed out.
“A calculation by a computer isn't a human ear”.

- Another way to at least sonically evaluate a model/ensemble, is to test it on a set of [AI killing tracks](#) which tend to have specific issues after separation with most if not all models, and to see how better or worse it got. Childish Gambino – Algorhythm is a good starting point to chase differences in vocal bleeding in instrumentals among various models, due to specific effects applied to vocals.

How does SDR even work in Python

```
def sdr(reference, estimate):
    delta = 1e-7 # avoid numerical errors
    num = np.sum(np.square(reference), axis=(1, 2))
    den = np.sum(np.square(reference - estimate), axis=(1, 2))
    num += delta
    den += delta
    return 10 * np.log10(num / den)
```

Q: Is there a way to compare SDR between an official instrumental and the filtered instrumental

A: Bas has shared an [.exe](#) script to do that easily ([uvr-general](#))

What you could do, but only if u have the original vocal or instrumental, is to check on SDR with this:

usage:

```
sdrcalc.exe "c:\your-input-folder" "c:\your-output-folder"
make sure they have the exact same extension + filename
```

“Here is an idea for multisong leaderboard V2, with the songs edited to have the loudness of real music. In this paper, they show that lots of models SDR value decrease when evaluated on real music <https://arxiv.org/pdf/2208.14355>”

After the community used SDR on synth, and later multisong dataset extensively, later jarredou invented a new method of automated evaluation of models:

Bleedness and fullness leaderboard

Python evaluation script by jarredou (prob. [mirror](#)), [Torch version](#) (with Bas Curtiz), used on [Quality Checker](#)

Librosa version added to ZFTurbo training repo.

More detailed and reliable method of evaluation on multisong dataset than SDR.

(old) Bas Curtiz' evaluation chart with some Roformers tested with that method:

https://docs.google.com/spreadsheets/d/1pPEJpu4tZjTkjPh_F5YitlyHq8v0SxLnBydfUBUNlBl/edit?usp=sharing ([shortened version](#) - it's outdated - all metrics are rewritten in the models sections [above](#))

For some newer models not on the list, you could search [here](#) for the model name, and bleedless/fullness metrics for new models are now provided in the evaluation description when you click on the result, but plenty of model evaluations have names not corresponding to final model names and were shared along with models on our Discord and later pasted to this document above.

Also, sorting by specific metric on MVSEP was added in June 2025, so you can track the exact evaluation by provided metrics in this document that way, or by searching Discord, but it can be difficult, as links to some older models' evaluations were not indexed by the metrics bot, so once the evaluation was posted, the bot wasn't showing metrics from the beginning, and not all models were evaluated along with the model release.

Explanations on the metric

Spectrogram difference showcase [diagram](#)

“Blue is what is missing from separated stem (compared to clean source).

Red is bleed in separated stem.

White is perfect

(dB scale on right seems wrong, I haven't checked, but it's not really important to see what is going on).

Same formula [can] be used for a metric, which would theoretically measure bleedness and fullness of the evaluated models

I think that for a metric, it's better to then separate negative values of diff array on one side, and keep positive values on other side, and average/scale each of them separately, so we get 2 scores, 1 for bleedness and 1 for fullness.

It has to be experimented further (and with better stft, it's only working on single chunk currently)

(Not sure that so high n_fft/mel_bins values are really needed, it was just nicer on the plot with that”)

“bleedless/fullness metrics are stft magnitude-only based and as they are discarding the phase data, they have some kind of blind spots.” - jarredou

Random noise added to results can increase fullness metric:

https://mvsep.com/quality_checker/entry/7709

https://mvsep.com/quality_checker/entry/7708

"I freq, the simplest way to explain it - it's a mix between fullness and bleedless but without the noise issue (in a sense it's the real fullness/bleedless metric) (...) there's no universal metric still sadly, we have to rely on a combination of them (and our ears)"

"-I1_freq = bleedless (higher is cleaner)

-aura_mrstft = fullness (higher is fuller)

they maybe don't have the issues fullness and bleedless have but I haven't played to check that" becruily

Read for [discussion](#)

"[The] problem with bleedless/fullness metric is that you can easily increase them by multiplying stem on constant.

Multiply predictions by 0.97 - it increases fullness and reduces bleedless

Multiply predictions by 1.03 - it greatly increases bleedless and reduces fullness" - [ZFTurbo \(metrics/discussion\)](#)

Other metrics:

Log WMSE - good "at least for drums or anything rich in low frequency content" - jarredou
"It is a relatively new time-domain metric over SDR and SI-SDR that is not overly sensitive to low frequencies like SDR and can accurately evaluate silent intervals.

In addition, time-domain metrics can be evaluated for both amplitude and phase." - Unwa
Metrics ignore phase, so probably phase fixer won't affect fullness/bleeedless metric.

For evaluating specific instrument stems, interesting read:

<https://arxiv.org/abs/2507.06917v2>

Top metrics of publicly available Roformers for instrumentals available for download

(as for 18.06.25)

*Instrumental models sorted by instrumental **fullness** metric:*

INSTV6N (41.68)>inst_Fv4Noise (40.40)/INSTV7N (no metrics)/Inst V1e (38.87)>Inst Fv3 (38.71).

While V1e+ (37.89) might be already muddy in some cases

*Instrumental models sorted by instrumental **bleedless** metric:*

Gabox inst_fv7b

Fullness: 27.07 (worse than most vocal Mel-Roformers later below)

Bleedless: 47.49

Inst_GaboxFv7z

Fullness: 29.38

Bleedless: 44.95

Unwa BS-Roformer-Inst-FNO

Fullness: 32.03

Bleedless: 42.87

Unwa v2

Fullness: 31.85

Bleedless: 41.73

Inst_gaboxBv3

Fullness: 32.13

Bleedless: 41.69

Inst_GaboxFv8 (its replaced v2 variant)

Fullness: 33.22

Bleedless: 40.71

Becruilly inst

Fullness: 33.98

Bleedless: 40.49

Gabox instv7plus

Fullness: 29.83

Bleedless: 39.36

Unwa HyperACE

Fullness: 36.91

Bleedless: 38.77

Unwa v1

Fullness: 35.69

Bleedless: 37.59

Gabox fv3

Fullness: 38.71

Bleedless: 35.62

Unwa v1e

Fullness: 38.87

Bleedless: 35.59

Gabox fv5

Fullness: 39.40

Bleedless: 33.49

Vocal models/ensembles sorted by instrumental **bleedless** metric:

(more muddy; Gabox and Unwa's Revive models not evaluated yet):

[Descriptions](#) of the public models

MVSep BS-Roformer (2025.07.20) - the 2 previous versions got replaced on the site by it

Inst. Fullness 27.83

Inst. Bleedless 49.12

MVSep Ensemble 11.50 (2024.12.20)

Inst. Fullness 27.17

Inst. Bleedless 47.94

MVSep Ensemble (4 stem) 11.93 (2025.06.30)

Inst. Fullness 28.70

Inst. Bleedless 47.68

MVSep MelBand Roformer (2024.10)

Inst. Fullness 27.73

Inst. Bleedless 47.48

BS-RoFormer SW 6 stem (MVSEP/Colab/undef13 splifft)

Inst. Fullness 27.45

Inst. Bleedless 47.41

(use inversion from vocals and not mixed stems for better instrumental metrics)

MDX23 Colab fork v2.5 by jarredou

Inst. Fullness 28.02

Inst. Bleedless 47.24

(more noticeable bleeding/noise than MVSep Ensemble above)

voc_fv4

xx

xx

(Good if you need less vocal residues than typical instrumental Roformers (even less than Mel Kim, FT2 Bleedless, or Beta 6X - makidanyee).

MelBand Roformer Kim

Inst. Fullness 27.44

Inst. Bleedless 46.56

Kim | FT2 Bleedless (by Unwa)

Inst. Fullness 27.78

Inst. Bleedless 46.31

Beta 5e (by unwa)

Inst. Fullness 27.63 (bigger metric than Kim)

Inst. Bleedless 45.90

Kim | FT 2 (by unwa)

Inst. Fullness 28.36

Inst. Bleedless 45.58

Kim | FT (by unwa)

Inst. Fullness 29.18

Inst. Bleedless 45.36

MVSEP BS Roformer (2025.06)

Inst. fullness: 17.30

Inst. bleedless: 37.83

(can be still a good choice in case of some crossbleeding, vocal chops, or residues of reverbs or BGV)

MVSEP Ensemble 11.93 (also contains 2025.06)

Inst. fullness: 17.73

Inst. bleedless: 36.30

—

Outperformed vocal models for instrumental bleedless

(still metrics for instrumental stem, so after inversion if not duality)

SYHFT V3 (by SYH99999)

Fullness 28.07

Bleedless 45.15

Duality v1 (by unwa)

Fullness 29.08
Bleedless 43.26

Duality v2 (by unwa)
Fullness 28.03
Bleedless 44.16

Mel Becriuly vocal
Fullness 28.25
Bleedless 40.95

SYHFT V2.5 (by SYH99999)
Fullness 28.60
Bleedless 40.34

Big SYHFT V1 (by SYH99999)
Fullness 28.48
Bleedless 44.81

Unwa beta 4
Fullness 26.29
Bleedless 44.71

SYHFT V4 and V5 were never publicly released

bleedless+fullness/2=avg
experimental avg metric for vocals (favours bleedless metric)

Bas' Edition - 27.72
FT2 bleedless - 27.54 | 2,49
24.10 - 27.44 | 2,21
FT2 - 26.84 | 2,23
FT - 26.58
5e - 26.42 | 1,54
voc_gabox - 26.38
voc_fv2 - 26.36
voc_fv3 - 26.06
Becriuly - 25.99
beta 4 - 25.93
FullnessVocalModel - 25.91
voc_fv4 - 25.02

Other ensembles for UVR5

Best newer ensembles on the list at the [top](#) of the doc. Older configurations follow after the listed hidden results below.

For reference, read MVSEP's [SDR evaluation chart](#) (UVR ensembles will appear later in the chart).

Be aware that some of the results on the chart above at the top are not from UVR5 or use different methods and code to achieve better results and might be not public/still WiP, e.g. the following:

Hidden leaderboard results (all SDR results provided for instrumentals, Discord links below are dead, but at least some can be found by the search on Discord and by verifying the opened link address which initial URL hasn't changed):

- Bas' unreleased fullband vocal model epoch 299 + voc_ft - SDR [16.32](#))
- [this](#) older viperx' unreleased custom weights code (newer one is up already), besides, "instrumental vX" entries are his ones (it rather utilizes public models with his own non-public weighted inference, and he gatekeeps it for more than since MDX23 results were published). BTW. ebright is probably the 2nd place in MDX23, at least the result appeared in similar time like ByteDance. 2nd place decided not to publish their work.
- [32-bit](#) higher SDR result of original multisong dataset uploaded as output (opposed to the previous 16-bit currently on top). "Multisong dataset | Original stems | bass/drums/other joined" is not a model!

- Bytedance v.0.2 - inst. SDR [17.26](#), now it's outperformed by v.0.3 and is [17.28](#), now called 1.0),
- "MSS" - is probably ByteDance 2.0, not [multi source stable diffusion](#), as BD's test files which were published were starting with MSS name before, but the first doesn't necessarily contradict the latter, although they said to use novel arch - SDR [18.13](#), and probably another one by ByteDance - SDR [18.75](#), let's call it 2.1, but seeing inconsistent vocal result vs previous one here, we have some suspicions that the result was manipulated at least for vocals (or stems were given from different model).
- Ripple app/SAMI-Bytedance on the chart is 16.59, also input files weren't lossless.
- BS-Roformer results by viperx posted in [Training](#)

BTW. model_mel_band_roformer_ep_617_sdr_11.5882 is Bas Curtiz model trained purely on multisong dataset as an experiment, and won't give good results outside multisong dataset.

mel_band_roformer_ep_125_sdr_11.2069 is Bas Curtiz fine-tune model trained from ZFTurbo checkpoint, and it was shared with him under condition it will remain non-public/MVSEP exclusive.

Some of these models in the download center are visible after using the [VIP code](#).

Older the best ensembles for UVR by SDR :
(some newer/better ones than these located at the top of the doc)

For 28.07.23

Kim Vocal 2 + MDX23C_D1581 + Inst HQ3 + Voc FT | Avg/Avg

For 28.07.23 (#4563)

Kim Vocal 1 + Kim Vocal 2 + MDX23C_D1581 + Inst HQ3 + Voc FT + htdemucs_ft | Avg/Avg

For 27.07.23 (#4561)

Kim Vocal 1 + Kim Vocal 2 + Kim Inst + MDX23C_D1581 + Inst HQ3 + Voc FT +

htdemucs_ft | Avg/Avg (beta UVR)

For 24.06.23 (#3842)

Kim Vocal 1 + 2 + Kim Inst + HQ3 + Voc FT + htdemucs_ft | Avg/Avg | Chunks: [ON](#)

(but for ensembles instead of single models it can score better with chunks disabled)

[Consider using MDX23C_D1581 vocal model above as well, if ensemble in this arch works correctly, if not, perform manual ensemble, not sure here)

As for the very big ensemble from older synth leaderboard (2023-04-30):

MDX-Net: 292, 496, 406, 427, Kim Vocal 1, Kim Inst + Demucs ft

Optionally, with later released models - voc_ft and Kim Vocal 2 -

It doesn't score too good SDR-wise on newer synth dataset, since it uses older models which have better counterparts already. Synth dataset is not used for evaluations for a long time.

For 13.06.23 (#3322)

Inst HQ2 + 427 + Inst Main + Kim Inst + Kim Vocal 1 + 2 + Demucs FT | Avg/Avg | Chunks Batch | Spectral inversion OFF

Most probably you can safely replace Inst HQ2 with HQ3 and 4 (better SDR) getting a slightly better SDR in ensemble (it's just not tested in ensemble yet).

But be aware that "The moment you introduce Instrumental models, there will be a bit of residue in the vocal output.

However, the SDR scores higher.

I'd say go with Vocal models only, if you care about your vocal output."

The same is vice versa for instrumentals.

- *Older ensemble configurations or custom settings with lower SDR*

(but might be useful for some specific songs or genres if further info is given)

From public models, the best SDR on 14.04.23:

Ensemble | Kim vocal 1 + Inst HQ 2 + Main 427 + htdemucs_ft | Avg/Avg | Chunks Batch | Denoise Output ON | Spectral Inversion OFF | WAV
For instrumentals

And

Ensemble | Kim vocal 1 + Inst 3 + Inst HQ 2 + Inst Main + htdemucs_ft | Avg/Avg | Chunks Batch | Denoise Output ON | Spectral Inversion OFF | WAV
For vocals

As of 01.01.23 the best SDR for vocals/instrumentals has:

-UVR-MDX-NET INST MAIN + UVR-MDX-NET Inst 3` + `kim vocal model fine tuned (old)` + `Demucs: v4 | htdemucs_ft - Shifts: 2 - Ensemble Algorithm: Avg/Avg`, chunk margin: 44100 (better SDR compared to 22050), denoise output on (-||- off), spectral inversion off (-||- on)

- MDX-Net: Kim vocal model fine-tuned (old) + UVR-MDX-NET_Main_427 + Demucs: v4 | htdemucs_ft - Ensemble Algorithm: Avg/Avg, Volume Compensation: Auto (it sets `1.035` - the best for Kim (old) model vs other options)
Shifts: 10 - Overlap: 0.25

- a bit worse ensemble settings than both ensemble settings above SDR-wise:

UVR-MDX-NET Inst 3 (464) and “UVR-MDX-NET_Main_438” vocal model (main) and htdemucs_ft - Ensemble Algorithm: Average/Average

- Also good combo (for instrumentals, vocals in half of the cases):

MDX-Net: UVR-MDX-NET Inst Main

VR Arc: 7_HP2_UVR

Demucs: v4 | htdemucs_ft

Max Spec/Max Spec

- UVR-MDX-NET Inst 3 as a main model and 7_HP2-UVR as a secondary with the scale set to 75%

(Anjok 21.12.22: Personally, I found that using [it] produces the cleanest instrumental." "It means the final track will be 25% hp2 model and 75% inst 3 (similar to ensemble feature, but you have more control over how strong you want the secondary model to be)"

- MDX-NET inst3 model (464) with secondary model 9_HP2_UVR 71% (hendrysetiadi: seems to get the best results with e.g. disco songs).

- Inst Main + 427 + Net 1 (CyPha-SaRin: was a pretty good combo. One big model, one medium, one small, pretty decent results across the board. If a song going to have problematic parts, it's going to have regardless of what combo you picked, it seems.)
- kim vocal 1 + instr 3 + full 403 + inst HQ 1 + full 292 + instr main with MAX/MAX (hendrysetiadi: i think that's the best combination of ensemble that i found)
- For Rock/Metal - The MDX-Net/VR Architecture ensemble with the Noise Reduction set between 5-10 (depending on the track) and Aggression to 10.
- For Pop - The MDX-Net/VR Architecture ensemble with the Noise Reduction set between 0-4 and Aggression to 10. (Anjok, 13.05.22)
- Here is another ensemble that I have tried myself "VR Arc: 1_HP-UVR x MDX-Net: Kim Vocal 1 x MDX-Net: UVR-MDX-NET: Inst HQ 1 x MDX-Net: UVR-MDX-NET: Inst HQ 2" All with the average/average ensemble (Mikey/K-Pop Filters)
- Inst HQ 1 & Main 427 are best for India
- VR: 7_HP2-UVR, MDX: Kim vocal 1, Inst 3, Inst Main, Main, htdemucs_ft
Max/Max, main pair: vocals/instrumental

"Instrumentals sound so good using these settings also. I can't believe this is possible. What an amazing software. Thank you to whoever made this." StepsFan

- I got an ensemble that works well for loud and crazy tracks (this instance it's dariacore lol) - by knock:

Models: Inst HQ 3, Main, Voc FT
 Ensemble Algorithm: Avg/Avg
 MDX-Net settings:
 Vol Comp: Auto
 Segment Size: 4096 (you can go up to 6144 if you want to wait longer, 4096 has seemed to be perfect for me)
 Overlap: Default (which I believe is 0.5)
 Shift Conversion Pitch: -6 (semitones)
 Match Freq Cut-off: Off
 Denoise Output: Yes
 Spectral Inversion: No

Mateus Contini's methods

#1 (old)

- "TIP! For busy songs: I was testing some ensembles trying to get Instrumental Stems with less volume variation (muddy), preserving guitar solos, pads the most and I had great results doing the following, for anyone interested:

Ensemble (Demucs + 5_HP-Karaoke with Max for Instrumental stem) - The result will be the Instruments + Backing Vocals and this preserves most of the guitar solos, pads and things that MDX struggles.

Instrumental Stem Output > Demucs to remove the Backing Vocals from the track - This pass will remove the rest of the Vocals. In some cases will be some minor leftovers that you can clean later with other methods.

I find the results better than Demucs alone/ MDX models or other ensembles for what I'm looking for. I'm not evaluating noise, but fuller instrumental Stems, trying to preserve most of it and also the cost (time) to do it.

Since I'm not interested, for this case, in doing manual work song by song and just use these stems to sing over it, I find the results great." - Mateus Contini

Q: Do you mean that you process Demucs 2 times? Once for ensemble with VR then the result was processed using Demucs again?

A: You can add other models with the ensemble, like Demucs, VR_5-Karaoke and HQ3 for an extra, before processing again with Demucs.

Also, this method is very good for leave good backing vocals into the instrumentals (only the ensemble result). I find extracting bv from the Vocal Stem to be less effective, giving you less material (comparing if you would join the bv with instrumentals later)

M.Contini Method #2 (newer)

Well, I tried to improve the results of the method I posted, so here it is, for **anyone interested in get fuller Instrumentals**, with a bit of bleed in some songs, yielding great results overall.

I'm doing this in the UVR-gui. The idea behind it is to scoop the vocals little by little, so the instrumentals is preserved the most. The process requires 3 extractions. Here are the Ensembles:

1. pass Ensemble: 5_HP-Karaoke-UVR + Inst HQ3 + htde mucs - Min/ Max

- If the song doesn't have BV, this will already give you good Instrumental Stem results. If you have Vocals bleeding into the Instr, continue to pass 2, but sometimes jumping straight to pass3 will produce better results.

- If the song have BV, this you keep a fuller **Instrumental Stem with BV** in it.

If you want to keep the BV, but there is some Main Vocals bleeding through the Instr, continue to pass 2.

2. pass Ensemble: Kim Vocal 2 + Inst HQ3 + MDX Karaoke 2 - Min/Max

- This pass will try to preserve the BV in the Instrumental Stem while removing Main Vocal bleed. You can stop here if you want the **Instrumental Stem with BV**

3. pass Ensemble: Kim Inst + Inst HQ3 + htdemucs - Min/Max

- This pass will try to remove BV from the instrumental Stem and other Main Vocal Bleed while keep the Instrumental fuller.

The idea behind it, is to have less volume variation where the vocals are extracted, leaving the Instrumental Stem less muddy. Since the extraction of the vocals is done little by little using the Min/Max, the Models will not be so aggressive. This is a great starting point if you want to improve further in a DAW or just sing over it. The Con is that, sometimes, the track will have tiny bleeds. If you try this method, please post the results here.

#3

- -try this ensemble: 9_HP (10 agression) + HQ3 (chunks on) + demucs_ft, Min/Max
- it preserves most of the instruments.

M. Contini method #4 (new)

Another Ensemble suggestion for good instrumentals with minimized bleeding vocals and a bit of noise in some cases:

Ensemble: 9_HP + HQ3 + Demucs_6s (secondary model 50%: full_292) - Algorithm [min/max]

Configs:

9_HP Window[512], Agress[10], TTA[on], Post[off], High-End [off])
HQ3 Chunks[on] [auto], Denoise[on], Spectral[off]
Demucs_6s Chunks[on] [auto], Split[off], Combine[off], Spectral[off], Mixer[off], Secondary Model - Vocals/Instr [MDX-Inst_full_292] [50%]

Why Demucs_6s and not _ft - I compare them in some songs and 6s have less vocal bleed in the instrumental track.

Description:

The idea is to take the good bits of the models using only one from each Group (VR, MDX and Demucs). The secondary model on Demucs is to minimize some vocal bleeding with sustained notes that was happening in some songs.

Comparing the results from multiple models, I find that Chunks enabled on MDX and Demucs removes some bleeding vocals from the Instrumental track and gives better results overall. This ensemble in my machine completes in about 5 min per song (GTX 1070 8GB, 16GB RAM, Ryzen 1600x). [chunks have been replaced by newer method in newer UVR GUI versions]

-
- "The best combo is the HQ instrument models ensemble average/average including HQ3/Main/Main Inst/Kim1/2/Kim Inst/demucs3 (mdx_extra)/htdemucs_ft/htdemucs6s" (MohammedMehdiTBER)

"Wow, I tried out the ensemble with all those models you said, and it actually sounds pretty good. There's a definitely more vocal bleed but in a saturated/detailed distortion type of way. I can't tell which one I like better, the ensemble sounds more full and has more detailed frequencies, but the vocal bleed is a lot more obvious. The HQ_3 by itself has almost no vocal bleed but sounds more thin and watery."

- Kim instr + mdx net instr3 + HQ2 + HQ3 + voc ft max/max

The result is so amazing... Now can hear more detail on instrumental result where before I cannot hear a bit of music parts. (Henry)

- "I am very much enjoying making an ensemble of HQ3 and MDX23C_D1581, then inverting the vocals into the instrumental and running that through hq3 with 0.5 overlap" (Rosé)
-

Ensembles for specific genres

Evaluation based on public models available at 23.04.23 and multisong dataset on MVSEP. The list might be outdated, as it doesn't take all the current models into account.

SDR sorted by genre

By Bas Curtiz

"If we remove **Kim vocal 2**, so only those that are available right now will be taken into account:

- Ensemble Rating 1 scores highest on average overall

[Probably this one:

[Kim vocal 2 + Kim FT other + Inst Main + 406 + 427 + htdemucs_ft | Avg/Avg](#)

At least it was the best for the given date.

But now we have ensembles which score better.]

- Kim vocal 1 is best for Rock

- Kim vocal 1 & Ensemble Rating 1 are best for RnB/Latin/Soul/Funk

- MDX'23 Best Model is best for Pop
- Main 427 & MDX'23 Best Model are best for Other
- Main 427 & MDX'23 Best Model are best for Blues/Country
- Main 427 & Ensemble Rating 1 are best for Jazz
- Main 427 & Ensemble Rating 1 are best for Acoustic genres
- Ensemble Rating 1 is best for Beats
- Ensemble Rating 1 is best for Hip Hop
- Ensemble Rating 1 is best for House

Sheet where **Kim vocal 2 **is removed:

<https://docs.google.com/spreadsheets/d/1ceXA7XKmECwnsQvs7a0S81XZOUokIXUN8ndsUDcYRcc/edit?usp=sharing>

Further single MDX-UVR models descriptions

E.g. used for ensembles above, but if a model has a cutoff, using ensemble with models/AIs without cutoff like Demucs 2-4 will fill the gap above. But it's still a good alternative for people without decent Nvidia GPUs or are forced to use Colab.

UVR-MDX models naming scheme

All models called "main" are vocal models.

All models called "inst" and "inst main" are instrumentals.

NET-X [9.X/9.XXX in Colab] are vocal models

Kim vocal 1/2 (self-explanatory)

Inst main is 496

Kim other ft is Kim inst

Model labelled as just 'main' is vocal, and was reported to have the same checksums as 427 and 423, but it doesn't seem to be true as 427 and main have different SDR (427 has better SDR than main, so apparently main is 423 [CRC32: E3C998A6]).

- MDX HQ_1/2 models - excellent, vivid snares, no cutoff (22kHz) high quality, rarely worse results than narrowband inst1-3 models, HQ_2 might have slightly less loud snares, but can have fewer problems with removing some vocals from instrumentals

- MDX-UVR Inst 3 model (464) - 17.7 cutoff (the same cutoff as for Inst 1, 2 inst main, but maybe not applicable for vocals after inversion in Colab), it was the third-best single model in our [SDR chart](#) at the time, available in Colab [update](#) and [UVR5 GUI](#) with [VIP models package](#) - now available for free.

- Forth-best single model for instrumentals back then was inst main (496, MDX 2.1), then inst 1 and inst2.

- There was some confusion about MDX 2.1 model (iirc on x-minus) being vocal 438 (even 411), but it's currently inst main.

- Full band MDX-Net models without cutoff (better SDR than Demucs 4 ft)

As for SDR, the epochs score is following: 292<403<386<(inst 1)<338<382<309<337<450 (first final, HQ_1)<498 (HQ_2)<(inst3)<(Kim inst)<HQ_3<HQ_4

Epochs 292, 403 and 450 and newer are also in [Colab](#) (and in UVR5, older when VIP code is redeemed)

- (currently the best, maybe not single model, but custom ensemble, as for vocals) MDX23 in [MVSEP beta](#),

and in UVR5 - Kim vocal model -

It's a further trained MDX-UVR vocal model from their last epoch (probably UVR-MDX-NET Main). It's based on a higher n_fft scale which uses more resources.

Not always gives that good results for instrumental as SDR may suggest, and also more people shares that opinion [both Colab and UVR users, so it's not due to no cutoff in Colab]. In UVR5 generally for the best vocal result use vocal models, and for the best instrumental result use instrumental models or eventually 4 stem Demucs 4 ft.

"[Kim_Vocal_1] is an older model (November), than Kim uploaded at 2022-12-04 to"
https://mvsep.com/quality_checker/leaderboard.php?sort=insrum&ensemble=0

(steps below no longer necessary, the model is added to GUI and these are the same models)

You can download her (so-called "old") model from here (it still gets better results for vocals than inst 3 and main):

<https://drive.google.com/drive/folders/1exdP1CkpYHUUksaz-gApS-0O1EtB0S82?usp=sharing>

When you copy/paste the model in

`C:\Users\YOURUSERNAME\AppData\Local\Programs\Ultimate Vocal Remover\models\MDX_Net_Models` It asks you to configure, hit Yes.
Then change `n_fft to 7680`."

For instrumentals, it gets worse results, frequently with more bleeding, and UVR manually applies cutoff above training frequency to instrumentals after inversion, to avoid some noise and possibly bleeding. Colab version of Kim model doesn't have that cutoff, so instrumentals as a result of inversion have max 22kHz frequency (but UVR applies it to prevent some noise).

- (*generally outperformed by models above*) [MDX-UVR 9.7 vocal model](#) a.k.a.

UVR-MDX-NET 1 (instrumental is done by inversion, older model) - available in [Google Colab/mvsep](#) (here 24 bit for instrumentals)/[UVR5 GUI](#).

Compared to 9.682 NET 2 model, it might have better results on vocals, where 9.682 NET might have better results for instrumentals, but everything might still depend on a song. Generally, 9.7 model got better SDR both in Sony Demixing Challenge and on MVSEP. Generally, 438 vocal, or 464 inst_3 should give better results for instrumentals. 427 vocal model tends to give worse results for instrumentals than even this older 9.7/NET1 model.

More about MDX-UVR models -

If they don't have more vocal bleeding than GSEP, they're better in filtering more vocal leftovers which sometimes GSEP tend to leave (scratches, additional vocal sounding sounds, also so-called "cuts" [short multiple lo-fi vocal parts] which GSEP doesn't catch, but MDX-UVR does probably due to bigger dataset). But using single instrumental MDX-UVR models instead of ensemble will result in cut off of a training frequency (e.g. 17.7kHz or lower).

Also, MDX-UVR like GSEP may not have this weird constant "fuzz" which VR models tend to leave as vocal leftovers (but in other cases, 9.7 model can leave very audible vocal residues, so test out everything on this list, till you get the best result).

The 9.7 model (or currently newer models) is also good for cleaning inverts (e.g. when having lossy a cappella and regular song).

If you tested all the alternatives, and you stick to the MDX-UVR 9.7 for some song, and it doesn't have (too much) bleeding, to fine-tune the results you can try out two 9.6 models to check whether it's better for you than 9.7 in this specific case (they're available at least in HV Colab and UVR5 GUI).

Newer MDX-UVR 423 vocal model usually provides more audible leftovers than 9.7 model.

To further experiment with MDX-UVR results, and you're stuck with Colab, you can enable Demucs 2 model on Colab to "ensemble" it with MDX-UVR model (although metrics say it slightly decreases SDR, I like what it does in hi-end - it was suspected at some point the SDR decreasing problems may come out from enabling chunking).

- [Demucs 4](#) (htdemucs_ft) - no cutoff, it's 4 stem, but you can perform mixdown without vocals in Audacity for instrumental - sometimes it may give you louder snare than in GSEP, but usually muffled shakers compared to GSEP. Also, it will give you more vocal residues than GSEP and MDX-UVR 464 (Inst 3). 6 stem models gives more vocal residues than 4 stem model (ft is the best one and also outperformed mdx_extra model [better than mdx_extra_q - quantized] but in some cases that might be worth to check old mdx_extra model as well (but

- (outperformed in many cases when used at least as a single models)
[VR-architecture models](#) (Colab, CLI or UVR5 GUI) sometimes provide cleaner and less muddy results for instrumentals than single narrowband models of MDX or even GSEP, only if they do not output too much vocal bleeding (which really happens for VR models

frequently - especially for heavily processed vocals in contemporary music), but bleeding also depends on specific model:

- E.g. *500m_1 (9_HP2-UVR)* and *MSB2 (7_HP2-UVR)* models are the most aggressive in filtering vocals among VR models, but other, less aggressive VR models may provide better sounding, less spoiled instrumentals (only if it is not paid for with worse vocal bleeding [BTW. I haven't heard the newest 2022 VR model yet (available at least in UVR5 GUI, maybe for Patreons, not sure)]).

All parameters and settings corresponding to specific models you'll find in "[VR architecture models settings](#)" section.

- [VR models-only ensemble settings](#) - if your track doesn't have too many problems with bleeding using VR-models above, to fine-tune the results achieved with VR, and to get rid of some mud, and e.g. get better sounding drums in the mix, I generally recommend VR-architecture models ensemble with settings I described in the linked section above. I'd say it's pretty universal, though the most time/resource-consuming method.

Also, [these](#) ensemble settings from the UVR HV Colab seem to make decent job for extracting vocals in some cases when above solutions failed (e.g. claps leftovers). Check also demucs_6s with 9 HP UVR and gsep in min-specs mode

Also, UVR5 GUI has rewritten MDX, so it can use their Demucs-UVR models from Demucs 3 (I think mvsep doesn't provide ensembling for any MDX models):

- (generally outperformed by MDX-UVR 4xx models) *Demucs-UVR models* - 1 and 2 models beside "bag" are worth trying out (mainly 1) on their own if the results achieved with above methods still provide too much bleeding - better results than e.g. bare MDX-UVR 9.7 or VR models or even GSEP in some specific cases (available on [MVSEP](#) and [UVR5 GUI](#)). They're Demucs 3, 2 stem better trained models by UVR team. No cutoff - 22kHz.

- As for extracting -

Karaoke / Backing Vocals

(more up-to date, but less descriptive list at the [top](#))

check MDX-UVR Karokee 2 model (available on MVSEP, UVR 5 GUI)

TL;DR - "Usually MDX B Karaoke has really good lead vocals and UVR Karaoke has really good backing vox"

"There are 3 good karaoke models (the ones I'm referring to are on mvsep.com [they seem to be no longer available there]). "MDX B (Karaoke)" seems to be the best at getting lead vocals from karaoke while "karokee_4band_v2_sn" (UVR) and

"HP_KAROKEE-MSB2-3BAND-3090" (UVR) seem to be best for backing vocals. I recommend using a mix of the 3 to get as many layers as possible, and then use Melodyne to extract layers as best as possible. Then combine the filter results and Melodyne and you should have smthn that sounds pretty good" karokee_4band_v2_sn model might be not compatible with Colab (check mvsep or UVR5 GUI)

- Demix Pro may do a better job in B.V. than models on x-minus.
Even than the new model on x-minus since 01.02.23, but might be worth trying out on some songs (the problem is probably bound to MDX architecture itself).

"MDX in its pure form is too aggressive and removes a lot of backing vocals. However, if we apply min_mag_k processing, the results become closer to Demix Pro"

- Medley Vox
(installation [tutorial](#))
For separating different voices, including harmonies or backing vocals check out this vocal separator, the demos sound quite good and Cyrus model has pretty similar results.
It's for already separated or original acapellas. Sometimes it gives better results than BVE models. Output sample rate is 24kHz, but it can be easily upscaled by AudioSR well.

Org. repository
<https://github.com/jeonchangbin49/medleyvox>

Old info (dead link):
https://media.discordapp.net/attachments/900904142669754399/1050444866464784384/Screenshot_81.jpg

How to get vocals stems by using specific models:

Song -> vocal model -> Voc & Inst
Vocal model -> Karaoke model -> Lead_Voc & Backing_Voc
Lead_Voc + Inst = Lead_Inst

- How to get backing vocals using x-minus
https://x-minus.pro/page/bv-isolation?locale=en_US
"Method two is terrible and I do not recommend it" - Aufr33

-If you have x-minus subscription, you can use chain mode for Karaoke as it currently gives the best results

How it probably works under the hood?
"On sitting down and reading
<https://discord.com/channels/708579735583588363/900904142669754399/1071599186350440540>

It's a multistep process where it mixes a little bit from MDX's split vocals and instruments. Then passes that mixture through the UVR v2 karaoke/backing vocals model. Then with those results, it inverts the separated lead vocal, and adds it to the instrumental result"

- As for **4 stem separation**, check GSEP or [Demucs 4](#) (now check better MDX23 Colab by jarredou)

(other stem is usually the best in GSEP, bass in Demucs 4, rest depends also on a song, and as for drums, if you further process them in DAW using plugins, then Demucs 4 is usually better as it's lossless and supports up to 32-bit float output).

Demucs 4 has also experimental **6 stem** feature. Guitar (can give good results) and piano (it's bad and worse than GSEP).

- As for free **electric guitar and piano stems**, currently GSEP and MVSEP models are the best, but paid Audioshake provides better results than GSEP. Also in GSEP "when the guitar model works (and it grabs the electric), the remaining 'other' stem often is a great way to hear acoustic guitar layers that are otherwise hidden.". LALAL.AI also has piano model and is "drastically" better than Demucs.

- From paid solutions for separating drums' sections, there are [FactorSynth](#), UnmixingStation, or free [Drumsep](#) (but rather use MDX23C model).

- As for specific sounds separation, check [Zero Shot Audio](#).

Cutoffs examination with spectrograms for various models and AIs, available in UVR5 GUI, along with examined times needed for each model to process on CPU or GPU (1700x/1080 Ti) by Bas Curtiz (cutoffs examination not applicable for MDX Colab where there is none unlike in UVR [it's to prevent noise]):

https://docs.google.com/spreadsheets/d/1R_pOURv8z9GmVkCt-x1wwApgAnpIM9SHiPO_ViHWI1Q/edit#gid=23473506

Spreadsheet of songs that use Vocals as a melody with snippets how they separate on various models/AIs

<http://vocalisolationtesting.x10.mx/>

In below sections you'll find more details, links, Colabs, all tools/AIs listed, more information about specific models as alternatives to experiment further (mostly MDX-UVR instrumental and vocal models available in UVR5 GUI and <https://x-minus.pro/ai> and MVSEP). I also provide some technicalities/troubleshooting everywhere when necessary.

Table of content

(click on an entry to be redirected to a specific section;
the section is outdated - check it in document outline instead if you can)

Last updates and news.....	1
General reading advice.....	30

Instrumental, vocal, stems separation & mastering guide

The best models

for specific stems

for instrumentals.....	31
for vocals.....	34
How to check whether a model in UVR5 GUI is vocal or instrumental?.....	39
for karaoke.....	39
for 4-6 stems (drums, bass, others, vocals + opt. guitar, piano):.....	43
SFX.....	45
De-reverb.....	46
Vinyl noise/white noise (or simply noise).....	50
Mixing and mastering.....	51
Audio upscalers list.....	52
More descriptions of models.....	53
MDX settings in UVR5 explained.....	57
Tips to enhance separation.....	63
Other ensembles in UVR5 - list.....	71
50 models sorted by SDR.....	87
Separating speakers in recording.....	93
General section of UVR5 GUI (MDX-Net, VR, Demucs 2-4, MDX23)	95
GUI FAQ & troubleshooting.....	96
Chunks may alter separation results.....	99
Q: Why I shouldn't use more than 4-5 models for UVR ensemble (in most cases).....	100
(older) UVR & x-minus.pro updates.....	101
MVSEP models from UVR5 GUI.....	107
Manual ensemble Colab for various AI/models.....	108
Joining frequencies from two models.....	109
DAW ensemble.....	110
Manual ensemble in UVR5 GUI of single models from e.g. Colabs.....	110
UVR's VR architecture models (settings and recommendations).....	110
VR Colab by HV.....	110
VR settings.....	111

VR models settings and list.....	113
VR ensemble settings.....	116
VR Colab troubleshooting.....	123
First vocal models trained by UVR for MDX-Net arch:.....	125
(the old) Google Colab by HV.....	126
Upd. by KoD & DtN & Crusty Crab & jarredou, HV (12.06.23).....	126
Other archs general section	
Demucs 3.....	134
Demucs 4 (+ Colab) (4, 6 stem).....	135
Gsep (2, 4, 5, 6 stem, karaoke).....	139
Dango.ai.....	144
MDX23 by ZFTurbo /w jarredou fork (2, 4 stems).....	145
KaraFan by Captain FLAM (2 stems).....	149
Ripple/SAMI-Bytedance/Volcengine/Capcut (Jianying)/BS-RoFormer (2-4 stem).....	152
Single percussion instruments separation (from drums stem).....	159
drumsep (free).....	159
FactorSynth.....	160
Regroover.....	161
UnMixingStation.....	161
VirtualDJ 2023/Stems 2.0 (kick, hi-hat).....	162
RipX DeepAudio (- -) (6 stems [piano, guitar]).....	162
Spectralayers 10.....	162
USS-Bytedance (any; esp. SFX).....	163
Zero Shot (any sample; esp. instruments).....	164
Medley Vox (different voices).....	165
About other services:.....	167
Spleeter.....	167
Izotope RX-8/9/10.....	167
moises.ai (3 EU/month).....	167
phonicmind.....	167
melody.ml.....	167
ByteDance.....	167
Real-time separation	
Serato.....	167
Stems 2.0.....	168
Acon Digital Remix.....	168
Misc	
FL Studio (Demucs).....	168
Fadr.com from SongtoStems.com.....	168
Apple Music Sing.....	168
Music to MIDI transcribers/converters.....	169

Piano2Notes.....	169
Audioshake.....	169
Lalal.ai.....	170
DeMIX Pro V3.....	171
Hit'n'Mix RipX DeepAudio.....	171
Moises.ai.....	172
How to remove artefacts from an inverted acapella? (can be outdated).....	174
Sources of FLACs for the best quality for separation process.....	175
Dolby Atmos ripping.....	184
AI mastering services.....	186
How to get the best quality on YouTube for your audio uploads.....	192
How to get the best quality from YouTube and Soundcloud - squeeze out the most from the music taken from YT for separation.....	193
Custom UVR models.....	195
Repository of other Colab notebooks.....	196
Google Colab troubleshooting (old).....	199
Repository of stems/multitracks from music - for creating your own dataset.....	200
List of cloud services with a lot of space.....	205
AI killing tracks - difficult songs to get instrumentals.....	211
Training models guides.....	215
Volume compensation for MDX models.....	229
UVR hashes decoded by Bas Curtiz.....	231
Local SDR testing script.....	233
Best ensemble finder for a song script.....	233

Models master list

50 models sorted by SDR

(from the public ones - so available to download and offline use)
(07.10.2024)

These are basically the top single models for now
(conventionally after these, additional vocal residues kick in, especially if not a vocal model)
Based on Multisong dataset evaluation on MVSEP chart with similar or the same parameters and inference if applicable.

Kim's Mel Roformer

model_bs_roformer_ep_317_sdr_12.9755
model_bs_roformer_ep_368_sdr_12.9628 (viperx/UVR beta)
BS-Roformer_LargeV1 (unwa's ft)
Unwa's Mel-Roformer Beta 3 (although for this day, SDR wasn't tested with the same parameters vs above, so it's based on assumption that unwa used the same parameters in synth dataset measurement)
Unwa's Mel-Roformer Beta 4
Unwa's Mel-Roformer Beta 5e
0) InstVoc MDX23C HQ (fullband a.k.a. 1648, 8K FFT)
0b) InstVoc MDX23C HQ 2 (fullband)
1) voc_ft
1b) UVR-MDX-NET HQ_4 (inst)
2) MDX23C_D1581 (a.k.a. narrowband)
3) Kim Vocal 2
4) Kim Vocal 1
5) UVR-MDX-NET_Main_427 (voc)
6) UVR-MDX-NET_Main_406 (voc)
7) UVR-MDX-NET HQ_5 (inst)
7) UVR-MDX-NET HQ_3 (inst)
8) UVR-MDX-NET_Main_438 (voc)
9) UVR-MDX-NET_Main_390 (voc)
10) Kim inst (a.k.a. other)
11) UVR-MDX-NET_Main_340 (voc)
12) Inst 3 (a.k.a. 464)
13) UVR-MDX-NET HQ_2 (inst)

(for vocal models, here start those with more vocal residues in instrumentals - can be still handy for specific songs)
+4 pos.

- 9) Inst Main (496)
- 10) Inst 2
- 11) UVR-MDX-NET HQ1
- 12) UVR-MDX-NET HQ 337 >382>338 epoch
- 13) Inst 1
- 14) HQ 386>403>292 epoch
- 15) UVR-MDX-NET2>NET3>NET1>9482 (NET3 a.k.a. 9.7)
- 16) htdemucs_ft (4 stem) (S 10/O 0.95)
- 17) htdemucs_mmi (4 stem)
- 18) htdemucs_6s (6 stem)
- 19) UVR-MDX-NET_Inst_82_beta
- 20) Demucs3 Model B (4 stem)
- 21) UVR-MDX-NET_Inst_187_beta

(dango.ai, Audioshake, Bandlab not evaluated)

Somewhere here, trash begins (excluding GSEP)

- 22) Moises.ai (probably before transition to newer Roformer models)
- 23) DeMIX Pro 4.1.0
- 24) Myxt (AudioShake 128kbps)
- 25) UVR-MDX-NET_Inst_90_beta
- 26) RipX DeepRemix 6.0.3
- 27) kuielab_b (4 stem) (MDX Model B from 2021 MDX Challenge)
- 28) kuielab_a (4 stem)
- 29) LALAL.AI
- 30) GSEP (6 stem) (although it sometimes gives much better results than its SDR)

VR arch

- 31) 7_HP2-UVR (a.k.a. HP2-MAIN-MSB2-3BAND-3090_arch-500m)
- 32) 3_HP-Vocal-UVR
- 33) 2_HP-UVR (HP-4BAND-V2_arch-124m)
- 34) 9_HP2-UVR (HP2-4BAND-3090_4band_arch-500m_1)
- 35) 1_HP-UVR (HP_4BAND_3090_arch-124m)
- 36) 8_HP2-UVR (HP2-4BAND-3090_4band_arch-500m_2)

- 37) 14_SP-UVR-4B-44100-2 (4 band beta 2)
- 38) 4_HP-Vocal-UVR
- 39) 13_SP-UVR-4B-44100-1 (4 band beta 1)

- 39) 15_SP-UVR-MID-44100-1
- 40) 16_SP-UVR-MID-44100-2
- 41) 14_HP-Vocal-UVR
- 42) VR | MGM_LOWEND_A_v4
- 43) 12_SP-UVR-3B-44100

- 44) Demucs 2 (4 stem)
(6 other old VR models proceed)

- 50) Spleeter 4 stems
- 51) Spleeter 2 stems
- 52) GSEP after mixdown from 4 stems separation

*Only instrumental models listed (outdated)
(4 stem and MDX23C models lies in all categories):*

Tier 1

MDX-Net models (trained by UVR team)

- 0) MDX23C HQ 1648 fullband
- 1) MDX23C HQ 2 fullband
- 1b) UVR-MDX-NET HQ_4 (inst)
- 2) MDX23C_D1581 narrowband
- 7) HQ3
- 10) Kim inst (other)
- 12) Inst 3
- 13) HQ2

Tier 2

+4 pos.

- 9) Inst Main (496)
- 10) Inst 2
- 11) HQ1
- 12) HQ 337 >382>338 epoch
- 13) Inst 1
- 14) HQ 386>403>292 epoch

Demucs 4

- 16) ht demucs_ft (S 10/O 0.95)
- 17) ht demucs_mmi
- 18) ht demucs_6s

20) Demucs 3 Model B (mdx_extra)

Tier 3

(somewhere between place 9-20 might be dango.ai, Audioshake, later maybe Bandlab)

- 22) Moises.ai
- 23) DeMIX Pro 4.1.0
- 24) Myxt (AudioShake 128kbps)
- 26) RipX DeepRemix 6.0.3
- 27) MDX-Net Model B from 2021 MDX Challenge (kuielab_b)
- 28) kuielab_a
- 29) LALAL.AI
- 30) GSEP (although it sometimes gives much better results than its SDR)

Tier 4

VR arch

- 31) 7_HP2-UVR (a.k.a. HP2-MAIN-MSB2-3BAND-3090_arch-500m)
- 33) 2_HP-UVR (HP-4BAND-V2_arch-124m)
- 34) 9_HP2-UVR (HP2-4BAND-3090_4band_arch-500m_1)
- 35) 1_HP-UVR (HP_4BAND_3090_arch-124m)
- 36) 8_HP2-UVR (HP2-4BAND-3090_4band_arch-500m_2)

Tier 5

- 37) 14_SP-UVR-4B-44100-2 (4 band beta 2)
- 38) 13_SP-UVR-4B-44100-1 (4 band beta 1)

Tier 6

- 39) 15_SP-UVR-MID-44100-1
- 40) 16_SP-UVR-MID-44100-2
- 42) VR | MGM_LOWEND_A_v4
- 43) 12_SP-UVR-3B-44100

- 44) Demucs 2
(6 other old VR models proceeds)

Tier 7

- 50) Spleeter 4 stems
- 51) Spleeter 2 stems
- 52) GSEP after mixdown from 4 stems separation

Differences by SDR divided for vocals and instrumentals are important to divide I think only in ensembles. In all other cases, if SDR is bigger for instrumentals in some model, it will be bigger for vocals vs the same model. At least only for ensembles there were so little differences that we had two top ensembles for both vocals and instrumentals.

Hall of fame

Great thanks to Anjok, Aufr33 (creators of UVR), KimberleyJSN a.k.a. Kim (model contributor and MDX/Roformers support), viperx (our former heavy user, supporter and now private models creator), tsurumeso (the creator of VR arch base code), BoskanDilan (creator of the old UVR GUI), IELab a.k.a Kuielab & Woosung Choi (MDX-Net arch creators), ZFTurbo (creator of MVSEP, MDX23, and many models), GAudio (GSEP creators),

Alexandre Deffosez a.k.a. Adefossez (Demucs creator), Bytedance with asriver (Roformer arch and Ripple app), lucidrains (for recreating the BS and Mel Roformer from released papers), jarredou (MDX23 fork, drumsep model, tons of support and work Colabs), Bas Curtiz (model trainer, insane amount of testing and UVR5 settings guidance, tutorials with SDR evaluating, models creator), Captain FLAM (KaraFan creator), unwa (for his Roformer fine-tunes and training advice), Gabox (-||-), becruily (model trainer, tons of advice), FoxyJoy (de-reverb, de-echo, denoise models), Not Eddy (UVR UI, Colabs, HF, KF fork), Sir Joseph (WebUI Colabs) - thanks to all of these people for the best freely available AI separation technologies and models so far, mesk (metal models and trianing guide).

Special thanks for users of our Discord:

HV (MDX and VR Colabs creator and UVR contributor), txmutt (Demucs Colab), CyberWaifu (lots of testing, some older Colabs), KoD/Mixmasher (first HV MDX Colab fork), dca100fb1 (a.k.a dca100fb8) (VR ppr bug, finding tons of UVR bugs and models testing and feedback), mesk (training guide and Roformers fine-tuning), Isling (lot of testing and suggestions), CyPha-SaRin (lots of models/UVR testing), BubbleG, Aspleas, Joe, santilli, RC, Matteoki (a.k.a. Albacore Tuna, our “upscale” guru), Syrkov, ryrycd, Mikeyyyyy/K-Kop Filters, Mr. Crusty crab (our mod; compensation values finding, MDX Colab mods and testing), knock (ZF’s MDX23 fine-tuning), A5 (lots of feedback on existing models), Infisrael (MDX23 guide and model testing), Pashahlis/ai_characters (WhisperX guide and script), Sam Hocking (our most valuable pro sound engineer and industry insider), Kubinka (for his Colabs and coding help), Vinctekan (one of our most valuable sound engineers and tools creator), CC Karaoke, “am able to use uvr with gpu”/vernignt (both lots of testing and advice), hendry.setiadi, raiboomdash (lots of model tests with vast descriptions), wancitte, essid64, makidanyee, - thanks to all of these people - for knowledge, help, testing and everyone whose advice, quotes and stuff appear in this doc. This guide wouldn’t be created without you. If I forgot someone, forgive me.

You can support UVR team by these links:

<https://www.buymeacoffee.com/uvr5/vip-model-download-instructions>

and

<https://boosty.to/uvr>

(subscription to <https://x-minus.pro/ai> to process some VIP models there online)

If you see duplicated models on the list in UVR5, click refresh.

X-minus FAQ

Q: how come level 1 will be eliminated? is it possible to leave it since i use this site very little and paying (2.79\$) per month is too much and anyway 360 minutes of audio per week is a lot. i do 5/ 6 per week. it is a waste of minutes.

A: If you renew your subscription several months in advance, you can use Level 1 even after removal. In addition, once your subscription Level 1 expires, you can use it for another month for free (after removing it in February).

Similarity/Phantom Center/Mid channel Extractor

“It extracts the phantom centre. I.e. what you hear as being in front of you in stereo audio”
“very useful for older mixes. Like 60s songs with hard panning”

- Dry Paint Dealer Undr (a.k.a. wesleyr36) released new Phantom Centre Models HTDemucs Similarity/Phantom Centre Extraction model:
[GDrive](#) / [HF](#) / [Colab](#) (it tends to be more “correct” in center extraction than the MDX23C model below)
That Demucs model won’t work with UVR giving bag_num error even with the yaml prepared in the same way as for Imagoy Drumsep and after renaming ckpt to th (it’s probably because it needs ZFTurbo inference code it was trained with).
- SCNet Similarity/Phantom Centre Extraction model by Dry Paint Dealer:
<https://drive.google.com/drive/folders/1CM0uKDf60vhYyYOCg2G1Ft4aAiK1sLwZ?usp=sharing>

VR6 models don’t work in UVR:

- iter41_l1_loss [model](#) for [VR v6.0.0b4](#) - similarity/phantom centre extractor by wesleyr36/drypaintdealerundr,
“I think this arch makes for a much better similarity extractor
(make sure to use the complex flag when running this model --complex or -X)”

Compared to MDX23C models below, VR6 ones were trained on limited dataset, but they can still perform better.

- 4096 [model](#) for [VR v6.0.0b4](#) by wesleyr36/drypaintdealerundr (it should perform better than the MDX23C 2048 model below)
“must be run not only with -X or --complex but also --n_fft 4096 --hop_length 2048 or -f 4096 -H 2048”
VS 2048 - “pros: less bleed
cons: less complete results as a similarity extractor
it seems to benefit from running the centre channel results back through the model for more complete results just like the original similarity extractor for more complete results although with the trade-off of more bleed
you end up with overall more bleed than the other model but with even more complete results”

Usage for [VR v6.0.0b4](#):

```
python inference.py -i path/to/an/audio/file --gpu 0 -P path/to/model.pth -X  
-f 4096 -H 2048 -o folder/you/wish/to/save/to
```

you can just drag a file/folder into the terminal/CMD to get the path too if that's more convenient

The command for Nvidia GPU, but CPU inferencing should be possible too.

- 2048 MDX23C [model](#) by wesleyr36/drypaintdealerundr

Can be used on MVSEP (in Experimental section) and x-minus.pro (option Extract backing vocals) or using ZFTurbo inference CML [code](#) (it doesn't work in the OG MDX23C inference code and in UVR).

"This model is similar to the Center Channel Extractor effect in Adobe Audition or Center Extract in iZotope RX [and Audacity/Bertom], but works better.

Although it does not isolate vocals, it can be useful." Aufr33

"The main thing I trained it for was to be used in a similarity extractor, since the original also used an AI model

The steps for that being:

1. Take the L channel from Audio_1 and the L channel from Audio_2 and merge them into a stereo file.

2. Run that through the model

3. Repeat for R channels

4. Merge the L and R channels back together, and you have the similarity, assuming the audio files were perfectly aligned."

It was trained in a period of 6 days on Quadro RTX 4000

"Some bits were better on the model, others on [the] Audacity's [Vocal and Center Isolation feature]"

- Melband Roformer Similarity/Phantom Centre Extraction [model](#) (beta) + lora by wesleyr36/drypaintdealerundr

"results are relatively clean but sound a bit filtered at times, comes with 2 lora checkpoints for frazer's [LoRA repo](#)"

- Mel-Roformer de-reverb by anvuew (a.k.a. v2/19.1729 SDR) | [DL](#) | [config](#) | [Colab](#) (it can serve also as a phantom center model, removing sides)

- Older VR model by HV Colab (Colab fixed 16.02.24)

<https://colab.research.google.com/drive/1WP5ljdutCc-RRsvfaFFIhnZadRhw-8ig?usp=sharing>

g

Don't forget to click cell with dependencies after mounting

If you want to use the repo locally, use just this [fix](#)

"If you have two language track it'll remove the vocals, but not its adlibs"

"It works like invert but instead of mixing the inverts together, it removes the difference and leaves the ones that sound the same"

It uses a specifically trained model on 100 pairs.

- Sadly, "It's like a downgrade of Audacity Vocal and Center Isolation feature" - it's muddier

Audacity can be used in browser at:

<https://wavacity.com/>

The option in version prior 3.5.0 is located in:

Effect>Special>Vocal Reduction and isolation

on 2.x: Effect>Vocal Reduction and isolation (at the very bottom)

3.5.0 or later: downloadable as Nyquist plugin from [here](#)

"Adobe Audition works a similar, but you can actually tweak a lot of settings. But the difference is pretty much non-existent. Or any better for that matter. Similar way. Even with Audacity, Adobe Audition, and PEEL [3d Audio Visualize], we are still not quite there yet. Currently, Audacity, and maybe Waves Stereo Center plugin have the best capabilities, but they are still aren't perfect." Vinctekan

Sadly, it turns out that all the three solutions can sound worse than current models for the use case of getting rid of dubbing in movies.

It can be used with window size 768 on CPU as well. Probably the lowest supported for GPU is 272 (352 was set, and 320 is possible too), but probably it won't change here much.

One of the use cases of Audacity method to get lead vocals (in 2021) was by obtaining e.g. main vocals from vocal or BVE model, and processing that stem with these settings:

Audacity>Effect>Vocal reduction and isolation>

on action, make it Isolate Center

Strength: 1.1 or 1.6

Then click OK. That effect must go on vocal part. If you use center isolation, low/high cut will be ignored

"Q: wouldn't it be possible to extract anything that is panned to a specific point yk like extract anything that is panned 100% exactly, or anything that is panned 80% or 50% etc, would that not be possible?

A: Mashtactic does that since almost 20 years now (coupled with dynamics and EQ filtering)

<https://www.youtube.com/watch?v=0IDAY0va4VE>

- zplane has released a clone recently, but it doesn't have the transient/sustain filtering iirc" (isling/jarredou)
- Also, you can use AudioSourceRE RePAN for center extraction as well (IncT)
- Or less complex free/paid Bertom Phantom Center (sometimes it's better, sometimes worse than MDX23C model).
"Bertom Audio claim to not use basic mid/side processing to extract the center so probably uses decorrelation on the sides instead of mid/side processing which by its nature negatively correlates them. The net result of Bertom is the sides are decorrelated from the center and so are maintained more strongly in the stereo field." Sam Hocking
- [AOM Stereo Imager D](#) "turn off auto-gain & turn down center"
- [Reaper guide](#)

Hints on using similarity models

Q: "Hi there! I have a question regarding audio separation in movies. I have an old movie with a stereo track in English, and a mono (!) track in French. I couldn't for the life of me find anything better than mono for my native language, which is a shame (even a source with supposedly stereo French is actually mono, you hear it right away).

So I'm willing to try and use the English track to reinject some stereo in the background, to widen the music and sfx at least (the voices being centered don't bother me). i.e. I could separate the voices from the rest in both languages, then mix the French voices with the English sfx and music. Whatever artifacts remaining could blend enough that it wouldn't be noticeable... Maybe...

How would you guys go about it? I've used UVR in the past but only on music, not movies – and it was months ago. Also my GPU is old (Nvidia GTX 970 with 4 GB VRAM) so this might be a limitation. Thanks for any advice you can give me!"

A: "If the French rip is a downmix between stereo channels, then you don't have to separate dialogue in the French rip. Only separate the vocals in the English version, encode it to mid/side, replace the mid channel with the French rip, decode back to stereo, and you're done. English vocal separation will get rid of the stereo bits from the side channel, so you won't be hearing both languages at once. Obv you have to align the sources too, which can be tricky if there are any changes in timing between the versions. If the French rip is only a single channel from source stereo, then do your way of isolating both languages.

Instead of using your own GPU you can send your audio file to [Colab](#) and use smth like melroformer v1e with default settings." (introC)

Q: What difference do you mean with "a downmix between stereo channels" and "a single channel from source stereo"? It seems the same thing to me but I may be missing sth obvious.

A: A single channel is either left or right, downmix is an average of left and right

Q: Then I guess what I have is a downmix?

A: Not sure, you need to check. I can check if you want, sent the English and mono audio. But basically, align the tracks, downmix the English track and check for null when mixing with the French mono track. If it isn't null, then it's not a downmix (or the audio tracks have other differences)

[OG](#) VR broken Colab by HV fixing history

It was fixed by adding these lines to it:

```
!apt-get install python3.8
!apt-get install python3.8-distutils
!apt-get install python3.8 pip
!python3.8 -m pip install librosa==0.9.1
!python3.8 -m pip install numpy==1.19.5
!python3.8 -m pip install numba==0.55.0
!python3.8 -m pip install tqdm
!python3.8 -m pip install torch==1.13.1
```

and renamed inference Colab line to python3.8

(not necessary)

```
! pip install soundfile=0.11.0
```

distutils was necessary to fix numpy wheel error, but regular 3.8 installed before was necessary for Colab to recognize !python3.8 commands. Because 3.8 was bare, it needed pip installed separately for this 3.8 installation. Then the rest of the necessary packages are installed for 3.8 - the old librosa fix, numpy for 3.8, and broken dependencies numba and tqdm. Then, the last torch working in HV Colabs was 1.13.1, 1.4 didn't work though it's compatible with 3.8. Maybe CUDA or generally upgraded Ubuntu problem. Can't tell. It was necessary anyway because Torch wasn't installed for 3.8.

Additionally, to fix the regular VR Colab, this line was necessary:

```
!python3.8 -m pip install opencv-python
```

And for some reason, I needed to install these with normal pip like below, and with python 3.8, so basically twice, otherwise it gave module not found

```
! pip install pathvalidate
```

```
! pip install yt_dlp
```

That all hassle with Python 3.8 is necessary because numpy on Colab got newer version, and newer ones no longer supports function used in HV Colabs, as they got deprecated.

Separating people in recording

Guide and script for [WhisperX](#) by Pashahlis/ai_characters

“A script on the AI hub discord for automatically separating specific voices from an audio file, like separating a main character's voice from an anime episode.

I massively updated this script now, and I am also posting it here now, since this discord is literally about that kinda stuff.

Script to automatically isolate specific voices from audio files

(e.g. isolating the main character's voice from an anime episode where many different characters are speaking).

After literal hours of work directing ChatGPT, fixing errors, etc, there is now a heavily updated and upgraded script available:

I encountered some transcription errors (musical notes, missing speaker or start and end times) that would result in the entire script failing to work. So the updated script now skips such audio. That is not a problem, however, as for a 22-min file it skipped only 16s of audio and the errored audio is just music or silence anyway.

It now also automatically merges all your audio files into one if you provide multiple, so that the speaker diarization remains consistent. This increases diarization time by quite a lot, but is necessary. The merged file will be temporarily saved as a .flac file, as .wav files have a maximum file size of 4gb. The resulting speaker files at the end of the script are created as .wav again, though, as it is unlikely they will reach 4gb in size.

I also added helpful messages that tell you at which state of the script it currently is at and which audio files it is processing at the start with the total length of audio being processed.

I also made sure that it saves the speaker files in the original stereo or mono and 16 bit or 32 bit format.

At the end of the script execution, it also lists all the speakers that were identified in order of and with the audio length for each speaker. It also lists the total amount of audio length that had to be skipped due to processing errors, as well as the total time it took to execute the script.

Last but not least, I ran this script on a vast.ai rented Ubuntu based VM with a 4090 GPU and it worked. I did this to test Linux as well as because I was processing over 4h of audio, so I wanted this to be fast. Keep in mind that if you are running this script on your home PC with a bad GPU and are processing a lot of audio, it can take quite a while to complete.

Script is attached.

https://drive.google.com/file/d/13iY2knyABBU-MOaMN5_zNAoDHFMZY6SD/view?usp=sharing

example console output and example speaker output:

<https://discord.com/channels/708579735583588363/708579735583588366/1132503652033114192>

Usage instructions:

install whisperx and its additional dependencies such as FFmpeg as per the instructions on the GitHub page <https://github.com/m-bain/whisperX>

Additionally, install pydub (and any other dependencies you might be missing if the script gives an error message indicating you are missing a dependency)

install ffmpeg-python, make sure to use the following command instead of pip install if you're running this in a conda environment, otherwise it won't work: conda install -c conda-forge ffmpeg-python

edit the script to include your huggingface token and path to the folder containing the audio files you want to process

run the script simply by python your_filename_here.py

Results are quite good for what it is, but you'll definitely need to do some additional editing in audacity and ultimate vocal remover or whatever afterwards to cut out music, noise, and other speakers that were wrongfully included. It definitely works best with speakers that appear a lot in the audio file, like main characters. It does a very good job at separating those.

I won't provide tech support beyond this, as I am no programmer and did this all by just directing ChatGPT."

Or check [alternatives](#)

UVR5 GUI (MDX, VR, Demucs 2-4 and UVR team models)

GUI provides more functionalities and models/AIs compared to Colabs, incl. custom model import:

<https://github.com/Anjok07/ultimatevocalremovergui/releases>

Official app Win 11 installation tutorial:

<https://youtu.be/u8faZW7mzYs>

MacOS build:

<https://github.com/Anjok07/ultimatevocalremovergui/releases/tag/v5.6> (or [beta](#))

MacOS Catalina tutorial (outdated at this point):

<https://www.youtube.com/watch?v=u8faZW7mzYs>

(you better don't run Windows build in W10 VM or you will get like 3 hours processing time)

Windows 7 users:

"To use the newest python 3.8+ with Windows 7 install VxKex API extensions and in case of problems select Windows 10 compatibility in EXE installer properties."

Here you can find a searchable PDF guide by the devs for UVR5 GUI describing functions and parameters (can be outdated):

https://drive.google.com/file/d/1RtMRj8FpSpMHk1XxaBrKoWQImMfCSmy/view?usp=drive_link

Video guide:

<https://youtu.be/jQE3oHXfc7g>

Online guide:

<https://multimedia.easeus.com/ai-article/how-to-use-ultimate-vocal-remover.html>

(their instructions for installing and using stable version seems to be fine, despite the fact they recommend to clone repo for Macs. It's already available as appropriate binaries for M1 and Intel CPUs (/wo Roformer beta patch at the moment)

Some of UVR5 GUI models described in this guide can be downloaded via the expansion pack:

https://github.com/Anjok07/ultimatevocalremovergui/releases/download/v5.3.0/v5_model_expansion_pack.zip

VIP models

<https://www.buymeacoffee.com/uvr5/vip-model-download-instructions>

(some older) settings for the GUI:

<https://photos.app.goo.gl/EUNMxm1XwnjMHKmW6>

(though it's mostly outdated).

(no longer necessary as UVR now has separate DirectML branch and executable:)

Optional fork of UVR GUI for AMD and Intel cards, currently supporting only VR Architecture and MDX using DirectML (Demucs currently not supported). If you have Nvidia card, then use official app above since CUDA is supposed to be faster.

"A four minute and 20 second audio takes about 30 seconds (including saving) using 1_HP-UVR on an Intel Arc A770 16GB. It takes up approximately 6GB of VRAM."

If you only use MDX models, in most cases it won't be faster than processing with CPU - i5 4460 has similar performance to RX 6700 XT here, so better stick to official app.

Compared to Roformer beta #8, it's still much faster at least for VR models, but you might get some issues with MDX-Net models, though.

<https://github.com/Aloereed/ultimatevocalremovergui-directml>

Python command line fork of UVR 5 with current models support:

<https://github.com/nomadkaraoke/python-audio-separator>

(moves from: <https://github.com/karaokenerds/python-audio-separator>)

It was based on some outdated UVR code, but probably got updated since then to support more models.

GUI FAQ & troubleshooting for UVR

Start with reading information about [Roformer patch](#) and its common issues section

- "If you enable the "enable help hints" setting" "you can hover parameters with the mouse, [and] you'll get [settings] info hints (...) if [it's] not activated by default)"

- See the section [above](#) for UVR installation and usage guides

- It's not guaranteed to run on older versions of Windows than 10, so do it at your own risk.

"3.8.10 is the last [Python] official installer that works on Win7, however I was able to find an unofficial [Python] installer from GitHub for 3.10.13 on Win7 and that seemed to do the trick! No more error on load of UVR"

- You may encounter "Encoding failed. ffmpeg/avlib returned error code: 3221225477" while using Manual ensemble and output set to mp3 on Windows 7

"Think I've found my problem. I used a full build of FFMPEG instead of the essentials one."

- "If anyone needs the solution to running it on [MacOS] Mojave+ go to the Releases page on GitHub scroll down to 5.5, under assets grab UVR 5.5 x86_64_9_29.dmg. Confirmed working now on my Mojave machine. Thanks to @ambresakura on GH"

- Installing the GUI outside the default location on C:\ drive, esp. with older versions may result in e.g. startup issues (although they seem to be fixed in 5.6 and Roformer patches). If you lack space on C: drive, create your folders using [Symlink Creator](#) to redirect the content to some other disk, keeping the C:\ location in the Windows file system logic.

Or else, copying only Ultimate Vocal Remover\gui_data folder to the C:\ drive while keeping the GUI installation on another drive might work as well. Although there seem to be no issues with the latest stable 5.6 opened from a different location than default, and standalone Roformer patch installed in a different location.

- There's no way to bypass the 3 GB free disk space requirement on C: drive, even for using AudioTools. [Here](#) someone set UVR to L: drive, and it read free space from that letter, but I'm not sure if they just installed UVR in that location, and whether it's still possible using the latest UVR versions (even when UVR was uninstalled previously) or whether it's enough to copy UVR elsewhere and/or change something in the registry about installation path.

- UVR GUI will only process files with English characters (might be fixed, although some complicated names/paths still give “System error” during separation)

E.g. for RuntimeError: "Error opening 'F:/FI studio files/ACAPELLAS\1y2mate.com - 2 AM Full Video Karan Aujla Roach Killa Rupan Bal Latest Punjabi Songs 2019(Vocals).wav': System error." your file path/file name is too long or contains some unsupported charts. You need to shorten/simplify it and/or copy the file to a different location.

- Be aware that your system may occasionally become unresponsive on slow 2 and 4 core configurations with GPU Conversion disabled while separation is progressing (although, you can set all the priorities in Process Lasso to Idle, and it will be saved for future use).

- The provided directory is not writable or read only

Run UVR as admin, or potentially changing privileges to the output/input folders to everyone might help too (alternatively use “Context Menu” for it described [here](#))

Q: Why Vocal Dereverb Options are greyed out. I can't select more, only "main vocals"

A: This option removes reverb from a vocal stem.

You must have the "UVR-DeEcho-DeReverb" VR Arch model installed to use this option.

- Matchering doesn't work correctly with Opus files (error occurs)

- Matchering doesn't work correctly with mp3 files on Mac (at least x86, error occurs)

- Matchering input audio file length limit before error occur is 14:44 or 15 minutes

- Matchering and Manual Ensemble use only CPU and are fast

- “Download speed of models via Download center was really slow for no apparent reason, like some users have already reported.

I've reduced [the] UVR's window while the download was ongoing and the download speed fastly improved instantly.

I've restored the window, download speed was again instantly slowed down. Re-reduce UVR window, download speed back to normal again...”

- Official UVR requirements from GH page:

Nvidia RTX 1060 6GB is the minimum requirement for GPU conversions.

Nvidia GPUs with at least 8GBs of VRAM are recommended.

Intel Pentium might be unsupported, but AVX or SSE4.2 instructions are not required, so even newer C2Q like Q9650 with SSE4.1 will suffice.

- 2GB VRAM GPUs had some issues even on CPU, maybe it's fixed already

- Official minimum RAM requirement is 8GB, although it works correctly on 6GB RAM too. With 4GB RAM you can run out of memory on longer tracks (probably fixed in many cases in the v 5.5 and newer - you're able to separate Roformers on GPU on 4GB VRAM with 2,5s chunks).
- As from new Nvidia GPUs, something like RTX 3050 (8GB) is a good, cheap choice for even the heaviest processing and is (theoretically) equivalent to Colab's Tesla T4 for CUDA computing power (but it's not really enough for training, of course, and in Colab slower like 3 times). But watch out for smaller 4GB laptop variants, as they can be more problematic. But if you separate a lot using Rofomers, definitely consider something better (look for as many CUDA cores as possible)
- Sometimes 32/64 bit float output set can trigger "FileNotFoundException: "[WinError 2] The system cannot find the file specified"

(troubleshooting continues later beneath)

CPU/GPU performance in UVR

- The higher the total amount of CUDA cores for Nvidia GPU, the faster separation in UVR
- AMD and Intel ARC GPUs using OpenCL are slower in this separation task than CUDA used in Nvidia GPUs. So it's safe to say that Nvidia GPUs from the same performance segment will be most likely faster for separation.
- Min. 4GB VRAM GPUs tested (with some yaml tweaking for Roformers described below), On AMD, 16GB VRAM recommended (so no modifications are required).
Min. NVIDIA Maxwell/900 series GPUs/compute compatibility 5 is the minimum requirement (at least NVIDIA GT 700 series and older are unsupported returning CUDNN_STATUS_NOT_INITIALIZED).
For AMD, at least RX 4GB models tested (not sure about R9 200 4GB GPUs - either if on newer modded Radeon-ID drivers and/or with downgraded DirectML.dll attached with your drivers, copied to UVR\torch_directml folder)
Intel was confirmed to work with ARC GPUs, and Xe integrated graphics (e.g. Tiger Lake 2021) with MDX-Net HQ (v2) models.
RTX 5090 is not yet supported in official UVR packages, you can use OpenCL (DirectML) in options instead (slower).
2GB cards will probably cause issues, and 4GB VRAM too - at least on certain Roformer models and settings (unless dim_t equivalent of chunk_size is set to 301/201, but 201 might have a bit of audible audio skipping at times - see [here](#) later below for dim_t>chunks_size conversion)

- Not meeting these requirements, you're forced using CPU processing, which is very slow - even Ryzen 5950X is slower than 1050 Ti in this application, and 1700X is slower by double than even 940M.
- Since the last updates you can also use AMD/Intel GPU, with separate installer with OpenCL support (most likely min. requirement is GCN or Polaris architectures and up - HD 7XXX and RX 4XX, but even 4GB variants may crash on certain settings).
- Old drivers for GCN GPUs might fail using GPU Conversion option. Consider using Radeon.ID drivers or using DirectML.dll copied from your Windows installation into Ultimate Vocal Remover\torch_directml
- You can also use Mac M1 for GPU acceleration (MPS GPU support in separate installer), but also Radeons acceleration on Intel Macs is supported
- GTX 1660 Ti (6GB) is slow for separation using Roformer models. A better choice is RTX 3050 despite similar performance in games, 3050 has the same amount of CUDA cores as Tesla T4 on Colab, but with less VRAM. Avoid the laptop variant of 3050 which has only 4GB of VRAM, and not 8GB like the desktop variant.
- If you want a fast 2nd hand GPU with more VRAM, consider 1080 Ti or 2080 Ti or even 3080 Ti (16GB). Pretty fast ones for separations.
1080 Ti is much faster in this task than 3060 12GB.
<https://media.discordapp.net/attachments/767947630403387393/1133164474749169864/image.png> (dead)

Q: My AMD/Intel GPU have sudden spikes of usage, or just 30% is being utilized. Is that a CPU bottleneck?

A: Nope. It's just how inefficiently DirectML behaves. It's normal and happens for all people, even on some ancient 4 cores.
We've tested various DirectML.dll libraries major versions since 1.9 (and besides 1.12 and 1.14), up to 1.15.4, and the one attached with UVR (1.10.1.0) was the fastest (1.5.1.0 didn't work).

Separation times chart by Bas Curtiz (various CPUs and GPUs, model cut-off examination)

https://docs.google.com/spreadsheets/d/1R_pOURv8z9GmVkJt-x1wwApgAnplM9SHiPO_ViHWI1Q/edit?gid=460807774#gid=460807774

(probably the results made with old UVR Roformer patch when lower overlap means longer processing time, so the opposite to what's in patches newer than #2)

In addition to the above -

for CPU-only:

- MDX-Net HQ_3 in UVR with CPU takes 2 minutes with Ryzen 5 3600 (some regular song time)
- HQ_4 takes ~13 minutes on C2Q @3.6 DDR2 on CPU, 6GB RAM with default settings
- HQ_3 for 4:19 track takes 20 minutes and 22 seconds ([default](#) overlap and 256 segments, iirc default too)
- HQ_4 is much faster on the same CPU
- On AMD A6-9225 Dual-Core CPU (2/2), 4GB RAM three models ensemble (MDX, MDX, Demucs 4) it took almost 17 hours.
- On i3 3xxx it took around 8 hours (not sure about song elapsed time).
- The main burden in this ensemble on such configuration is Demucs
- MDX23C and Demucs ht/ft cannot be processed under ~5-17 hours without GPU acceleration with CPU-only using C2Q. Probably the same for Roformers.
- MDX-Net HQ_3/4 models and VR models with 512 window size are fine on the same configuration
- MDX-Net HQ_3 in UVR with CPU takes 2 minutes with Ryzen 5 3600 - for unknown regular song between 2-4 minutes)
- HQ_4 takes ~13 minutes on C2Q @3.6 DDR2 with default settings (and it's faster than HQ_3) - for unknown regular song between 2-4 minutes)

for GPU Conversion:

- It will take 39m 28s for 3:28 song using 1296 model on RX 470 4GB and C2Q @3.6 DDR2 and beta 2 patch (GPU OC doesn't really matter here, so the same will be for RX 570 which is basically the same chip after OC/different BIOS) and 18 minutes for Kim Mel-Roformer and 3:01 song, and 45 minutes for unwa v2 and 3:52 song. It's pretty possible that we have a huge CPU bottleneck in that case, as CPU still takes crucial part in separation, even for GPU separation.
- iGPU Intel Iris Xe Tiger Lake 11th Gen i7 on LG Gram notebook from 2021 (newer UVR patch)

I used Unwa's duality v2, chunk size: 2s

"I separated a 6 minute long song, it was a flac file, it took 38 minutes.

- With CPU processing it took around and an hour and twenty minutes irc
keep in mind, I had stuff like VSCode, librewolf, Firefox etc in the background hogging up
memory and CPU as well, during both those instances" 12/16GB RAM recommended - it
uses RAM as VRAM and you will clog almost whole 16GB RAM fast when using apps in
background

- Using HQ_4 is much faster than real-time using default settings, but even longer than
accelerated Roformer, when on CPU only using old Core 2 Quad @3.6 DDR2 800MHz.

- On Mac M1 using dedicated Roformer patch:

- it takes 9 minutes to process a 3-minute song using BS-Roformer 12xx viper model (dim_t
1101, batch size 2, overlap 8) with "constant throttling".

- below 4 minutes for Kim Mel-Roformer (overlap 1, dim 801)

and 11:12 for MDX23C-InstVoc HQ for 04:11 track with default settings

- On 6800 XT it takes one hour for 5 minute song using overlap 5 and unwa's inst v2
Mel-Roformer model (newer UVR patch)

- RTX 3060 Ti allows 3x realtime using BS-Roformer SW 6 stems:
3 minutes of audio takes a minute to process (newer UVR patch)

*Some separation times above could have changed a bit in newer UVR beta Roformer
patches than #2*

(FAQ continues)

- Vocal chops using MDX models are more likely to appear on 4GB VRAM cards (use CPU
processing with e.g. 12GB of RAM to get rid of the issue). MDX HQ_1 (or later) model can
cause errors on some 4GB VRAM laptop GPUs at least with wrong parameters (you might
want to use CPU processing instead, then min. 8GB RAM recommended). We're talking
about the newer Batchmode (you cannot choose Chunk mode anymore in newer versions).

- (no longer needed) 4GB VRAM cards should be supported out of the box with chunks set
to auto (6GB may be required for longer tracks for auto setting or batch processing for
chunks higher than at least 10).

(probably fixed) 4GB GPUs will sometimes force you to reopen UVR after each separation to
free up VRAM or else separation might be unsuccessful (setting chunks in old versions of
UVR to 10 or lower might alleviate the issue).

- UVR5 GUI instead of old CML “mirroring” has now “hi-end process” for VR models which is actual mirroring (no mirroring2, not sure about possible automatic bypass from CML while using ensemble of VR models) but don’t confuse it with old “hi-end process” from CML version which was dedicated for 16kHz models.

Q: If you run a single model with default configuration, it is okay with success. The problem is when ensemble 2 models, it does not have enough resources to complete the process. Unless using a manual ensemble. It also has an error if the chunk size was changed, even with a single model. Seems there is not enough VRAM for processing the song.

A: I had the same issue the other day running ensemble 4 models.

Turned out - as the error msg showed, the chunk size was too big...

I prolly must have changed it by accident to ‘Full’ - when I set it back to ‘Auto’ - it was able to process.

U can find this setting under Settings > Advanced MDX Options.

- (probably fixed) For 4GB VRAM GPU and VR 3090 models (e.g. 1,2,9_HP-UVR) you may need to split e.g. 2:34 song into two parts (I recommend lossless-cut) or eventually use chunks option if you encounter run out of memory CUDA error. Lossless-cut will do chunking, so it won’t be necessary to set chunks in UVR in case of some problems (not in all cases on 4GB VRAM).

- (fixed in the latest Roformer patch iirc) "When choosing to save either vocals or instrumentals only, the app saves the exact opposite (if I want to save vocals only, it will save instrumental, and vice versa)"

- A value of 10 for aggressiveness in VR models is equivalent to 0.1 (10/100=0.1) in Colab

- Hidden feature of the GUI:

"All the old v5 beta models that weren't part of the main package are compatible as well. Only thing is, you need to append the name of the model parameter to the end of the model name"

Also, V4 models are compatible using this method.

- The GUI also has all 4-6 stem models from Demucs 4 implemented. For 4 stem, simply pick up _ft model since it's the best for 4 stems. Demucs-UVR 2 stem model trained on Demucs 3 gets worse results than newer Demucs 4 ft model.

- You might consider using Nvidia Studio Drivers for UVR5. Versus Game Ready normal drivers, they can be more stable, but less often updated. You can check your current type of drivers in GeForce Experience (but if you don't know which ones you have, they're probably Game Ready)

Q: Is there a way I can remove models I already have downloaded?

I want to remove all the HP models, but I don't want to delete them from the directory, I want to be able to get them back if I need them

A: Check the current Download center and if all the models you want are there, then you can delete them and redownload from there later

1. Delete the models from the directory, or
2. Move the models to a separate folder out of the directory

- At least since introduction of Batch mode (now default and the only option in 5.6/Roformer patch), stability of the app on lower VRAM GPUs got improved, but you can see more vocal residues processing on 4GB GPU vs on CPU, while 11GB GPU doesn't really have that problem.

Maybe something changed since batch mode was introduced, but some vocal pop-ups could be fixed only with chunks set to 50-60 (11 and 16GB VRAM cards only) in the older UVR versions and CML code.

Some low values were still culprits of vocal pop-ups in chunks mode (at least before the patch).

I'm not sure if the way of handling chunks has changed since the integration of inference code with MSST repo in for MDX-Net menu models.

- Chart showing separation times for various MDX models and different chunks settings on desktop GTX 960 4GB - [click](#) (dead)

- More in-depth - Settings per model SDR vs Time elapsed -||- (incl. dim_t and overlap evaluation for Roformers) - [click](#) | [conclusion](#)

- Linked above frequency cutoff + Time elapsed per model (GPU vs. CPU) - chart by Bas Curtiz - [click](#)

- (no chunks in 5.6 anymore) On 4GB VRAM cards, you can encounter crashes with the newest instrumental and Kim vocal model while using batch processing. Lowering chunks to at least 10 (but better lower, sometimes still crashes) should help

There's no way to bypass 3GB free disk space requirement on C: drive, even for AudioTools

- If your disk space is not freed after separation, check in PowerShell if you have Memory Compression and Page Combining enabled, by typing:
MMAgent. If not 1) Type: Get-MMAgent 2) Then: Enable-MMAgent -mc ([video tutorial](#))
More typical ways to get more space on C:\

- If something is suddenly eating your disk space on the system disk, check: C:\Users\User\AppData\Local\CrashDumps because UVR can create even a few gigabyte crash dumps. Consider turning on compression in properties for that folder. Also, you can simply search for *.dmp and delete all the existing crash dumps on C: drive.
- You should have around 20GB of free space on C: drive after UVR installation on 12GB RAM configurations for separating top ensemble settings (it uses a lot of pagefile) and at least 10GB for 24GB RAM for long songs on 4GB VRAM cards. You can enable pagefile on another drive as well if you run out of space on the system drive (better if it was an SSD as well).
- Go to Safe Mode and delete your GPU drivers with DDU - it will delete all remain remnants from older versions of drivers
- Delete all restore points besides the newest
- Delete cache taking the biggest amounts of space in your browser if you don't want to clear browser data entirely (e.g. the old-fashioned cookie cleaning).

E.g. in Chrome you can do it here:

chrome://settings/content/all?sort=data-stored

- Consider using CompactGUI for using a better compression algorithm for system compression than built-in context menu. Some programs compress excellent. E.g. Office.
- Use old-fashioned disk cleaning feature in context menu of disk in Computer, and click on the next menu to see more entries (but cleaning up temp in appdata and Windows folder will do similar trick)
- Consider using TreeSize Free in order to investigate the biggest files and folders on your partition
- Sometimes Windows Update leaves lots of unused files after updates are installed - they can be cleaned up too by some methods.

This command helped me free some disk space in the past. IIRC, precisely for WU cache in C:\Windows\SoftwareDistribution

start %systemroot%\system32\rundll32.exe advapi32.dll,ProcessIdleTasks (it will take up to 15 minutes, leave PC for some time, and observe how some processes suddenly use CPU or disk, and suddenly stop, then it's done, eventually maybe after restart the space is freed)

- You can shrink the pagefile to min. 500MB on C: drive and use other partition for pagefile

<https://mcci.com/support/guides/how-to-change-the-windows-pagefile-size/>

—

- Q: When ensembling and having settings test mode enabled, UVR keeps all the different outputs before ensembling in a folder. If you're not careful, these quickly can stack up.
[Is it] Possible to have a feature where UVR automatically deletes those after ensembling?
A: Disable "*Save all outputs*" in *Ensemble Customization Options* > *Advanced Option Menu* is what you ask for.
- Performance of GPU per dollar in training and inference (running a model): [click](#)
- How to check whether the model is instrumental or vocal?

Q: Are VR Arc models also grouped between instrumentals/vocal models, or it's just MDX-Net models?

A: The moment you see Instrumental on top (and Vocal below) in the list where GPU conversion is mentioned, you know it's an instrumental model.

When it flips the sequence, so Vocal on top, you know it's a vocal model.

Same happens for MDX and VR archs.

Q: [How to] have UVR automatically deleting the ensemble result folder after processing a song.

A: Go to settings, ensemble options, uncheck "Save all outputs".

- You can perform the Manual ensemble on your own already separated files (e.g. from Colab) in UVR5 under "Audio Tools". Just ensure that files are aligned (begin in the same place). Sometimes using lossy files can mess with offset and file alignment.

- Furthermore, you can use Matchering in Audio Tools, e.g. to fit muddy results without residues, to the separation with more clarity, but containing residues you want to get rid of. Just use file without residues as target,

- If you have crashes on "saving stem" uninstall odrive

- Q: An option to add the model's name to the output files (this existed in a previous version of UVR but now it's gone) it was really useful when you needed to test multiple models on the same song

A: It's still there under additional settings "Model Test Mode"

- Q: I want to separate an audio from a video (input is still empty when I choose a file)

A: Go to General Process Settings>Accept Any Input

- Q: First time trying the ensemble mode and I used the VR Models: "De-Echo-Aggressive, De-Echo-Normal, DeEcho-DeReverb, DeNoise" now the outputs confuse me. In the folder called "Ensembled-Outputs" there are many files which are from each of the models.

Outside that directory are 2 wav files, one says Echo the other No Echo. Isn't the ensemble mode basically a wav file that goes through each model and saves a final wav file after it went through all the models listed?

A: The two files outside the ensemble folder are the final ensembled files.

The folder is all the separate outputs from each model (you've enabled that in settings)

Q: Those files are final after they went through all the models, right? Not just the DeEcho model.

A: Yes

Q: I am just suspicious of the naming, I see at the time, and it makes sense that the files outside the directory are the final version although are they after all the models or just 1 model.

A: The naming is just whatever stem is assigned to the models, in your case all the models output echo and no echo file

so the final ensemble files will have that in the name

- Q: What is this "band" that I keep seeing in the spectrograph of tracks that I've isolated with x-minus?

A: MDX noise - a noise it produces no matter what. In UVR you can use denoise standard/model in options>Advanced MDX-Net it will do exactly what the below describes: You can either use UVR De-noise model or isolate the track twice. Once normal one and already inverted,

then you add the results of normal-inst, inverted-inst, reinvert the inverted-inst, merge both normal and reinverted-inst.

The merged will be without noise, but 6 dBs higher - so lower the gain accordingly, and you'll get the same, just no noise. Repeat that for vocals obviously.

- Q: voc_ft doesn't have any spectrum above 17.7kHz. How to restore it, and have e.g. 48kHz or 96kHz output like the input file has?

A: Turn off "Match Freq Cut-off" but it copies the remaining frequencies from the original, leading to possibly more noise.

"if you want true 96 kHz you need to manually lower the rate for 44100 Hz or less since the models themselves are 44100 Hz"

- It can happen that VR models using 512 window size can crash on 4GB cards, but 272 will be fine, although it will take more time

Q: "I have tried everything and also googled a lot, but UVR with MDX-Net is producing me this type of noise in every sample I have tried, that was not in the recording before. Anybody have an idea what can cause it?"

A: "It's just part of the architecture. Either run it through a denoise model or run it through it twice with the second time the sound being phase-inverted"

"Enabling Denoise Output should do the trick. I use the Denoise Model option, seems to work quite well, to my ears, at least"

Q: "Is there any way to fix the uvr bve model saying "vocals" on the bgv and "instrumental" on the lead vocal file? It's unbelievably annoying"

A: Change primary stem from whatever it is set to the opposite in model options ([screenshot](#))

Q: Matchering gives errors with long files.

A: 14:44 input length limit for both target and reference audio is set, and sth slightly above it caused error (probably a bit above 15 mins, so maybe 15 minutes is a limit).

If you see the error log, it will specify whether the reference or target file is too long, but the limit is the same for both.

Q: "Is there any way to batch-process multiple different models on the same file?

A: Yeah, ensemble, turn individual outputs on [in options], you'll have the same song over and over, each with a different model name attached, all saved before the final min/max/avg mix"

If you drag and drop many files at the same time into the input field and save intermediate files in options, you don't have to do manually start separation for every song.

The feature to save intermediate files is probably enabled by default in options>Choose Advanced>Ensemble>Save all outputs

IDK if "Model test mode" in Options>Additional is necessary for it (Settings Test Mode can be additionally enabled too, just in case something gets overwritten by accident if you change settings).

So you can simply use Ensemble to pick all models you want to batch process, and using drag and drop, to separate all songs you want, using all models you picked in the Ensemble, and probably intermediate files (so from the models) will be saved rather intact, no matter what ensemble algorithm you'll use. You can make a manual ensemble with any algorithm with existing intermediate files later.

- "Invalid buffer size: 17.34 GB" (Mac ARM) when using demucs_ft

Try to uninstall and make a clean installation of UVR.

Consider also using the latest Roformer [patch](#).

(4-5 max ensemble explained moved to MDX settings)

- Chunks may alter separation results

(update: chunks are now replaced with batch mode on even 4GB cards, feature was introduced in one of beta patches and is available in v. 5.6, and you cannot use chunks experimentally in this version if batch mode gives you some vocal pop-ups vs 11GB GPUs which is a pretty common issue in 5.6; the old text for old UVR pre 5.6 code with chunks available follows).

E.g. a bigger chunk value will less likely cause instruments disappearing.

Chunks 1 is not the same as chunks full (disabled). Also, chunks may cause distorting briefly some vocals in the middle when split is being made. Chunks "auto" is calculated individually for your VRAM and RAM configuration (also song length), so the result will differ among various users for the same song. Maximum chunks value differ for various MDX models (e.g. NET 1 will allow for bigger values than newer Inst models with higher training frequency). You can test what is the maximum supported chunk size for your computer specs till you encounter crash (e.g. for 5:11 song and inst main 496 - chunks 15 (20 for 30 s song) for 4GB desktop card, 38 for 6GB laptop card (50 for NET 1 model), and around 50 for 11GB). Sweet spot for 3:39 track is chunks 55 (works at least on 16GB VRAM) - more than that gives worse results. Also on some GPUs/configuration you may notice some variations in very short vocal bleeding not (fully) associated with chunks which don't happen on e.g. x-minus or

other configurations (1660 Ti vs 1080 Ti and 960 (we don't know what causes it). In this case, you can only alleviate the issue by changing chunks. Be aware that low maximum chunks on 4GB cards beside more sudden vocal residues and cuts in the result, may cause also specific artefacts like e.g. beeping not existing on e.g. 11GB card (the issue happen in Kim vocal model).

(older) UVR & x-minus.pro updates/news (2021-2023)

Q: What is the segment size/overlap for VOC FT processing for uve bve models on x-minus, aufr33?

A: --segments_mdx 384

--overlap_mdx 0.1

uvr bve v1

-0.2, -0.05 and 0.15

Average aggressiveness is 0.0 (for v2)

- Anjok (UVR5) "I made a few more fixes to batch mode for MDX-Net before I release it publicly to GitHub later this week. This install also includes a new full band model that will be included in this week's public patch. Please let me know if you run into any bugs or issues." Link (not needed anymore):"

(the model is called UVR-MDX-NET-Inst_HQ_1 - it's epoch 450, better SDR than 337 and 403 models, only sometimes worse than narrowband inst3 [464])

- Anjok: "I decided to make a public beta for everyone here who wants to try the new patch with **batch mode for MDX-Net** before I release it publicly to GitHub next week. This install also includes a **new full band beta model**! [full_403] Please let me know if you run into any bugs or issues." Patch download link

If you don't have the new model on the list, make sure you have "Download more models" on your list.

- The beta patches are currently only for Windows (but just the fb 403/450 models can be used in the older UVR version, and it works correctly - the patch itself is an exe installer which has the model inside and doesn't check for current UVR installation)

Update 14.02.23

"I found a bug in the MDX-NET.

If the input song contains a DC offset,
there will be a lot of noise in the output!

It has already been fixed on the XM.

It will also be fixed soon in the next UVR GUI update." [Examples](#)

Update 11/12.02.23

"I will soon add a new setting to fine tune the Karokee / B.V. model. This will help remove **even wide stereo lead vocals**.

"You can now specify the placement of the lead vocal. The percentages are approximate vocal wideness."

[Here](#) is the current result. As you can hear, the lead vocals are hardly removed [in the old setting]."

"this is super cool, if you invert the 2 results you can actually get the stereo width vocals isolated

1 step closer to more than just 1 track bgvox separation"

"Ooo that's very interesting, stereo lead vocals always get confused for background ones"

Update 4.02.23

New chain ensemble mode for B.V. models available on x-minus

"the chain is the best bg vox filtering I've ever heard"

"It mixes MDX lead vocal and a little bit of instruments. The resulting mix is then processed by the UVR (b.v.) v2 model and the cleaned lead vocal is inverted with the input mix (song).

Unlike min_mag and other methods, when using chain, processing is sequential. One model processes the result of another model. That's why I called it a "chain". Aufr33

Update 31.01.23

The new MDX Karokee model is ready and will be added to [x-minus.com] tomorrow!
aufr33

New Demucs 4 (probably instrumental) model is in training. edit. training stopped due to technical issues and full band MDX models were trained instead.

"Throwing a Demix Pro karaoke model for comparison... I think the bgv parts still sound better for this song, but demix has more noise on the lead parts

Demix keeps more backing [background] (and somehow the lead vocals are also better most of the time, with fuller sound)"

"MDX in its pure form is too aggressive and removes a lot of backing vocals. However, if we apply min_mag_k processing, the results become closer to Demix Pro."

"In the future, we will create a [b.v.] model for Demucs V4. The MDX-NET is not really well suited for such a purpose."

Update 24.12.22

Wind instruments model (saxophone, trumpet, etc.) added to x-minus for premium users (since March now also in UVR5).

"I tested. Maximum aggressiveness extracts the most amount of instrument, while minimum the least. The model is not bad at all, but has hiccups often (maybe it needs a much larger dataset)"

Maximum aggressiveness "gives you more wind".

Update 20/19.12.22

New UVR5 GUI 5.5.0 rewrite was released. Lots of changes and faster processing.

MDX 2.1 model added as inst main (inst main 496) in UVR5 GUI.

- There was some confusion about MDX 2.1 model being vocal 438, but it's inst main.

MacOS native build available on GitHub.

VIP models are now available for free with a donation option.

More changes:

"Pre-process mode for Demucs is actually very useful. Basically, you can choose a solid mdx-net or VR model to do the heavy lifting in removing vocals and Demucs can get the rest with far less vocal bleed"

"Secondary Models are a massive expansion of the old "Demucs Model" check button MDX-Net used to have. You'll want to play around with those to find what works for the tracks your processing."

There was also Spectral Inversion added, but it seems to decrease SDR slightly.

There was an additional cutoff to MDX models introduced - "Just a heads up, for mdx-net, the secondary stem frequencies have the same cut-off as the primary stems now

There were complaints about lingering vocals (or instrumentals depending on the model) in the upper frequencies that was audible and very bothersome"

Update 04.12.2022

***A new MDX model has been added!**

This model uses non-standard FFT settings optimized for high temporal resolution: 2048 / 5210

<https://x-minus.pro/ai?hp&test>

[results are very promising]

edit. 19.12. Final main model sometimes leaves more vocal leftovers.

Update 16.11.2022

"Due to anti-Russian sanctions, I will no longer be able to receive your donations from December 9th. All available withdrawal methods are no longer available to me. I will try to solve this issue, and probably move to another country such as Kazakhstan or Uzbekistan, but it will take some time, and servers must be paid for monthly.

As a temporary solution, I will use Boosty. I ask everyone who is subscribed to Patreon to cancel your subscription and subscribe to Boosty: <https://boosty.to/uvr>

Just a reminder that I'm switching from Patreon to Boosty.

If you want to renew your subscription but don't want to mess with Boosty, I've found an alternative for *European* users!

<https://www.profee.com>"

If you have any questions, DM aufr33 on Discord.

Update September 2022

New VR model added to UVR5 GUI for patreons.

Update 31.10.22

The release of the new instrumental models for patreons - optimised for better hi-end (lower FFT parameter), not so big cutoff during training and possibly better results for hip-hop (and possibly more genres).

<https://www.patreon.com/uvr>

UVR-MDX-NET-Inst_1 is Epoch 415

UVR-MDX-NET-Inst_2 is Epoch 418

UVR-MDX-NET-Inst_3 is Epoch 464

The last one is the best model (at least out of these three) so far, although - "I like it 50/50. In some cases it does a really good job, but on others it's worse than 418."

"New models are great! I'm having a little issue on higher frequencies hanging in the vocals, but I found I can remove that by processing again"

"Anyone else still uses inst 464? I've been testing it and my conclusion is that it's a great model alongside 418

the pros of it are that it sounds fuller and doesn't have a lot of vocal residues, but it falls short when picking up some vocals, there might be occasions where it misses some bits, or you can hear some very low or very high-pitched vocals (though this is mostly fixed by using other models)"

"I've only tested one track so far, with 468 (My usual first test; Rush - The Pass). First off, it's the cleanest vocal removal of the track yet. First model to really deal with the reverb/echo and faint residuals ... but also the first model to trap a ton of instrumentation in the vocal stem.

Fascinatingly again, the UVR Karaokee model was able to almost perfectly remove the trapped instrumentation from the vocal line, creating a much more perfect result. I don't know if the new models were trained with this in mind, but the Karaokee model has proven to be extremely effective at this. The two almost work as a necessary pair."
(UVR Karaoke model should be available on MVSEP or maybe also x-minus, and of course UVR5 GUI and it's free and public)

September update

of MDX vocal models added only for premium users (more models available in GUI, to be redeemed with code). They're available online exclusively for our Discord server via this link:
<https://x-minus.pro/ai?hp&test-mdx>

(probably not needed or working anymore as training is finished and final models are already released from this training period, but I'll leave it just in case).

edit. Be aware that models below are outdated and newer above supposed to outperform already them

(outdated, as some old models got deleted from x-minus)

mdx v2 (inst) = 418 epoch (inst model)

mdx v2 (voc) = 340 epoch (voc model)

Description for new **MDX** VIP vocal ones (instrumental based on inversion) and instrumental models (vocal models 9.7 (NET 1) and 423 available on MVSEP under MDX-B option):

Vocal models:

- beta 340 is better for vocals, while -
- 390 has better quality for instrumentals, though it has more vocal residues.
- "423 is really nice for extracting vocals, but is not good for instrumentals
- 427 is not good for me."
- "In the last 438 vocals are really nice, also backing vocals. Unfortunately, we can hear more music noises, but voices are amazing" (it's good for cleaning artifacts from inverts).
"(no longer available, at least on x-minus).
- Beta 390 is better than 340. Instruments are cleaner but have more vocal disturbances.
- I've tried a combination of MDX 390 - UVR min_mag_k. Not really bad at all".
- "406 keeps most of these trumpets/saxes or other similar instruments, and ensembling with max_mag means it combines it with UVR instrumental which already keeps such instruments, so you get best of both worlds".

Instrumental models:

- 430 or 418 are worth checking.

Update 17.11.2021 - older public UVR 9.6 and 9.7 vocal models (but still decent) for MDX are described in "[MDX-Net with UVR team](#)" section.

Upcoming UVR5 updates (outdated)

Since the training of MDX September models is completed, some older beta models might not be available anymore.

As of the middle of September a new VR model was in training, but cancelled due to not "conclusive" results, although later a new VR model was released.

"these models will be next:

1. Saxophone model for UVR.
2. "Karokee" model for MDX v2."

Also, completely rewritten UVR5 GUI version.

Among many new features - new denoiser for MDX models available and new Demucs 4 models (SDR 9).

Online sites and Colabs for separation - the best quality freebies you can currently get

Refrain from using lossy audio files for separation (e.g. downloaded from YouTube) for the best results.

See [here](#) for ripping lossless music from Tidal, Qobuz, Deezer, Apple Music or Amazon.

If you don't have a computer, or decent CPU/GPU and separation is too slow on your machine using UVR 5 GUI, or it doesn't work correctly for you, you can use these online sites to separate for free:

[mvsep.com](#) (lots of the best UVR 5 GUI models incl. various Roformers, and some exclusive models not available in UVR, and ensemble of these for paid users)

The page by MDX23 code/Colab original author and models' creator - ZFTurbo.

~~If you register an account on MVSep, you can output in .flac and .wav 32-bit float.~~

Since 28.07.25, now 32-bit float for WAV will be used only if gain level fall outside 1.0 range, otherwise 16 bit PCM will be used.

Also, now FLAC uses 16-bit instead of 24-bit.

If you have troubles with nulling due to the new changes in free version, consider decreasing volume of your mixtures by e.g. 5dB, and you won't be affected, although it might slightly affect separation results.

If your credits are higher than 0, you have shorter queue (also users using the mobile app have "a bit higher priority"), 100 MB max file size/10 minutes (up to 10 concurrent separations in premium and 1GB/100MB in premium). You can disable using credits for non-ensembles in settings, for the cost of a longer queue again. Shorter queues seem to be currently around mornings of GMT +2/CEST (9 a.m.) or even early afternoon or late at night, depends - sometimes the queue goes crazy long randomly, but if you don't care, you can just set your jobs and download it the next day.

Selecting "Extract from vocals part" uses the best BS-Roformer models as preprocessor for the chosen model (currently the ver. 2024.08 - subject to change).

For Mel Band Kar dim_t 801 and overlap 2 is used, and for Mel Bebruily inst/voc, Mel 2024.10, Mel Rofo Decrowd: 1101 and 2.

If downloading from the site is too slow try out e.g. Free Download Manager (Win) or ADM (Android) and/or VPN, or if you have premium you can use your credits to pack to zip the separation after your separation is done.

Batch processing with [API](#) and [GUI](#) - [click \(Mac\)](#). You can use MVSEP download links as remote URL in order to further separate the result (e.g. MVSEP Drums>drumsep for more stems).

Q: Is there a way to turn off the normalization when using FLAC?

It's annoying when you have to combine the outputs later

A: "No, if you turn off normalization, FLAC will cut all above 1.0

And if it was normalized, it means you had these values."

FLAC doesn't support 32-bit float, it's 32 int, so normalization is still needed."

So if your stems don't invert correctly, just use WAV.

Q: How multichannel is handled by MVSEP:

A: [librosa script](#) which performs stereo downmixing (for 5.1 or 7.1 inputs)

Q: I convert a song using v1e+ and use phase fix, then do another conversion using for example Gabox V7 and use phase fix, if I go back to upload the same song using v1e+ it gives the stems instantly but if I use phase fix it will process again, in the past it would remember

A: This may be a temporary issue. Sometimes that server may be unavailable, then processing will start on another server.

Q: What's "include results from independent models"?

A: "When you use an ensemble, you will also get results from each model of the ensemble and not only the ensemble final result."

Q: What means "Disown Expired Separations" option

A: "we do not delete expired separation data (they are needed for analytics), but just remove your ownership from expired separations

We could have written delete expired separations, but wanted to be more clear about your data"

Q: "So I understand, all the uploads are kept, regardless of 'disowning' or not. So what is the distinction between disowning and not disowning? Is there one?"

A: no uploads are kept, just settings. If you disown, you won't see your expired separations

Q: I will need a refresher in terms. Separations are created from (audio) uploads.

Separations are also not kept? Only the settings used, i.e. kuielab_a_drums, aufr33-jarredou_DrumSep_model_mdx23c_ep_141_sdr_10.8059, and whatever segment, aggression, vocal only, etc are selected at the point of hitting 'do it'.. in a manner of speaking..

A: separation is when you choose settings and upload file, we just save the settings and delete file.

Q: How to use the same file over and over for different models in order to test them, but without reuploading the same file over and over

A: "You can use remote upload for this. Just use link on file from previous separation. So you will not need to upload anything. <https://mvsep.com/remote>"

x-minus.pro / uvronline.app (-||-, 10 minute daily limit for free, very fast, mp3 192kbps output for free (lossless for premium), some exclusive models for paid users, Roformers will be back for free around 31 December 2024)

The site is made by one of the UVR creators and models creator - Aufr33 with dedicated Overlap 2 used for Roformers. At subscription level standard and above, song limit for Roformers is 20 minutes. For Mel Karaoke model, dim_t 256 and overlap 2 is being used "Mel-RoFormer by Kim & unwa ft3 and some other models are hidden. As before, you can find them here: <https://uvronline.app/ai?hp&test>" - Aufr33 ([link](#) for free users)

Model used for phase fixer/swapper/correction on the site is Mel-Roformer Becriuly Vocal

Alternatively, you can use Google Colab notebooks for free (with time limits), which are virtual runtime environments configured to use specific architectures and models or ensembles

(see dedicated sections in the document outline for more information on specific Colabs).

If downloading from the site is too slow, go to settings and turn on “Use CDN...” or try out e.g. Free Download Manager and/or VPN.

[Music Source Separation Colab by jarredou](#)

Single models inference kept up to date - new MDX23C Drumsep model, most if not all Roformers (now also /w 1053, and unwa/Gabo models), plus VitLarge, and Bandit model support for SFX and MelBand Decrowd, plus experimental SCNet, and both MDX23C and Mel-Roformer dereverb.

Based on ZFTurbo inference repo (optimized dependencies and frozen commit to avoid issues).

No BigShifts here, and fixed overlap issues with Roformers from UVR.

Use drumsep on already separated drums from already good sounding instrumental.

Use 4 stem models at best on already well separated instrumental with a single model.

Detailed instruction how to use the Colab:

<https://reentry.org/msst-colab>

Q: What is TTA option?

A: “It means “test time augmentation”, with ZFTurbo’s script, it will do 3 passes on the audio file instead of 1. 1 pass will be with original audio. 1 will be with inverted stereo (L becomes R, R become L). 1 will be with phase inverted and then results are averaged for final output. It gives a little better SDR score, but hard to tell if it’s really audible in most cases”

“overlap: This helps improve separation quality slightly. Higher overlap might give better results at the cost of slower processing speeds.” I’ll go for 8. For instrumentals, oeverlap higher than

“chunk_size: Just leave it default unless the model uses higher chunk_size in yaml (the Colab overrides the parameter).

Sometimes it might fail to detect files uploaded on GDrive after mounting on Colab was done. Then open file manager in the Colab and show your input folder, so your file will appear and start working. Sometimes adding a new code field with:

```
drive.mount("/content/drive", force_remount=True)  
will be necessary (it forces remounting).
```

Colab instruction for newbies

0. If you plan to use your GDrive for input files, go there now, and create a folder called “input” and upload your files there. Create also “output” folder in the root GDrive directory (not sure if the Colab creates both already). That way you may decrease the time till timeout, when the Colab is initialized (esp. for people with slower connection), and also you will avoid an occasional bug when files uploaded on GDrive appear with some delay in the Colab.

The Colab is case aware - e.g. call your folder “input” not “Input” to match what is written in the Colab

1. Now open the Colab link in your browser
2. Click the “play” button on the "GDrive connection" cell. Grant all the privileges (otherwise there will be an error).
Don't use any other account than you're already logged in in the right top corner (otherwise it will error out).
3. Click the “play” button on the "Install" cell, and wait patiently til it's finished (it should show a green checkmark on the side afterwards) - be aware that rarely it can take a longer time than in most cases.
4. Now pick your model in "Separation" cell.
5. Click the “play” button on the "Separation" cell.
Don't provide any filenames in input_folder path there. It will batch process all the files inside the input folder.
Default settings are already balanced in terms of SDR, and not too resource-intensive (increasing overlap might muddy instrumentals a bit, 8 might have a bit more information on spectrogram iirc in vocals).
TTA increases SDR a bit. I'd leave it turned on, although it will separate 3 times.
Chunk_size should be left default, as it's the value used by most models, but iirc beta 6 uses higher chunks. Refer to the yaml of the model, as the Colab will override yaml setting.
6. After it's done, it will output the stems in the output.
7. Before closing, go to Environment and delete the environment manually, so you won't exceed your free Colab credits (so you'll be able to use it e.g. next day).
You should be able to use the Colab for 3,5h+ per day (I think 4h in at least not one single separation job started).
If your GPU gets disconnected, change the Google account in the right top corner of the Colab and use the same account to mount GDrive.

[Colab of 6 stem undef13 splifft](#) - SW BS-Roformer model, but in FP16 (almost identical metrics, but faster)

[Custom Model Import](#) Version of the inference Colab by jarredou.

You can use it if we don't add any new model to the main Colab on time, or you test your own models.

Just make sure you “you have downloaded the webpage presenting the model instead of the model itself.”

E.g. for yaml from GH, use e.g.:

https://raw.githubusercontent.com/ZFTurbo/Music-Source-Separation-Training/main/configs/config_vocals_mdx23c.yaml

Instead of:

https://github.com/ZFTurbo/Music-Source-Separation-Training/main/configs/config_vocals_mdx23c.yaml

And for HF, follow the pattern presented in the Colab example (so with the resolve in the address)

“If you don't delete the failed yaml/ckpts downloads you've made before [e.g. wrong link pasted], the Colab will continue to use them.” so delete the files manually from file manager or restart environment while still getting errors.

[Phase fixer Colab](#) by santilli_ using Kim model phase for unwa's v1e/v1/v2 and other models to get rid of some noise ([older outdated](#), [newer](#))

[UVR5 UI HuggingFace \(mirror\)](#) maintained by NotEddy and hosted by their friend - running on Zero GPU (A100 cluster), it has most models from the inference Colab. Might be faster. (HF has a quota ~12 min of usage each 2 hours, and it doesn't have TTA). Some [advice](#) to make it work on PC.

[SESA Colab by yusuf v3](#) - WebUI for the same ZFTurbo inference code (might differ in available models)

[Extended inference Colab](#) by makidanyee (also based on jarredou's) containing phase fixer as a separate cell to work on ready separations, zip/unzip cell, manual ensemble (all in one)

[Multi-arch Colab by Not Eddy](#)

Archs: MDX-Net, MDX23C, Roformers (incl. 1053), Demucs, and all VR models (incl. e.g. de-echo not supported in VR HV Colab) with YouTube support and batch separation.

If you encounter increased separation time (like 5 hours) using some high parameters for MDX-Net models (e.g. 512 segment size and 0.95 overlap) use another Google account. You could've reached free daily limit.

Plus, be aware that the Colab uses broken overlap from OG beta UVR core for Roformers, so the same fix for the issue applies:

Don't set overlap higher than 10 for 1101 segments, and overlap 8 for 801. Best SDR is dim_t=1101 and overlap 2.

[Not Eddy's multi-arch Colab](#) in form of UI (like in e.g. KaraFan)

In case of “FileNotFoundError: [Errno 2]” try other location than “input”, or other Google account in case of ERROR - mdxc_separator (helps for both errors).

Or use the Colab below for Roformers instead:

[MVSEP MDX23 jarredou fork Colab v.2.5](#) (2-4 stems)

It has adjustable ensemble of BS-Roformer Viperx, Kim Mel-Roformer, UVR MDX-Net HQ_4, MDX23C HQ 1, VitLarge, voc_ft and has optional output to 4 stems using ensemble of various 4 stem demucs models. Original 1.0 code made by ZFTurbo (MVSEP). Consider using already well separated instrumental as input from the above Colabs.

You can manipulate with weights there to have more of a specific model in the final result. Default settings can be a good start.

Sometimes you might want to disable VitLarge.

Also, some people like to increase BigShifts to 20 or even 30 with all other default settings (some songs might be less muddy that way), but default 3 is already a balanced value, although exceeding 5 or 7 may not give a noticeable difference, while increasing separation time severely over default settings. [Read](#) for more.

[KaraFan by Captain FLAM](#)

It allows using currently all notable UVR instrumental and vocal models besides BS-Roformer, also in ensemble (with suggested ensemble presets - start with P5 for instrumentals and P4 for vocals), but with further tweaks and tricks in order to get the best quality of instrumentals and vocals sonically, but without overfocusing on SDR only, but on the overall sound. Usually more vocal residues than instrumental Roformers.

[MDX-Net Colab by HV](#)

All older notable UVR-MDX models are in this fork, including HQ_5 (don't confuse with MDX23C arch) - very fast once the Colab initializes, but more vocal residues than instrumental Roformers.

[MDX-Net alternative kae Colab](#) (fork of one earlier HV Colab version, not sure if it still works)
In comparison to the above, it has the old min/avg/max mag mixing algorithms and optional Demucs 2 ensemble for only vocal models.

[VR HV Colab](#) (even older archs with even more residues, no de-echo model - it's included in Not Eddy's HF/Colabs above)

[Demucs 4 - for 4 stems](#) (lower SDR than MDX23 Colab)

You might want to use here already well separated instrumental with the methods above

[Batch separation for Demucs](#) by jarredou (less friendly GUI, but should be usable too)

older [Drumsep](#) by Imagoy (newer model above in Music Source Separation Colab) - kick, snare, hi-hat, toms (based on Demucs v3) - also use on already separated drums from already good sounding instrumental

[LarsNet](#) - kick, snare, hihats, toms and also cymbals separation (can be worse than the old Imagoy's based on Demucs at times, but has more stems)

[UVR on hugging_space](#) (incorporates VR de-echo not available in HV Colab, it's slower than Multi-arch Colab above)

Bandit Plus, Mel-Roformer by jazzpear SFX separation - Colab by joowon
<https://colab.research.google.com/drive/1efoJFKeRNOulk6F4rKXkjg63RBUm0AnJ>

[ByteDance-USS](#) (SFX separation based on audio sample, March 2024 update)
<https://colab.research.google.com/drive/1f2qUITs5RR6Fr3MKfQeYaaj9ciTz93B2>
Colab by jazzpear94

[MedleyVox Colab](#) by Cyrus (can be used on MVSEP too)
with chunking introduced
Use already separated vocals as input (e.g. by [these](#) models).

Collabs for upscalers (AudioSR, FlashSR, Apollo and more) - [here](#)

Other Colabs

MVSEP-MDX23 v2
https://colab.research.google.com/github/jarredou/MVSEP-MDX23-Colab_v2/blob/main/MVSEP-MDX23-Colab.ipynb

MVSEP-MDX23 v2.1
https://colab.research.google.com/github/deton24/MVSEP-MDX23-Colab_v2.1/blob/main/MVSep_MDX23_Colab.ipynb

MVSEP-MDX23 v2.2
https://colab.research.google.com/github/jarredou/MVSEP-MDX23-Colab_v2/blob/v2.2/MVSEP-MDX23-Colab.ipynb

MVSEP-MDX23 v2.3
https://colab.research.google.com/github/jarredou/MVSEP-MDX23-Colab_v2/blob/v2.3/MVSEP-MDX23-Colab.ipynb

Not sure if all of these older versions has the following fix for slow separations:
!python -m pip -q install onnxruntime-gpu --extra-index-url
https://aiinfra.pkgs.visualstudio.com/PublicPackages/_packaging/onnxruntime-cuda-12/pypi/simple/

jazzpear's soon to be 17-stem separation Colab (probably doesn't work anymore)
<https://colab.research.google.com/drive/1jrw-cAi-JqZpBi6wyT3YIp3x-XHhDm1W>

Similarity Extractor

<https://colab.research.google.com/drive/1WP5ljdutcc-RRsvfaFFIhnZadRhw-8ig>

But Audacity's center extraction which can be used also online works better:

wavacity.com

There was also MDX23C model for the same purpose released:

<https://github.com/ZFTurbo/Music-Source-Separation-Training/issues/1#issuecomment-2417116936>

Useful repositories

Python command line fork of UVR 5 with current models support

<https://github.com/karaokenerds/python-audio-separator>

(it used to have the same broken overlaps from UVR for Roformers)

OG repo on which jarredou's single models Colab separation is made

<https://github.com/ZFTurbo/Music-Source-Separation-Training>

(can be used locally both for inference [separation] and training)

It has other GUI too:

<https://github.com/AliceNavigator/Music-Source-Separation-Training-GUI>

Using only CPU in the GUI might be fixed by changing line 149 to
device = 'cuda'

<https://github.com/AliceNavigator/Music-Source-Separation-Training-GUI/blob/66ada053a623a20865cac7b9d26a02615204d178/inference.py#L148> ~frazier

WebUI:

<https://github.com/SUC-DriverOld/MSST-WebUI>

Other GUI for UVR

<https://github.com/TheStingerX/Illaria-UVR>

Good paid sites:

- dango.ai (expensive, one of the best results for instrumentals)
- moises.ai (probably in-house BS-Roformer models)
- studio.gaudiolab.io (a.k.a. GSEP, still good for specific cases)

- [Music AI](#) - better results than those on Moises (same team). \$25 per month or pay as you go, pricing chart, no free trial, Good [selection](#) of models and interesting [module stacking](#) feature. To upload files instead of using URLs “you make the workflow, and you start a job from the main page using that custom workflow” by [~ D I O ~].

“Bass was a fair bit better than Demucs HT, Drums about the same. Guitars were very good though. Vocal was almost the same as my cleaned up work. (...) I’d say a little clearer than mvsep 4 ensemble. It seems to get the instrument bleed out quite well, (...) An engineer I’ve worked with demixed to almost the same results, it took me a few hours and achieve it [in] 39 seconds” by Sam Hocking

- [Audioshake](#)

- [Myxt](#) - 3 stem model, unfortunately, it has/had WAVs with 16kHz cutoff which Audioshake normally doesn’t have. No other stem. Results, maybe slightly better than Demucs. Might be good for vocals.

Thanks to Mr. CRUSTY crab for gathering lots of the links.

More online site descriptions

<https://mvsep.com/> (FLAC, WAV 24 bit/32 bit for MDX instrumentals and Demucs, Roformers, 100MB per file limit, MP3 320kbps available, 512 window size for VR models (all UVR 5 GUI models including WiP piano [it’s better than Spleeter worse than GSEP]), /wo HQ_4, big choice of various architectures and models.

Good instrumental models: MDX23C 16.66, MDX B>HQ_3, BS-Roformer 17.55

Good vocal models: MDX B>voc_ft, MDX23C 16.66 (more bleeding, better quality)

Ensemble for paid users (instrumentals have fewer residues than 2.4 Jarredou Colab, but are muddier)

(old) In **Demucs 3-UVR** instrumental models - model 1 is less aggressive, model 2 is more destructive (sometimes it happens the opposite, though), the “bag” leaks even more, also, regular 4 stem model B - mdx_extra from Demucs 3 and also HT Demucs 4 (better ft model). For UVR-MDX models choose MDX model B, and the new field will appear. Biggest queue in the evenings till around 10 PM CEST, close to none around 15:00 (working days).

<https://x-minus.pro/ai> (10 minutes daily limit for free users - it can exceed like 1 minute on the last song)

Currently, more models are available for free (like MDX and Roformer models), but some more resource hungry methods like drums max mag are behind payroll.

Good methods:

Models ensembled - available only for premium users:

- demudder (used on Mel-Roformer)
- Mel-Roformer + MDX23C

- drums ensemble max_mag with Roformer

(old) Previously for free users only one UVR model without parameters for "lo-fi" option was available (unreleased model, mp3, 17kHz cutoff) and Demucs 3 (2 stem) (or 6 stems?) for registered users (site by of the authors) and Demucs 4 (4 stem) for premium users (and its better htdemucs_ft model for songs shorter than 5 minutes [better equivalent of previous demucs_extra model which wasn't quantized] and 7-8 minutes in the future (not sure if it also got replaced by 6s model for premium users as well)).

Besides WAV, paid users get exclusive unreleased VR model when aggressiveness is set to minimum.

As the site development dynamically progresses, some info above can be outdated.

MSST / MSST-GUI by ZFTurbo

Repository of MVSEP creator and model trainer:

<https://github.com/ZFTurbo/Music-Source-Separation-Training>

It can be used either for training or also inference (separation using models).

MSST-[GUI](#) by Bas Curtiz (default list of models can get outdated, but you can provide file paths manually there too) other GUIs linked at the bottom. The GUI has screen reader compatibility.

(If you have a hard time setting your local Python environment with the above, you could try out portable installation of [Scial's WebUI](#) fork, but I cannot guarantee the compatibility with all the latest models - the 1.7.0 code derives from April 2025)

1. If you deal with dequantization error while occurring on e.g. crowd model on 1 hour mp3 file, use this repo instead:

<https://github.com/jarredou/Music-Source-Separation-Training/tree/colab-inference>

“It’s the one used in Colabs” - jarredou. Sometimes MSST updates might break things, while here a certain older checked commit is used.

2. “For some reason when I use unwa models it just like gives back a really quiet resampled version of whatever I put in.”

A: “Check that you are using an up-to-date version of the repo, IIRC, an edit made some months ago to Roformers code was creating weird issues similar to this for some people and was removed later”

Q: “Works now”

Hint: Requirements just for inference might be faster to install, like presented in the Installation cell in [this](#) Colab.

3. For state_dict error using existing MSST installation, update MSST to the last repo version with:

!rm -rf /content/Music-Source-Separation-Training

```
!git clone https://github.com/ZFTurbo/Music-Source-Separation-Training  
“and you must reinstall the main branch's requirement.txt. (before it, edit requirements.txt to  
remove wxpython)” - Essid  
wxpython is for GUI.
```

- MSST works on CPU or NVIDIA GPUs (by default it uses GPU if it's properly configured), and separates up to 3 times faster than UVR on 8GB GPUs (on 4GB it might be even slower). It was also tested on Linux and ROCm 6 and 7 on AMD GPU.

- Unlike in UVR, MDX-Net v2 and VR archs are unsupported here.

- Officially [supported](#) AMD consumer GPUs for ROCm on Linux are:
RX 7900 XTX, RX 7900 XT, RX 7900 GRE and AMD Radeon VII on Ubuntu 24.04 LTS using Pytorch 2.6 for ROCm 6.3.3, but also RX 9070 and RX 9070 XT and RX 6700 XT should be manageable to work, and probably 5700 XT with older ROCm version.

- For RX 7900 XTX “No special editing of the code was necessary. All we had to do was install a ROCm-compatible version of the OS, install the AMD driver, create a venv, and install ROCm-compatible PyTorch, Torchaudio, and other dependencies on it.” - unwa

- ROCm 7 officially working with Instinct MI350 CDNA 4 was released, providing 3-7x performance gains over 6.0 ([more](#)). You can try your luck with it on other GPUs by e.g:

```
pip install --pre torch torchvision torchaudio --index-url  
https://download.pytorch.org/whl/nightly/rocm7.0
```

- Since then also ROCm 6.4.4 allowing using PyTorch natively on Linux and Windows on RX 7000 and 9000 was released ([more](#)), but one of our users had building errors with MSST ([fix](#)), and also separate [instruction](#) was written for MSST-WebUI. Instead, you could consider using WSL on Windows instead too (it's Ubuntu with direct GPU access on Windows with near zero performance overlay, and even GUI support in newer WSL/Windows version), then follow e.g. the below on WSL:

- “I managed to make [MSST-WebUI](#) work [on Linux] with:
Torch 2.10.0.dev20251110+rocm7.0
on RX 7600
(...) it seems like ROCm 7.0 is about a second faster [than 6.x]
(probably by adding just pip install before it)

turns out that if you do:

```
export TORCH_ROCM_AOTRITON_ENABLE_EXPERIMENTAL=1  
it uses waay less VRAM and processes even faster  
inst_V1e_plus batch_size=2 overlap=3 chunk_size= 485100, 51.78s/it [3:50 of audio in 61  
seconds]
```

For ROCm 6.x (a tad slower, might work on more GPUs):

torch 2.9.0+rocm6.3 torchvision0.24.0+rocm6.3 [--index-url

https://download.pytorch.org/whl/rocm6.3]

Thanks, fr4z49.

Official support for PyTorch on RX 400/500 (a.k.a. Polaris/GCN 4/GFX803) GPUs was dropped, but you can follow [this](#) Ubuntu guide for unofficial ROCm 6 support (it might even potentially work from Windows using WSL with almost no GPU performance overhead).

Or for ROCm 5, read [this](#) Ubuntu guide.

Also, there seems to be some Arch Linux community package to install Pytorch still compatible for these GPUs ([click](#)).

Or optionally also might be potentially supported with some other specific versions of ROC, e.g. 5.7.2 and also described above:

export ROC_ENABLE_PRE_VEGA=1 (deprecated in ROCm 6; might help for lacking dependencies or wheel building issues). Or also check out this:

<https://github.com/AUTOMATIC1111/stable-diffusion-webui/issues/10435#issuecomment-1555399844>, or alternatively follow below instructions:

<https://pytorch.org/get-started/locally/> and then execute:

pip3 install torch torchvision torchaudio --index-url

<https://download.pytorch.org/whl/rocm5.4.2>

4. If you have SageAttention error, you need an arch corresponding to e.g.: RTX 5000, 4000, 3000, H100, H200 (Ada Lovelace, Hopper, Ampere, Blackwell) which will probably work out of the box. Otherwise, it will probably fall back to CPU. To fix it, make sure you have installed CUDA/Torch/Torchvision/Torchaudio compatible with your GPU (probably it will work down to Maxwell GPUs (not sure about Kepler)):

"For the [GTX] 1660 the minimum [CUDA] version is 10" E.g. minimum compatible CUDA version requirement for GTX 1660 is 10 (on GTX 1060, Torch 2.5.1+cu121 can be used), but pip doesn't find such a package of Torch (and usually it fixes issues when CPU is only used on those GPUs).

Check out index-url method described later below:

pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118

or

pip install torch==2.3.0+cu118 torchvision torchaudio --extra-index-url

<https://download.pytorch.org/whl/cu118>

or

pip install torch==2.3.0+cu118 --extra-index-url https://download.pytorch.org/whl/cu118

and

pip install torchaudio==2.3.0+cu118 --extra-index-url https://download.pytorch.org/whl/cu118

Replacing cu118 with newer cu121 seems to give a proper working URL too.

Maybe replacing 2.3.0 with 2.3.1 will work too. cu126 is the latest supported for GTX 1080, while cu128 isn't (but supports RTX 5000 series), although it works with torch 2.7.0 which can cause unpickling errors with some models:

```
"pip install torch==2.7.0 torchvision --upgrade --index-url  
https://download.pytorch.org/whl/cu126".
```

JFYI, the official PyTorch page: <https://pytorch.org/get-started/previous-versions/> lacks links for CUDA 10 compatible versions for older GPUs other than v1.12.1 (which is pretty old, and might be a bit slower if even compatible at all), so the only way to install newer versions for CUDA 10 is --extra-index-url trick, as executing normally "pip install torch==2.3.0+cu118" will end up with the version not found error.

Alternatively, you might also try out installing it from wheels from [here](#) by the following command:

"pip install SomePackage-1.0-py2.py3-none-any.whl" - providing a full path with the file name should do the trick. Just for the location with spaces, you also need ". ". On GTX 1660 and Turing GPUs, you might seek for e.g. cu121/torch-2.3.1" and those various CP wheels (there are no newer versions). But the -extra-index-url trick above should be enough.

4.1. After performing all of these, you might still have SageAttention not found error on GPUs up to Turing arch. Then perform the following:

"Had to replace cufft64_10.dll from C:\Users\user\AppData\Local\Programs\Python\Python313\Lib\site-packages\torch\lib by the one from C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\bin" It is even compatible with the newest Torch 2.8.0 (if you followed the instruction to fix the dict issue above) if you grab that apparently "fixed version of cufft64_10.dll from CUDA v10.0" - dca

5. If you write Python in CMD, and it wasn't found, start with method 2 described here:

<https://www.liquidweb.com/help-docs/server-administration/windows/adding-python-path-to-windows-10-or-11-path-environment-variable/>

Or make sure you've checked the option to add path environmental variable during Python installation.

Also, you can try out "disabling the python executable in app execution aliases." - neoculture

6. For state_dict = torch_load/unpickling_error

"add the following line above torch.load (at utils/model_utils.py line 479):

```
with torch.serialization.safe_globals([torch._C._nn.gelu]):
```

[So, the code like the following:]

```
else:  
    with torch.serialization.safe_globals([torch._C._nn.gelu]):
```

```
    state_dict = torch.load(args.start_check_point, map_location=device,
weights_only=True)
    model.load_state_dict(state_dict, strict=False)
```

“(~unwa)

*. For “np.complex” error with incompatible Numpy (Python 12) execute:

pip install numpy==1.26.4

pip install -U librosa audiomembrations

7. For: “failed to build diffq pesq” [click](#)

8. For errors while installing py file for HyperACE model in Sacial’s WebUI:

from models.bs_roformer.attend import Attend

ModuleNotFoundError: No module named 'models'"

fix: “SUC-DriverOld/MSST-WebUI use the name "modules" and ZFTurbo/Music-Source-Separation-Training use the name "models". And Unwa's bs_roformer.py that you replace with, also use "models". So you'll have to do some coding and symlink to make it work.” - fjordfish

9. "ERROR: No matching distribution found for pedalboard~=0.8.1" when MSST requirements

fix: “downgrade python” - Stray Kids Filters

More notes:

- 4GB VRAM GPUs will give out of memory errors on for Roformers. You can use CPU instead, or potentially decreasing chunk_size as described [here](#) might help too.

- Leave the checkbox “extract instrumental” disabled for duality or potentially other models with more than one stem target (it will have worse quality than dedicated stem output)

CML guide by mesk (working on RTX 3070 Mobile):

“0 – You need Python:

<https://www.python.org/downloads/>

0a – I would also recommend installing Pytorch too from here:

<https://pytorch.org/get-started/locally/>

(grab the command and enter it into the command prompt)

0b – But then you can just also double-click on guixw.py on the repo, and it is much easier.

That's the harder method with the command prompt

1 – Go there: <https://github.com/ZFTurbo/Music-Source-Separation-Training>
and clone the repository (click on Code => Download as zip)

2 – Go to the repo folder, create 3 new folders: results, input and separation_results.
Place your tracks in the input folder. Place the checkpoint in results and leave the yaml at
the root of the repo (where inference.py and requirements.txt are)

3 – Open command prompt, type in cd

C://Users/[YOURUSERNAME]/Desktop/Music-Source-Separation-Training-main
(changes directory to the repo folder on your desktop)

4 – Type in: python install -r requirements.txt

5 – Let it install the requirements

6 – Type in: python inference.py --model_type mel_band_roformer --config_path [NAME OF
YAML] --start_check_point results/[NAME OF CHECKPOINT] --input_folder input/ --store_dir
separation_results/ --extract_instrumental

7 – Make sure to replace the stuff in brackets with the actual stuff you need”

If you have decent Nvidia GPU, and no GPU acceleration, maybe “Check these commands
to install torch version that handle CUDA”:

pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
or

pip install torch==2.3.0+cu118 torchvision torchaudio --extra-index-url
https://download.pytorch.org/whl/cu118

or

pip install torch==2.3.0+cu118 --extra-index-url https://download.pytorch.org/whl/cu118
or

pip install torchaudio==2.3.0+cu118 --extra-index-url https://download.pytorch.org/whl/cu118

[various GPU archs will have different CUDA requirements for different Torch versions, refer
to documentation]

FAQ

Q: "When running inference, I am getting tired of it creating a folder for each input file and putting individual stems inside that folder. Especially since most of the time I'm running single stem models and I only need the primary stem. I remember on a much older version, it wouldn't create folders, it would just copy the original filename and put the stem name at the end of it. So I was wondering what I could modify in newer versions to restore that behavior. I'm guessing that would be in inference.py, but don't exactly know where to look." - Musicalman

A: "You can download the "old" version from my forked repo's "colab-inference" branch, with old behaviour for results and folders. It's the version used in my colab notebooks, it should be preselected with that link:

<https://github.com/jarredou/Music-Source-Separation-Training/tree/colab-inference>" - jarredou

- torch.load and load_state_dict, errors

A: "PyTorch 2.6 and later have improved security when loading checkpoints, which causes the problem. torch._C_.nn.gelu must be set to exception" or "add the following line above torch.load (at utils/model_utils.py line 479)

with torch.serialization.safe_globals([torch._C_.nn.gelu]):

"don't forget to align the indentation since it's Python code." - unwa

Also in the case of the unwa's FNO model: "edit the model file

As I mentioned in the model card, you need to change the MaskEstimator"

Q: "I've been using a really old version of msst for a while and finally decided to update it today. I noticed that the gui-wx.py file was moved to the gui folder (it used to be in the root). So now when I try to launch the gui, I get file not found errors. Gui still works at least for screen reader users like myself, but these errors should definitely be fixed.

I'm wondering if I should be trying to launch the gui from the main msst folder, or if I should be launching it from the gui folder. Either way I get errors. I could fix them by modifying paths in gui-wx.py, just need to know which folder I should be starting from lol" - Musicalman

A: "You can use python gui/gui-wx.py and replace (130-131 strings) right now:

```
font_path = "gui/Poppins Regular 400.ttf"
bold_font_path = "gui/Poppins Bold 700.ttf"
```

I will fix it at next pull-request" Kisel

MVSEP models from UVR5 GUI explained

- Ensemble option - further developed custom code of the original MDX23 (not available in UVR) - custom tech, consisting of various models from UVR and in-house, non-public models unavailable in UVR

- Demucs 4 ft - (settings might be shifts 1 and overlap 0.75 as he tested once) - same as in Colabs or UVR

MDX B (not sure about whether min, avg, max is set) - the option has MDX arch models:
- Newest MDX models added - Kimberley - Kim inst (ft other), Kim Vocal 1 & 2, HQ_2, 3
- 8.62 2022.01.01 - is NET 1 (9.7) with Demucs 2, this one has a new name now. It had slightly bigger SDR in the same [multisong](#) dataset as the newer model below - discrepancy vs UVR5 SDR results might be on the server side (e.g. different chunks), so it might be still the same. The dates can only relate to date of adding the model to the site and nothing more, not sure here, but it might be it - NET 1 is older model than below indeed. Looks like that the model is used with Demucs 2 enabled (at least he said it was configured like this at some point)

- 8.51 2022.07.25 - might be vocal 423 a.k.a main, not sure if with Demucs 2 (judging how instrumental from inversion in 423 looked like - cannot be any inst model yet, since they were released in the end of 2022 - epoch 418 in September to be precise) - it was tested in multisong dataset on page 2 as MDX B (UVR 2022.07.250 - the date is the same as before, so nothing new here), can't say now if Demucs 2 is used here. In times of 9.7/NET 1 it was decreasing SDR a bit on I don't know which dataset, but instrumentals usually sounded kinda richer with this enabled. Now it's better to use other models to ensemble.

The change in MDX-B models scheme was probably to unify SDR metrics to multisong dataset.

- Demucs 3 Model B - mdx_extra (and rather not mdx_extra_q as ZFTurbo said it's "original" and used mdx_extra name on the channel referring to this model; in most cases the one below should be better)

- Ultimate Vocal Remover HQ - the option has VR architecture models

Window size used - 512

[Here](#) are all VR arch model names

- UVRv5 Demucs - rather the same names
- MVSEp models - unavailable in UVR5
- MDX B Karaoke - Possibly MDX-UVR Karokee or MDX-UVR Karokee 2 (karokee_4band_v2_sn irc), maybe the latter

The rest below on the MVSEP's list is outdated and not really recommended to use anymore

Issues using MVSEP

- NaN error during upload is usually cause by unstable internet connection, and it usually happens on mobile connections when you already upload more than one file elsewhere. If you have NaN error, just retry uploading your file.
- Rarely it can happen after upload that error about not uploaded file occurs - you need to upload your file again.

- If you finished separation and click back, model list can disappear till you won't click on other algorithm and pick yours again. But if you click separate instead, it will process with the first model which was previously on the list (at least if it was also your previous choice).

- Slow download issues. Separation was complete, and I was listening to the preview when playback on the preview page simply stopped, and couldn't be started. Main page didn't load (other site worked).

Also, I couldn't download anything. It showed 0b/s during attempt of downloading.

Two solutions:

- close all MVSEP tabs completely and reopen
 - Connect to VPN, preview some track, but after a short time, the same can happen and nothing is playing or buffering. Then fire up Free Download Manager, and simply copy the download link there, and it will start downloading. Later, the browser can also start downloading something you clicked a moment ago. Crazy.
-

Comparing to MDX with 14.7 cutoff, depending on a track, VR models only (not MDX/Demucs) might leave or cut more instruments or leave more constant vocal residues, but in general VR is trained model at 20kHz with possible mirroring covering 20-22kHz, generally less aggressive vocal removing (with exceptions) but most importantly, comparing to MDX, VR tends to leave some specific noise in a form of leftover artifacts of vocal bleeding, but from the other hand MDX, especially models with cutoff, can be more muddy and recall original track mastering less.

Manual ensemble Colab

You can perform manual ensemble on your own files in UVR5 under "Audio Tools" or beside [DAW method](#), you can also use:

Ensemble Colab for various AI/models

If you want to combine ready result files from various MDX and Roformer models or different archs/AIs from external sources using Google Colab, here's a notebook for you:

https://colab.research.google.com/github/jarredou/Music-Source-Separation-Training-Colab-Inference/blob/main/Manual_Elaborate_Colab.ipynb

(implementation of ZFTurbo code with drop-down menus plus manual weights and various ensemble algorithms by jarredou)

Once you mount GDrive, open file manager on the left, right click>copy path>paste in the first input field, then for the second file in the second field and so on and so forth. Then you can change type from max to e.g. avg or set weights manually - so to have the specified amount of one model in the result file (you could listen to the imported stems altogether in DAW to actually know what you're doing).

Possible fixes for the errors

- Don't use spaces in output file name

- "AttributeError: module 'numpy' has no attribute 'float'
np.float was a deprecated alias for the builtin float."

> "Try to rerun the install cell, this issue is because of a problem with numpy version
if it doesn't work you can force numpy upgrade by creating a new code cell with:

`!pip install -U numpy`

Sometimes install cell ask you to restart runtime because of numpy version too, if you don't say yes, you have to restart runtime by yourself to make it work"

"Try forcing librosa update too:

`!pip install -U librosa`

Have you tried to delete runtime and restart it from scratch ?

It's weird that these issues happen again, they were lots of these with old colab, but for recent ones, not much"

"If you face this error again, you can update the 2 libs at the same time with:

`!pip install -U numpy librosa" -jarredou`

- "ValueError: setting an array element with a sequence. The requested array has an inhomogeneous shape after 2 dimensions. The detected shape was (2, 2) + inhomogeneous part."

> In some specific cases (not always) converting your file to 32-bit float WAV might help.

Not sure if exactly the same length is necessary. But you can check it if above fails.

Also, lossy files will not align with lossless, and also files with different sample rate.

> For one person helped converting files to 320kbps mp3 for the ValueError

- ValueError: Homogenous shape

>? anything from the above

(Old not working Colab by ZFTurbo

https://colab.research.google.com/drive/1fmLUYC5P1hPcycl00F_TFYuh9-R2d_ap?usp=sharing

https://cdn.discordapp.com/attachments/708912741980569691/1102706707207032833/Copy_of_Ensemble.ipynb

Last backup

https://drive.google.com/file/d/1k1jD_sOWKLish2T3_pZoYpeE1DGwGfG3/view?usp=sharing

We got two reports that it throws out some errors now, and could stop working due to some changes Google made into Colabs this year)

You should be able to modify it to use with three models and different weights like 3, 2, 2 in example of Ensemble MDX-B (ONNX) + MVSep Vocal Model + Demucs4 HT on the old SDR chart (so it does not work like avg/avg in GUI).

Joining frequencies from two models

Sometimes it may happen that a regular ensemble even with min spec doesn't give you complete freedom over what you want to achieve, having one cleaner narrowband model result with fullband model result with more vocal residues, but you still want to have a full spectrum.

Instead of using ensemble Colab, you can also mix in some DAW, MDX-UVR 464/inst3 or Kim inst model result which have 17.7Hz cutoff, with HQ_1-5 or Demucs 4 result, which has full 22kHz training frequency model.

First, import both tracks. Now rather the most correct attitude to avoid any noise or frequency overlapping is to use [brickwall highpass](#) in EQ at 17680Hz everywhere on Demucs 4 stems, and leave MDX untouched, and just it. You can use GSEP instead of Demucs 4 (possibly less vocal residues).

If you want to experiment further, as for a cut-off, once I ended up with 17725.00 flat high pass with -12dB slope for "drums" in Izotope Elements EQ Analog and I left MDX untouched. "Bass" stem set to 17680.00 in mono and "other" in stereo at 17680.00 with Maximiser with IRC 1 -0.2, -2, th. 1.55, st. 0, te 0. But it might produce hissy hi-hat in places with less busy mix or when hi-hat is very fast, so tweak it to your liking.

For free EQ you can use e.g. TDR Nova - click LP and set 17.7 and slope -72dB. As a free DAW you can use free Audacity (new versions support VST) or Cakewalk, Pro Tools Intro, or Ableton Lite.

The result of above will probably cause a small hole in a spectrum, and a bit lack of clarity. Alternatively, you can apply resonant high pass instead of brickwall, so the whole will be filled without overlapping frequencies.

Instead, you can also consider using linear phase EQ/mode like in free Qrange and its high pass to potentially cause less problems in phase.

Similar method to this can also be used for *joining YT Opus frequencies above 15750Hz with AAC (m4a) files*, which gives more clarity compared to normal Opus on YT. Read [this](#).

DAW ensemble

Averaging

The counterpart of avg ensemble from UVR ([more](#)) can also be made in a DAW (Audacity/Cakewalk/Reaper etc.). When you drag and align all stems you want to ensemble in your DAW, you simply need to lower the volume of stems according to the number of imported stems to ensemble.

It's -3 dB per one stem for replicating avg spec, so for a pair you need to decrease the volume of two stems by 6 dB (possibly by 6.02 as well).

So, when you add another stem (so for 3 models ensemble), you need to decrease the volume of all stems by 9dB, and so on.

The other way round, it's 3dB decrease for all stems every time you import a new track.

Weighting manually (more precise)

You can change volume of stems to your liking, just to not cause clipping having too loud output on master fader once you play. You can circumvent the problem to some extent using a limiter on the sum, but it might be not necessary.

Also, you can use different volume automation of stems towards specific verses and choruses, or just different volume relation of stems whenever a new verse or chorus appear.

Note: You won't be able to use that method if one stem had phase rotated or flipped.

"I've made some tests by simply overlaying each audio above each other and reducing their volume proportionally of the number of audio overlays (like you would do in a DAW), it scores like ~0.0002 better SDR than UVR's average."

You can use Audacity online at wavacity.com, although it might crash occasionally while using on at least smartphone.

Bandlab is more stable while using as app: <https://www.bandlab.com/explore/home> but also crashes when used in PC mode online.

The app at least vertically doesn't show master fader, so you cannot control the output volume meter. Probably the same in horizontal view. Plus, the app doesn't give the possibility to adjust the gain precisely, e.g. to 9dB instead of -9,5dB, so to use single files with found gain values in Wavacity, to mix them without crashes, you can use the [manual ensemble Colab](#).

If you want to just listen to stems offline, change their volume, panning, solo, mute, you can download [this](#) html (by cypha_sarin) and run it locally in your browser.

Manual ensemble in UVR5 GUI from single models (e.g. from inference Colabs or online sites)

"You can use Colabs to make individual model separations and then use the manual ensemble tool from UVR5 GUI to merge them together (you don't need special CPU/GPU to use that tool and it's fast! 15-year-old computers can handle it).

In UVR GUI > process method = Audio Tools, then choose "Manual Ensemble" and the desired ensembling method."

Combine input is even more aggressive than Max Spec.

E.g. it takes two -15 ilufs songs, and makes pretty loud -10 ilufs result.

To potentially deal with harshness of such output, you can set quality in options to 64 bit (sic!), or possibly manually decrease volume of ensembled files before passing through UVR Combine Inputs.

Combine input was good for ensembling KaraFan results of preset with the least amounts of residues, and preset 5 for more clarity, but a bit more residues. The instrumental result was fuller sound, better snares and clarity.

The downside is, you cannot control gain of ensembled stems precisely like in DAW, or using Colab.

Model fusion

You can perform fusion of models using [ZFTurbo script](#) or [Sucial script](#) (they're similar if not the same). "I think the models need to have at least the same dim and depth, but I'm not sure about that" - mesk.

They allow creating one model out of weighted models with specified parameters, so only one model is needed to inference instead of two or more.

How to separate in UVR using multiple models in batch to compare the results for the best manual ensemble?

Simply use **Ensemble Mode**, but before, go to Settings>Additional Settings and enable "**Settings Test Mode**" (adds 10-digits to every separation file name, so you won't overwrite the result of the same models with different settings) and "**Model Test Mode**" (adds model name to every output file name, so the file won't get overwritten by any other model separation) and now go to Settings>Settings Guide>Choose Advanced Menu>**Ensemble Customization Options** and enable "**Save All Outputs**" (now when you choose models to

separate in Ensemble, intermediate files won't be deleted, so not only min/max/avg mag ensemble result file will be left, but also result of separation from single models which you can use later to check the result manually for [manual weighted ensemble](#) e.g. in DAW or in [Colab](#)).

UVR's VR architecture models

(settings and recommendations;
mostly outdated arch for all vocals and instrumental models)

VR Colab by HV

(old)

https://colab.research.google.com/github/NaJeongMo/Colaboratory-Notebook-for-Ultimate-Vocal-Remover/blob/main/Vocal%20Remover%205_arch.ipynb

Use [this](#) fixed notebook for now (04.04.23)

Sometimes Google Colab might break itself (e.g. error: No module named 'pathvalidate'), and then you can simply try to go to Environment and delete it entirely and start over, and then it might start working.

(since 17.03.23 the official link above for HV Colab stopped working (librosa, and later pysound related issues with again YT links, but somehow fixed) "!pip install librosa==0.9.1" in OG Colab fixes the issue and is only necessary for both YT and local files and clean installation works too.)

- HV also made a [new](#) VR Colab which irc, now don't clutter all your GDrive, but only downloads models which you use (but without VR ensemble) and probably might work without GDrive mounting.

(Google Colab in general allows separating on free virtual machine with decent Nvidia GPUs - it's for all those who don't want to use their personal computer for such GPU/CPU-intensive tasks, or don't have Nvidia GPU or decent CPU, or you don't want to use online services - e.g. frequently wait in queues, etc.)

Video tutorial how to use the VR Colab (it's very easy to use):

<https://www.youtube.com/channel/UC0NiSV1jLMH-9E09wiDVFYw>

You can use VR models in UVR5 GUI or

To use the above tool locally (old command line branch for VR models only):

<https://github.com/Anjok07/ultimatevocalremovergui/tree/v5-beta-cml>

Installation tutorial: <https://www.youtube.com/watch?v=ps7GRvl1X80>

In case of CUDA out memory error due to too long files, use Lossless-cut to divide your song into two parts,
or use this Colab which includes chunks option turned on by default (no ensemble feature here):

https://colab.research.google.com/drive/1UA1aEw8flXJ_JqGalgkwNIGw4l0gFmV?usp=sharing#scrollTo=l4B1u_fLuzXE

Below, I'll explain Ultimate Vocal Remover 5 (VR architecture) models only (fork of vocal remover by tsurumeso).

For more information on VR arch, see here for official documentation and settings:

<https://github.com/Anjok07/ultimatevocalremovergui/tree/v5-beta-cml>

<https://github.com/Anjok07/ultimatevocalremovergui>

The best

VR settings

Explained in detail

Settings available in [Colab](#) and in CLI branch, and also UVR 5 GUI (but without at least mirroring2. mirroring in UVR5 GUI for VR arch got replaced entirely by High End Process (works as mirroring now, and not like original High End Process which was originally dedicated for very old 16kHz VR models only).

These VR models can be used in this 1) [Colab](#) or in 2) [UVR5 GUI](#) or on 3) [mvsep.com](#) (uses 512 windows size, aggressiveness option, various models) 4) x-minus.pro/uvronline.app (for free one UVR (unreleased) model without parameters ("lo-fi" option, mp3, 17,7 kHz cutoff) [Demucs 4 for registered users iirc (site by Aufr33 - one of the authors of UVR5)])

I had at least one report that results for just VR models are better using Colab above/old CLI branch instead of the newest UVR5 GUI, so be aware (besides both mirroring settings - only mirroring is working under high-end process - no mirroring2 [272 window size is added back as user input] all settings should be available in GUI). Interestingly, I received similar report for MDX models in UVR5 GUI comparing to Colab (be aware just in case). The problems might be also bound to VRAM, and don't exist on 11GB GPPUs and up or in CPU mode.

Before we start -

Issue with additional vocal residues when postprocess is enabled

- “*postprocess* option masks instrumental part based on the vocals volume to improve the separation quality.” (from: <https://github.com/tsurumeso/vocal-remover>) where in HV Colab it says: “Mute low volume vocals”. So, if it enhances separation quality, then maybe it should cancel some vocals residues (“low volume vocals”) so that’s maybe not too bad explanation.

But that setting enabled in at least Colab may leave some vocal residues:

(it’s fixed in UVR GUI “the very end bits of vocals don’t bleed anymore no matter which threshold value is used”)

Customizable postprocess settings (threshold, min range and fade size) in HV’s Colab were deleted, and were last time available in this revision:

https://colab.research.google.com/github/NaJeongMo/Colaboratory-Notebook-for-Ultimate-Vocal-Remover/blob/b072ad7418f6b1825d3dcff7cef70c5b0985d540/Vocal%20Remover%205_arch.ipynb#scrollTo=CT8TuXWLBrXF

So change default 0.3 or 0.2 threshold value (depending on revision) and set it to 0.01 if you have the issue when using postprocess.

The *threshold* parameter set to 0.01 fixes the issue (so quiet the opposite thing happened using default settings than this option should serve to, I believe).

Also, default threshold values for postprocess changed from 0.3 to 0.2 in later revisions of the Colab.

- *Window size* option set to anything other than 512 somehow decrease SDR, although most people like lower values (at least 320, me even 272; 352 is also possible, but anything above changes the tone of sound more noticeably) - we don’t know yet why lower window sizes mess with SDR (similar situation like with GSEP) - 512 might be a good setting for ensemble with other models than VR ones or for further mastering. Sometimes compared to 512 windows size, 272 can lead to a bit more noticeable vocal residues. You might find bigger window sizes less noisy in general, but also more blurry for some people.

- *Aggressiveness/Aggression* - “A value of 10 is equivalent to 0.1 (10/100=0.1) in Colab”. Strangely, the best SDR for aggressiveness using MSB2 instrumental model turned out to be 100 in GUI, 10 in Colab, while we usually used 0.3 for this model and 500m_x as well, while HP models usually behaves the best with lower values than HP2 models (0.09/10 in GUI).

- Mirroring turned out to enhance SDR. It adds to the spectrum e.g. above 20kHz for a base training frequency of VR model (all 4 bands).

none - No processing (default)

bypass - This copies the missing frequencies from the input.

mirroring - This algorithm is more advanced than correlation. It uses the high frequencies from the input and mirrored instrumental's frequencies. More aggressive.

mirroring2 - This version of mirroring is optimized for better performance.

--high_end_process - In the old CLI VR, this argument restored the high frequencies of the output audio. It was intended for models with a narrow bandwidth - 16 kHz and below (the oldest "lowend" and "32000" ones, none more). But now in UVR5 GUI, High-end process is counterpart of mirroring.

(current 500MB models don't have full 22kHz coverage, but 20kHz, so consider using mirroring instead or none if you want fuller spectrum)

- Be aware, that even for VR arch, the same rule for GPUs with less than 8GB VRAM applies (inb4 - Colab T4 has 15GB) - separations on 6GB VRAM have worse quality with the same parameters. In order to work around the issue, you can split your audio into specific parts (e.g. for all chorus, verses etc).

VR models settings and list

For VR architecture models, you can start with these two fast models:

Model: **HP_4BAND_3090_arch-124m** (1_HP-UVR)

1) Fast and reliable. V2 below has more "polished" drums, while here they're more aggressive and louder. Sometimes V2 might be safer and can fit in more cases where it's not hip-hop and music is not drum oriented, but that one rarely harms some instruments more in certain cases with more busy mix with e.g. repeatable synth. You may want to isolate using these two models and pick the best results on even the same album.

Windows size: 272

Aggressiveness: 0.09 (9 in GUI)

TTA: ON (OFF if snare is too harsh)

Post-processing: OFF (at least for this model - it can get muffle instruments in background beside drums of the track in some cases, e.g. guitar)

"Mirroring" (Hi-end process in GUI) (rarely "Mirroring2" here, since the model itself is less smooth and usually have better drums, but it sometimes leads to overkill - in that case check mirroring2 in CLI or V2 model above)

Better yet, to increase the quality of the separation (when drums in e.g. hip-hop can be frequently damaged too much during the process) go now straight to the Demucs section and read the "Anjok's tip".

If you have too many vocal residues vs 500m_1 model, increase aggressiveness from 0.09 to 0.2 or even 0.3, but it's destructive for some instruments (at least without Demucs trick above).

Model: **HP-4BAND-V2_arch-124m** (2_HP-UVR)

!) Fast and nice model, but sometimes gives lots of vocal residues comparing to above, but thanks to this, it may sometimes harm snare less in some cases (still 4 times faster than 500m_1) it's ~55/45 which model is better and depends on the album even on the same genre:

Window size: 272 (the lowest possible; in some very rare cases it can spoil the result on 4 band models, then check 320)

Aggressiveness: 0.09 (9 in GUI)

TTA: ON (instr. separation of a better quality)

Postprocess: (sometimes on, it rather compliments to the sound of this model especially when the result sounds a bit too harsh, but it also can spoil drums in some places when e.g. strong synths suddenly appear in mix for short, probably misidentifying them as vocals, so be aware)

Mirroring (it fits pretty well to this model in comparison to mirroring2 which is not "aggressive" enough here) [mirroring doesn't seem to be present in GUI so be aware)

Processing time for this model is 10 minutes using the weakest GPU in Colab (but currently you should be getting better Tesla T4).

(for users of x-minus) "slightly different models [than in GUI] are used for minimum aggressiveness. When we train models, we get many epochs. Some of these models differ in that they better preserve instruments such as the saxophone. These versions of the models don't get into the release, but are used exclusively on the XM website."

Model: **HP2-4BAND-3090_4band_arch-500m_1** (9_HP2-UVR)

3) Older good model, but resource heavy - check it if you get too many vocal residues, or in other cases - when your drums are too muffled - rarely there might be more bleeding and generally more spoiled other instruments in comparison to those above, it depends on a track. In some cases it bleeds vocal less than HP_4BAND_3090_arch-124m

Window size: 272

Aggressiveness: 0.3-0.32 (30-32 in GUI)

TTA: ON

Postprocess: (turned ON in most cases with exceptions (it's polishing high-end), and the problem with muffling instruments using ppr doesn't seem to exist in this model)

Mirroring2 (I find mirroring[1] too aggressive for this model, but with exceptions)

! Be aware these settings are very slow (40 minutes per track in Colab on the former default K80 GPU, but it's faster now) so just in case, you might want to experiment with 320/384, or at worse even 512 window size if you want to increase processing speed in cost of isolation precision.

Colab's former default Tesla K80 processes slower than even GTX 1050 Ti, so if you have a decent Nvidia GPU, consider using UVR locally. Since May 2022 there is faster Tesla T4 available as default, so there shouldn't be any problem.

HP2-4BAND-3090_4band_arch-500m_2 (8_HP2-UVR)

was worse in I think every case I tested, but it's good for a pair for ensemble (more about ensemble in section below).

Model: HP2-MAIN-MSB2-3BAND-3090_arch-500m (7_HP2-UVR.pth)

4) Last resort, e.g. when you have a lot of artifacts (heavily filtered vocal residues) some instruments spoiled, and no equal sound across the track. Last resort, because it's 3 band, instead of 4 band, and it lacks some hi-end/clarity, but if your track is very demanding to filter out vocal residues, then it's good choice. The best SDR among VR-arch models.

Window size: 272

Aggressiveness: 0.3

TTA: ON

Postprocess: ON

Mirroring

It's similarly nightmarishly slow in Colab just like 500m_1/2 using these settings (1 hour for a track on K80) when you got accidentally slower Tesla K80 assigned in Colab instead of Tesla T4.

HighPrecision_4band_arch-124m_1

*)

May sometimes harm instruments less than HP_4BAND_3090_arch-124m, but may leak vocals more in many cases, but generally instrumentals lacks some clarity, but it sounds more neutral vs 500m_1 with mirroring (not always an upside). It's not available in GUI by default due to its not fully satisfactory results vs models above.

Window size: 272

Aggressiveness: 0.2

TTA: ON

Postprocess: off

mirroring

SP in the GUI models stands for "Standard Precision". Those models use the least amount of computing resources of any other models in the application. HP on the other hand stands for "Higher Precision" those models use more resources but have better performance.

So, what's the best VR arch model?

I'd stick to **HP_4BAND_3090_arch-124m** (1_HP-UVR) if it only gives good result for your song (e.g. hip-hop). If you're forced to use any other VR model for a specific song due to unsatisfactory results with this model, then probably current MDX models will achieve better results.

Second most usable model for me was 500_m1(9_HP2), and then

HP-4BAND-V2_arch-124m (2_HP-UVR) or something in between, but compared to MDX-UVR models, it might be not worth to use it anymore due to possibility of more vocal residues.

- 13/14_SP models (called 4-band beta 1/2 in the Colab) - less aggressive than above (these are older UVR5 models by UVR team - less aggressive, give more vocal residues frequently' the mid ones have less clarity, but might be less noisy - but they're surpassed by MDX models)

- v4 models -

Even older models from times of previous VR codebase

"All the old v5 beta models that weren't part of the main package are compatible [with UVR] as well. Only thing is, you need to append the name of the model parameter to the end of the model name"

Also, V4 models are still compatible with UVR using this method.

Main Models

MGM_MAIN_v4_sr44100_hl512_nf2048.pth -

This is the main model that does an excellent job removing vocals from most tracks.

MGM_LOWEND_A_v4_sr32000_hl512_nf2048.pth -

This model focuses a bit more on removing vocals from lower frequencies.

MGM_LOWEND_B_v4_sr33075_hl384_nf2048.pth -

This is also a model that focuses on lower end frequencies, but trained with different parameters.

MGM_LOWEND_C_v4_sr16000_hl512_nf2048.pth -

This is also a model that focuses on lower end frequencies, but trained on a very low sample rate.

MGM_HIGHEND_v4_sr44100_hl1024_nf2048.pth -

This model slightly focuses a bit more on higher end frequencies.

MODEL_BVKARAOKE_by_auf33_v4_sr33075_hl384_nf1536.pth -

This is a beta model that removes main vocals while leaving background vocals intact.

Stacked Models

StackedMGM_MM_v4_sr44100_hl512_nf2048.pth -

This is a strong vocal artifact removal model. This model was made to run with MGM_MAIN_v4_sr44100_hl512_nf2048.pth -

However, any combination may yield a desired result.

StackedMGM_MLA_v4_sr32000_hl512_nf2048.pth -

This is a strong vocal artifact removal model. This model was made to run with MGM_MAIN_v4_sr44100_hl512_nf2048.pth -

However, any combination may yield a desired result.

StackedMGM_LL_v4_sr32000_hl512_nf2048.pth -

This is a strong vocal artifact removal model. This model was made to run with MGM_LOWEND_A_v4_sr32000_hl512_nf2048.pth -

However, any combination may yield a desired result.”

VR ensemble settings

As for VR architecture, ensemble is the most universal and versatile solution for lots of tracks. It delivers, when results achieved with single models fail - e.g. when snare is too muffled or distorted along with some instruments, but sometimes a single model can still provide more clarity, so it's not universal for every track.

In most cases, ensemble of only VR models is dedicated for the tracks when in the most prevailing moments of busy mix in the track, you don't have major bleeding using single VR model(s) because it rarely removes that well vocal residues from instrumentals better than current MDX models, or with high aggressiveness it becomes too destructive.

Order of models is crucial (at least in the Colab)! Set the model with the best results as the first one. Usually, using more than 4 models has a negative impact on the quality. Be aware that you cannot use postprocess in HV Colab in this mode, otherwise you'll encounter an error. *Please note that now UVR 5 GUI allows an ensemble of UVR and MDX models in the app exclusively, so feel free to check it too.* Here you will find settings for “only” UVR models ensemble only.

- HP2-4BAND-3090_4band_arch-500m_1.pth (9_HP2-UVR)
 - **HP2-4BAND-3090_4band_arch-500m_2.pth (8_HP2-UVR)
 - HighPrecision_4band_arch-124m_1.pth (probably deleted from GUI, and you'd need to copy this model from [here](#) to your GUI folder manually - if it will only work)
 - HP_4BAND_3090_arch-124m.pth (1_HP-UVR)
- (order in Colab is important, keep it that way!)

Or for less bleeding, but a bit more muffled snare, use this one instead:
HP-4BAND-V2_arch-124m.pth (model available only in Colab, recommended)

*on slower Tesla K80 you can run out of time due to runtime disconnection, but you should get faster Tesla T4 by default on first Colab connection on the account in 24h.

Aggressiveness: 0.1 (pretty universal in most cases, 0.09 rarely fits).

Or for more vivid snare if bleeding won't kick in too much: 0.01 (in cases when it's more singing than rapping - for the latter it can result in more unpleasant bleeding (or just in some parts of the track). Suggested very low aggressiveness here doesn't leak as much as it could using the same settings on a single model, but it leaks more in general vs single models' suggested settings).

0.05 is not good enough for anything AFAIK.

high_end_process: mirroring2 (just ON in GUI)

(for less vivid snare check "bypass", (not "mirroring" for ensemble - for some reason both make the sound more muffled), be aware that bypass on ensemble results with less vocal leftovers)

ensembling_parameter: 4band_44100.json

TTA: ON

Window size: 272

FlipVocalModels: ON

Ensemble algorithm: default on Colab (min_mag for instrumentals)

Other ensemble settings

- For clap leftovers in vocal stem, check out [this](#) ensemble settings.
- For creaking sounds, process your separation output more than once till you get there with [this](#) setting
- Also reported clean instrumentals with [this](#) setting

Make sure you checked separated file after the process and file length agrees with original file. Occasionally, the result file can be cut in the middle, and you'll need to start isolation again. Also, you can accidentally start isolation before uploading of source file is finished. In that case, it will be cut as well.

It takes 45 minutes using Tesla T4 (~RTX 3050 in CUDA benchmarks) for these 4 models settings. Change your songs for processing after finishing the task FAST, otherwise you'll be disconnected from runtime when the notebook is idle for some time (it can even freeze in the middle).

In reality, Tesla T4 maybe has much more memory, but what takes 30 minutes on a real RTX 3050, here might take even more than 2 hours and sometimes slower or sometimes slightly faster (usually slower). So you're warned.

**Be aware that these 4 model ensemble setting with both 500m models in most cases won't suffice for the slowest (and no longer available in 2023) Tesla K80 due to its time and performance limit to finish such a long operation which exceeds 2 hours (it takes around

02:25h). Certain tasks too much above 2 hours ends up with runtime disconnection, so you're warned.

Also be aware that the working icon of Colab on the opened tab sometimes doesn't refresh when operation is done.

Furthermore, it can happen that the Colab will hang near 01:45-02:17h time of executing the operation. To proceed, you can click F5 and press cancel on prompt to whether to refresh. Now the site will be functional again, but the process will stop without any notice. It is most likely the same case when you suddenly stop connection to the internet, and the process will still run virtually till you reconnect to the session. But here, you just don't have to click the reconnect button on the right top. Most likely you have very limited time to reestablish the connection till the process will stop permanently if you don't connect on connection lost (or eventually if progress tracker/Colab will stop responding). So in the worst case, you need to observe if the process is still working between 01:45-02:17h of processing. If you see that your GPU has 0.84GB instead of ~2GB, you're too late and your process is permanently interrupted, and the result is gone. It's harder to track how long it processes when you already used the workaround once, and the timer stopped, so you don't know how long it is separating already.

Limit for faster Tesla T4 is between 1:45 and 2:00h/+ (sometimes 2:25, but can disconnect sooner, so try not to exceed two hours) of constant batch operation, which suffice for 2 tracks being isolated using ensemble settings above with both 500m models (rarely 3 tracks).

HP2-4BAND-3090_4band_arch-500m_1 (9_HP2-UVR) - I think it tends to give the most consistent results for various songs (at least for songs when vocal residues are not too prevalent here)

HP-4BAND-V2_arch-124m (2_HP-UVR) - much faster and can give crisp results, but with too many vocal residues for some songs (like VR arch generally tends to)

HP_4BAND_3090_arch-124m (1_HP-UVR) - something between the two above, and can give the best results for some song too (out of other VR models)

HP2-MAIN-MSB2-3BAND-3090_arch-500m (7_HP2-UVR.pth) - tends to have the least vocal residues out of the VR models listed above, but in cost of instrumentals not sounding so "full"

HighPrecision_4band_arch-124m_1 (I think not available in UVR, you'd need to install it manually) - can be a good companion if you only have VR models for ensemble

HP2-4BAND-3090_4band_arch-500m_2 (9_HP2-UVR) - the same situation, I think it rarely gives any better results than 500m_1 (if in even any case) but it's good for purely VR ensemble

_____ VR algorithms of ensemble _____

by サナ(Hv#3868)

"np_min takes the highest value out, np_max does vice versa

it's also similar to min_mag and max_mag

So the min_mag is better for instrumental as you could remove artefacts.

comb_norm simply mixes and normalizes the tracks. I use this for acapella as you won't lose any data this way"

Batch conversion on UVR Colab

There's a "ConvertAll" batch option available in Colab. You can search for "idle check" in this document to prevent disconnections on long Colab sessions, but at least if you get the slowest K80 GPU, the limit is currently 2 hours of constant work, and it simply terminates the session with GPU limit error. The limit is enough for 5 tracks - 22 minutes with ~+-3m17s overhead (HP_4BAND_3090_arch-124m/TTA/272ws/noppr/~2it/s) so better execute bigger operations in smaller instances using various accounts and/or after 3-5 attempts you can also finally hit on better GPU than K80.

To get faster GPU simply go to Runtime>Manage session>Close and connect and execute Colab till you get faster Tesla T4 (up to 5 times). But be aware, that 5 reconnections will reach the limit on your account, and you will need to change it. It's easier to get T4 and not reach the limit reconnecting, around 12:00 CET in working days. 14:30 o'clock it was impossible to get T4, but probably it depended on a situation when I already used T4 this day since I received it immediately on another account.

For single files isolation instead of batch convert I think it took me 6-7 hours till the GPU limit was reached, and I processed 19 tracks using 272 ws in that session.

JFI: Even 5800X is slower than the slowest Colab GPU.

Shared UVR installation folder among various Google accounts

Since we no longer can use old Gdrive mounting method allowing mounting the same drive across various Colab sessions - to not clutter all of your accounts by UVR installation, simply share a folder with editing privileges and create a shortcut from it to your new account. Sadly the trick will work for one session at a time.

Firstly - sometimes you can have problems with opening the shared folder on proper account despite changing it after opening the link (it may leave you on old account anyway). In that case, you need to manually insert id of your account where you want to open your link to.

E.g. <https://drive.google.com/drive/u/9/folders/xxxxxxxx> (where 9 is an example of your account ID which shows right after you switch your account on main Google Drive page).

After you opened the shared UVR link on your desired account, you need to add the shortcut to your disk (arrow near folder's name) and when it's done, create "track" and "separated" folder on your own - so delete/rename shared "tracks" and "separated" folder and create it manually, otherwise you will get error during separation. If you still get an error anyway, refresh file browser in the left of Colab and/or retry running separation three times till error disappears (from now on it shows error occasionally, and you need to retry from time to time

and/or click refresh button in file manager view in the left or even navigate manually to tracks folder in order to refresh), Colab gets changes like moving files and folders on your disk with certain delay. And be aware that most likely such way of installing UVR will prevent you from any further updates from such account with shared UVR files, and on the account you shared the UVR files from, you need to repeat folder operations if you will use it back again on Colab.

Comparing 500m_1 and arch_124m above, in some cases you can notice that the snare is louder in the first, but you can easily make it up using mirroring instead of mirroring2. Downside of normal mirroring might be more pronounced vocal residues due to higher output frequency.

Also, in 500m_1 more instruments are damaged or muffled, though more aggressiveness in the default setting of 500m_1 sometimes makes an impression that more vocal residues are cancelled.

([evaluation tests](#) window size 272 vs 320 -

it's much slower, doesn't give noticeable difference on all sound systems, 272 got slightly worse score, but based on my personal experience I insist on using 272 anyway)

([evaluation tests](#) aggressiveness 0.3 vs 0.275 -

doesn't apply for all models - e.g. MGM - 0.09)

([evaluation tests](#) TTA ON vs OFF -

in some cases, people disable it)

5a) (haven't tested thoroughly these aggressiveness parameters yet)

HP2-4BAND-3090_4band_arch-500m_1.pth

w 272 ag 0.01, TTA, Mirroring

5c)

HP2-4BAND-3090_4band_arch-500m_1.pth

w 272, ag 0.0, TTA, Mirroring 2

Low or 0.0 aggressiveness leaves more noise, sometimes it makes instrumental cleaner, if you don't care for more vocal bleeding (it depends also on your sound system how you are able to catch them. E.g. whether you listen on headphones or speakers).

But be aware that:

"A 272 window size in v5 isn't recommended [in all cases]. Because of the differing bands. In some cases it can make conversions slightly worse. 272 is better for single band models (v4 models) and even then the difference is tiny" Anjok (developer)

(so on some tracks it might be better to use 320 and not below 352, but personally I haven't found such case yet)

DeepExtraction is very destructive, and I wouldn't recommend it with current good models.

Karokee V2 model for UVR v5 (MDX arch)
(leaves backing vocals, 4band, not in Colab yet, but available on MVSep)

Model:

https://mega.nz/file/yJIBXKxR#10vw6IRJmHRe3CMnab2-w6gAk-Htk1kEhlp_qQGCG3Y

Be sure to update your scripts (if you use older command line version instead of GUI):
<https://github.com/Anjok07/ultimatevocalremovergui/tree/v5-beta-cml>

Run:

```
python inference.py -g 0 -m modelparams\4band_v2_sn.json -P  
models\karokee_4band_v2_sn.pth -i <input>
```

5d) Web version for UVR/MDX/Demucs (alternative, no window size parameter for better quality):

<https://mvsep.com/>

How to use this free online stem splitter with a variety of quality algorithms -

1. Put your audio file in.
2. Choose an algorithm. Usually, you really only need to choose one of two algorithms:
 - The best algorithm for getting clean vocals/instrumental is selecting Ultimate Vocal Remover. Once you selected Ultimate Vocal Remover, select HP-4BAND-V2 as the "Model type".
 - The best algorithm for getting clean separate instrument tracks, like bass, drums and other, is Demucs 3 Model B.
3. Hit Separate, and mvsep will load it for you. This means you can do everything yourself, no need to ask for other people's isolations if you can't find them.

6) VR 3 band model (gives better results on some songs like K Pop)

[HP2-MAIN-MSB2-3BAND-3090](#)

(I think default aggressiveness was 0.3)

7) deprecated - in many cases lot of bleeding (not every time) but in some cases it hurts some instruments less than all above models (e.g. quiet claps).

MGM-v5-4Band-44100-BETA2/
(MGM-v5-4Band-44100-_arch-default-BETA2)
/BETA1
Agg 0.9, TTA, WS: 272

Sometimes I use Lossless-Cut to merge beta1 and beta2 certain fragments.

Models from point 4 surpasses ensemble of both BETA1 and BETA2 models.

(!) Interesting results (back in 2021)

"Whoever wants to know the HP1, HP2 plus v4 STACKED model method, I have a [...] group explaining it"

<https://discord.gg/PHbVxrV4yS>

Long story short - you need to ensemble HP1 and HP2 models, then on top of it, apply stacked model from v4.

Be aware that ensemble with postprocessing in Colab doesn't work.

Instruction:

1 Open this link

<https://colab.research.google.com/drive/189nHyAUfHIfTAXbm15Aj1Onlog2qcCp0?usp=sharing>

2. Proceed all the steps

3. After mounting GDrive upload your, at best, lossless song to GDrive\MDX\tracks

4. Uncheck download as MP3, begin isolation step

5. Download the track from "separated" folder on your GDrive. You can use GDrive preview on the left.

1*. Alternatively, if you have a paid account here, upload your song to:

<https://x-minus.pro/ai?hp>

Make sure you have "mdx" selected for the AI Model option. Wait for it to finish processing.

2*. Set the download format to "wav" then click "DL Music." Store the resulting file in the ROOT of your UVR installation.

6. Use a combination of UVR models to remove the vocals. Experiment to see what works with what. Here's a good starting point:

HP2-4BAND-3090_4band_arch-500m_1.pth

HP2-4BAND-3090_4band_arch-500m_2.pth

HP_4BAND_3090_arch-124m.pth

HP-4BAND-V2_arch-124m.pth

7. Store the resulting file in the ROOT of your UVR installation alongside your MDX result.

8. Finally, ensemble the two outputs together. cd into the root of your UVR installation and invoke spec_utils.py like so:

\$ python lib/spec_utils.py -a crossover <input1> <input2>

the output will be stored in the ensembled folder

9* (optional). Ensemble the output from spec_utils with the output from UVR 4 stacked models using the same algorithm

Ensemble

spec_utils.py allowing ensemble is standalone, and doesn't require UVR installed in order to work. It accepts any of the audio files

mul - multiplies two spectrograms

crossover - mixes the high frequencies of one spectrogram with the low frequencies of another spectrogram

Default usage from aufr33:

```
python lib/spec_utils.py -o inst_co -a crossover UVR_inst.wav MDX_inst.wav
```

https://github.com/Anjok07/ultimatevocalremovergui/blob/v5-beta-cml/lib/spec_utils.py

Custom UVR Piano Model:

https://drive.google.com/file/d/1_GEEhvZj1qyIod1d1MX2IM6u65CTpbmI/view?usp=s

VR Colab troubleshooting

If you somehow can't mount GDrive in the VR Colab because you have errors or your separation fails:

- Use the same account for Colab and for mounting GDrive (or you'll get an error)
- If you're on mobile, you might be unable to use Colab without PC mode checked in your browser settings (although now it works without it in Chrome Android)
- In some cases, you won't be able to write "Y" in empty box to continue on first mounting on some Google account. In that case, e.g. change browser to Chrome and check PC mode.
- In some cases, you won't be able to paste text from clipboard into Colab if necessary, when being in PC mode on Android, if some opened on-screen applications will prevent the access - you'll need to close them, or use mobile mode (PC mode unchecked)
- (probably fixed) If you started having problems with logging into Colabs.
> Actually, it doesn't show that you're logged in while the button says to log in.
So, it should respect redirections in Colab links to specific accounts, but if you're mounting to GDrive, and it fails with Colab error, simply click the button in the top right corner to log in. It will. Just won't show that you did that. Then Colab will start working.
- Don't use postprocess in ensemble, or you'll encounter error

- You can try checking force update in case of errors
- Go to runtime>manage sessions>terminate session and then try again with Trigger force update checked (ForceUpdate may not work before terminating session after Colab was launched already).
- Make sure you got 4.5GB free space on GDrive and mounting method is set to "new". You can try out "old" but it shouldn't work.
Try out a few times.
- If still nothing, delete VocalRemover5-COLAB_arch folder from GDrive, and retry without Trigger update.

On fresh installation, make sure you still have 4.5GB space on GDrive (empty recycle bin - automatic successful models installation will leave separate files there as well, so you can run out of space on cluttered GDrive easily)

- If still nothing (e.g. when models can't be found on separation attempt), then download that thing, and extract that folder to the root (main) directory of Gdrive, so it looks like following: Gdrive\VocalRemover5-COLAB_arch and files are inside, like in the following link:
https://drive.google.com/drive/folders/1UnjwPIX1uc9yrqE-L64ofJ5EP_a8X407?usp=sharing
and then try again running the Colab:
<https://colab.research.google.com/drive/16Q44VBJilrXOgTINztVDVeb0XKhLKHwl>
- if you cannot connect with GPU anymore and/or you exceeded your GPU limit
try to log into another Google account.
- Try not to exceed 1 hour when processing one file or one batch of files, otherwise you'll get disconnected.
- Always close the environment in Environment before you close the tab with the Colab. That way, you will be able to connect to the Colab again after some time, even if you previously connected to the runtime and stopped using it. Not shutting down the runtime before exit, makes it wait in idle, and hitting timeout. Then the error of limit reached will appear after you'll try to connect to Colab again if it wasn't closed before. Then you'll need to wait up to 24h, or switch Colab account, while using the same Google account as for Colab in the mounting cell (otherwise, it will end up with error when you'll use different account for Colab and different for GDrive mounting).
- New layer models may not work with 272 window size causing following error:
“raise ValueError('h1_shape[3] must be greater than h2_shape[3]')
ValueError: h1_shape[3] must be greater than h2_shape[3]”

- (fixed) Sometimes on running mounting cell you can have short “~from Google Colab error” on startup. It will happen if you didn’t log into any account in the top right corner of the Colab. Sometimes it will show a blue “log in” button, but actually it’s logged in, and Colab will work.

- *A network error occurred, and the request could not be completed.*

GapiError: A network error occurred and the request could not be completed.

In order to fix these errors in Colabs, go to hosts file in your
c:\Windows\System32\Drivers\etc\hosts and check if you don't have any lines looking like:
127.0.0.1 clients.google.com
127.0.0.1 clients1.google.com etc.
It can be introduced by RipX Pro DAW.

- Various Colabs might occasionally get unstable, and the environment disk might get unmounted, or you might get weird errors. In that case, simply kill the current environment and start over

- These are all the lines which fix problems in our VR Colabs since the beginning of the year when new versions of these dependencies became incompatible (but usually one Colab linked is forked when told and up-to-date with these necessary fixes applied already)

```
!pip install soundfile==0.11.0
!pip install librosa==0.9.1
!pip install torch==1.13.1
!pip install yt-dlp=2022.11.11
!pip install git+https://github.com/ytdl-org/ytdl-nightly.git@2023.08.07
```

Later in February 2024 we needed to switch to older Python 3.8 in order to make numpy work correctly with used deprecated functions. More details on these fixes and used lines below [Similarity Extractor](#) section (all those fixes should be already applied in the latest fixed Colab at the top).

MDX-Net [Colab](#) by HV (March 2025)
Models trained by UVR team models (aufr33 & Anjok)

First vocal models trained by UVR for MDX-Net arch:

*9.703 model is UVR-MDX-NET 1, UVR-MDX-NET 2 is UVR_MDXNET_2_9682,
NET 3 is 9662, all trained at 14.7kHz*

(instrumental based on processed phase inversion)

List of all (newer) available MDX models at the very top.

I think main was 438 in UVR 5 GUI at some point. At least now it's simply main_438 (if it wasn't from the beginning, but it was easy to confuse it with simply main model or even inst main)

(use MDX is a way to go now over VR) Generally use MDX when the results achieved with VR architecture are not satisfactory - e.g. too much vocal bleeding (e.g. in deep and low voices) or damaged instruments. If you only want acappella - it's currently the best solution. Actually the best in most cases now.

MDX-UVR models are also great for cleaning artifacts from inverts (e.g. mixture (regular track) minus official instrumental or acappella).

(outdated) 9.682 might be better for instrumentals and inversion in some cases, while 9.7 for vocals, but better check already also newer models like 464 from KoD update (should be better in most cases) and also check Kim Model in GUI.

Generally on MVSEP's multisong dataset, these models received different SDR than on MDX21 dataset back in the days.

On MVSEP there's 9.7 (NET 1) model, and it doesn't have any cutoff above training frequency for inverted instrumentals like currently GUI has. For (new) model it's vocal 423 model and possibly with Demucs 2 enabled like in Colab, but it doesn't have a specific jaggy spectrum above MDX training frequency which is specific to inverted vocal 4XX models from that period including Kim's model.

Non-onnx version of voc_ft model in pth by MusicMan - 20x faster on MPS devices:

<https://discord.com/channels/708579735583588363/887455924845944873/1204148534790852608> (roughly the same model size)

It won't work in UVR. Inference code mirror:

<https://drive.google.com/file/d/1aSe0bwglWhR7vvF1aoHQICHPj39Kd-YK/view?usp=sharing>

Mirror:

https://drive.google.com/drive/folders/16QbwuCBT0_w9nmNDq22m1niq0odtaZUP?usp=sharing

And the rest of MDX-Net v2 models: HQ_1-5, inst3, Kim inst, Kim Vocal 1-2, and older narrowband vocal and instrumental ones and Karaoke models.

(the old) Google Colab by HV

(with OG demucs 2 ensembling for vocal models)

<https://colab.research.google.com/drive/189nHyAUfHifTAXbm15Aj1Onlog2qcCp0?usp=sharing>

Add separate cell as following, or else it won't work

!pip install torch=1.13.1 (probably numpy 1.25 for this old Torch)

If you're still getting errors, delete whole MDX_Colab folder, terminate your session, make clean installation afterward, and don't forget to have this torch line executed after mounting (that might happen in case you manually replaced model.py with some of the ones below, and didn't restore the correct old one).

(The Colab to use MDX easily in Google's cloud. Newer models not included, and it gives error if you add other models manually - custom models.py necessary, only 9.7 [NET 1-3] and karaoke models included above)

(In case of "RuntimeError: Error opening 'separated/(trackname)/vocals.wav': System error." simply retry)

More MDX models explained in UVR section in the beginning of the document since they're a part of UVR GUI now.

Optionally, 423 model can be downloaded separately [here](#) (just in case, it's main). It is on MVSEP as well.

(defunct) Upd. by KoD & DtN & Crusty Crab & jarredou, HV (12.06.23)

(probably now requires !pip install numpy==1.26 and restarting env)

It might have more models than above (e.g. some beta HQ ones)

The newest MDX Colabs - now with automatic models downloading (no more manual GDrive models installation as in older updates). Consider everything in the divided section later below as unnecessary.

https://colab.research.google.com/github/kae0-0/Colab-for-MDX_B/blob/main/MDX_Colab.ipynb (stable, lacks voc_ft batch process + also manual parameters loading per model like in the two above)

https://colab.research.google.com/github/jarredou/Colab-for-MDX_B/blob/main/MDX_Colab.ipynb (Beta. Might lack HQ_3 and voc_ft. It supports batch processing. Works with a folder as input and will process all files in it.)

In "tracks_path" must be a folder containing (only) audio files (not the direct link to a file). But the below might still work.)

<https://colab.research.google.com/drive/1CO3KRvcFc1EuRh7YJea6DtMM6Tj8NHoB?usp=sharing> (older revision with also auto models downloader, but with manual n_fft dim_f dim_t parameters setting like HV added)

and working one by HV linked at the top:

https://colab.research.google.com/github/NaJeongMo/Colab-for-MDX_B/blob/main/MDX-Net_Colab.ipynb

(new one by HV with community edits - 2025)

Old update from before model downloader implementation (May which year?)

[MDX Colab](#) with separate input for 3 models parameters, so you don't need to change models.py every time you switch to some other model. Settings for all models listed in Colab. From now on, it uses reworked main.py and models.py downloaded automatically (made by jarredou). Don't replace models.py from below packages with models from now on. Now denoiser also optionally added.

(older Colab instruction)

To use more recent MDX-UVR models in Google Colab:

- 1) Use and install this [Colab \(new\)](#) to GDrive at least once, run all the cells, nothing more - if you used MDX HV Colab (the one in the section above) on your specific Google Drive account before, ignore this step.
- 2) Copy these files to onnx folder in MDX_Colab on your GDrive (inst1-3, 427) (down)
https://drive.google.com/drive/folders/13SsV7b_kC6SqkICeX5wKhx-Z05uC8dLI
(down)
- 3) Overwrite models.py in MDX_Colab folder by provided below (not for new Colab)
(compatible with inst1-3, 427, Kim vocal and other)
<https://cdn.discordapp.com/attachments/945913897033023559/1036947933536473159/models.py> (completely different one with self.n_fft set to 7680 - incompatible with NET-1/9.x and 496 models)
- 4) Use this notebook with added models
(the same as the link in point 1):
<https://colab.research.google.com/drive/1zx7DQM-W9i7MJuEu6VTYz1xRG6IKRKVL?usp=sharing>
- 5) For Kims vocal model (poor instrumentals on Colab and no cutoff after inversion)
copy vocals.onnx
(use the same models.py from point 3):
<https://drive.google.com/drive/folders/1exdP1CkpYHUUksaz-qApS-0O1EtB0S82?usp=sharing>
to onnx subfolder named "[MDX-UVR-Kim Vocal Model \(old\)](#)"
- 6) For 496 inst model (inst main/MDX 2.1) go to the link below and put the model to onnx subfolder named "MDX-UVR Ins Model 496 - inst main-MDX 2.1" but you must replace attached models.py in the link in your GDrive (it's from the OG HV Colab), and it is incompatible with the rest of the models in this new Colab - make a copy/rename the previous models.py in order to go back to it
(496 model is not as effective as 464/inst3 leaving more vocal residues in some cases, but might work well in specific scenarios). 496 is the only model requiring the

old models.py from 9.7/NET1-3 models (attached below).

https://drive.google.com/drive/folders/1il_Zvc506xUv_58_GPHfVpxmCIDfGhx?usp=share_link (if you place model in the wrong place, you'll get missing vocals.onnx error [e.g. wrong folder structure or name] or "Got invalid dimensions for input: input for the following indices index: 2 Got: 3072 Expected: 2048." [when having wrong models.py])

- 7) Demucs turned on works only with default mixing algorithm and vocal models (or else you'll get "ValueError: operands could not be broadcast together with shapes (8886272,2) (8886528,2)"). Also, chunks might have to be decreased.
- 8) Be aware that after following these steps if you launch the old HV Colab above, it may overwrite models.py by the old one in point 6, which is compatible only with inst main/496 or full band models, so you'll need to repeat step 3 or 10 in case of invalid dimensions error or cutoff of full band model.
- 9) In case of runtime error, to use Kim model decrease chunks from 55 to 50, and for Demucs on, decrease it to 40 (or respectively even lower)
- 10) (beta) Full band beta 292 model (with new, only working for that model, models.py file with self.n_fft changed to 6144).

Go to the link below, copy model file to onnx subfolder called "MDX-UVR Ins Model Full Band 292" as in the link, and replace models.py (ideally make a backup/rename the old one in order to use previous models)

Thanks for help to Kim

https://drive.google.com/drive/folders/1CTJ6ctlr_awwudua1qJJMPAd7OrS2yO?usp=sharing

- 11) (beta) Full band beta 403 model (with the same modified models.py for these two models)

Copy model file to:

Gdrive\MDX_Colab\onnx\MDX-UVR Ins Model Full Band 403\" as in the link below, and replace models.py in Gdrive\MDX_Colab

https://drive.google.com/drive/folders/1UXPxQMVAocpyDVb3agXu0Ho_vqFowHpA?usp=sharing

- 12) (final) Full band 450/HQ_1 model (with the same modified models.py for the full band models)

Copy model file to:

Gdrive\MDX_Colab\onnx\MDX-UVR Ins Model Full Band 450 (HQ_1)\\" as in the link below, and replace models.py in Gdrive\MDX_Colab (if you didn't already for full band models)

https://drive.google.com/drive/folders/126ErYgKw7DwCI07WprAXWPD_uX6hUz-e?usp=sharing

- 13) From now on, you're forced to run separately newly added torch cell to fix PyTorch issues

- 14) Newer full band 498/HQ_2 model (with the same modified models.py for the full band models)

Copy model file to:

Gdrive\MDX_Colab\onnx\MDX-UVR Ins Model Full Band 498 (HQ_2)\” as in the link below, and replace models.py in Gdrive\MDX_Colab (if you didn't already for full band models)

https://drive.google.com/drive/folders/1O5b-uBbRTn_A9B2QkefkICT41YR9voMq?usp=sharing

15) For full band models, use only modified models.py attached above, or you'll get cutoff at 14.7kHz instead of 22kHz in spectrograms while using 427 models.py file.

16) For Kim other FT instrumental model with cutoff but the highest SDR (even than inst3)

Copy both (vocals and other) model files to:

Gdrive\MDX_Colab\onnx\Kim ft other instrumental model\” as in the link below, and replace models.py in Gdrive\MDX_Colab (if you didn't already for full band models)

<https://drive.google.com/drive/folders/1v2Hy4AgFOJ9KysebGuOgn0rlveu510j6?usp=sharing> (it will give only 1 stem output, models duplicated fixes errors in Colab, models.py is from inst3 model)

17) If you use models.py from fullband model, it will output fullband for ft other model, but giving much more vocal residues (but it still might be even better in some busy mix parts than VR models, while having still less vocal residues only in those busy parts like chorus) - definitely use min_mag here.

18) To fix the following error, make sure both vocals and invert vocals are always checked:

shell-init: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory

Intel MKL FATAL ERROR: Cannot load

/usr/local/lib/python3.9/dist-packages/torch/lib/libtorch_cpu.so.

Above error can also mean you need to terminate your session and start over. It randomly happens after using the Colab:

19) I've reverted old "Karokee" and "Karokee_AGGR" models to use with the oldest HV's [models.py](#) file, but these are old models (maybe they will do the trick, though).

20) *ModuleNotFoundError: No module named 'models'*

Sometimes switching models.py doesn't work correctly (especially during working on previously shared Colab folder with editing privileges) in that case, check Colab's file manager if models.py is actually present after you've made a change on GDrive. If not, rename it to models.py (it might have been renamed to something else).

21) Collection of all three models.py for all models for your comfort:

https://drive.google.com/drive/folders/1J35h9RYhPFk8dH-vShSW_AUharXY1YsN?usp=sharing

22) Main_406 vocal model

https://mega.nz/file/dcREzKTR#PYKK3s1NPicC3mBBYH8ejC2rK_Im3sAj0p9xcOi1cpE

```
"compensate": 1.075,  
"mdx_dim_f_set": 3072,  
"mdx_dim_t_set": 8,  
"mdx_n_fft_scale_set": 7680,
```

Models include here only: baseline, instrumental models: 415 (inst_1), 418 (inst_2), 464 (inst_3) trained on 17.7kHz, and vocal model 427, and Kim's vocal model (old) (instrumental should be automatically made by inversion option, but it's not a very good one for it) and 292 and 403 full band. If you want to use older 9.7 models, use old HV Colab above.

464/inst 3 should be the best in most cases for instrumentals and vocals than previous 9.x models, but depending, even in half of the cases, 418 can achieve better results, while full band 403 might give better results than inst3/464 in half of the cases.

Settings

max_mag is for vocals

min_mag for instrumentals

default

(deleted from the new HV Colab, still in Kae Colab above)

But "min mag solve some unwanted vocal soundings, but instrumental [is] more muffled and less detailed."

Also check out "default" setting (whatever is that, compare checksums if not one of these).

Chunks

As low as possible, or disabled.

Equivalent of min_mag in UVR is min_spec.

Be aware that UVR5, opposed to MDX Google Colab, applies cutoff to inverted output, matching the frequency of training frequency e.g. 17.7kHz for inst 1 and 3 models. It was to avoid some noise and vocal leftovers. You might have to apply it manually.

Also, you can uncomment visibility of compensation value in Colab, and change it to e.g. 1.08 to experiment.

Compensation value for 464 MDX-UVR inst. model is 1.0568175092136585

Default 1.03597672895 is for 9.7 model, and it also does the trick with at least Kim (old) model in GUI (where 1.08 had worse SDR).

Or check + 3.07 in DAW (it worked on Karokee model).

In Collab above, I also enabled visibility of max_mag for vocals and min_mag for instrumentals settings (mixing_algorithm).

Also, if you want to use Demucs option (ensemble) in Kae Colab, it uses stock Demucs 2, which in UVR5 was rewritten to use Demucs-UVR models with Demucs 3 or even currently better Demucs 4.

According to MVSEP SDR measurements, for ensemble Max Spec/Min Spec was better than Min Spec/Max Spec, but Avg/Avg was still better than these both.

Also for ensemble, Avg/Avg is better compared to e.g. Max Spec/Max Spec - it's 10.84 v 10.56 SDR in other result.

How denoiser work

It's not frequency based, it processes "the audio in 2 passes, one pass with inverted phase, then after processing the phase is restored on that pass, and both passes mixed together with gain * 0.5. So only the MDX noise is phase cancelling itself."

Or the other way round:

"it's only processing the input 2 times, one time normal and one time phase inverted, then phased restored after separation, so when both passes are mixed back together only the noise is attenuated. There's no other processing involved"

Denoise serves to fix so called MDX noise existing in all inst/voc MDX-NET (v2) models.

Web version (32 bit float WAV as output for instrumentals, just use MDX-B for single MDX-Net models).

It was 9.682 MDX-UVR model in 2021, but in the end of 2022 it's probably inst 1 judging by SDR (not sure, as results are not exactly the same), then more models were added (e.g. HQ_3):

<https://mvsep.com/>

Web version (paid for MDX, lossless):

<https://x-minus.pro/>

In kae Colab, you can keep the option Demucs: off (ONNX only), it may provide better results in some cases even with the old MDX narrowband models (non-HQ).

In Colab you can change chunks to 10 if your track is below 5:00 minutes. It will take a bit more time, but the quality will be a bit cleaner, but more vocal residues can kick in (esp. short sudden ones).

Be aware that MDX Colabs for single models have 16 bit output.

And also noise cancellation implementation for MDX models in kae and HV Colab can differ a bit, plus there is also separate denoise method available as separate model.

Code for denoise method in HV Colab [here](#).

As for any other settings, just use defaults since they're the best and updated.

Just for a vocal it's one of the best free solutions on the market, very close to the result of paid and (partly) closed Audioshake service (#1 AI in a Sony separation contest; SDRs are from the contest evaluation based on private dataset). Very effective, high quality instrumental isolation AI and custom model (but the old models are trained at 14.7 kHz [NET-X a.k.a. 9.x] in comparison to VR models, and 17.7kHz in newer models like inst X and kim inst).

In most cases MDX-UVR inverted models give less bleeding than VR (especially on bassy voice), while occasionally the result can be worse comparing to VR above, especially in terms of hi-end frequencies quality, but in general, MDX with UVR team models behaves the best for vocals and instrumentals.

Even instrumental from inverted vocals from vocal models gets less impaired than in VR, since vocal filtering is less aggressive, but with even more bleeding in some cases. Depends on a song.

You can support the creators of UVR and the newest MDX model is also available on <https://www.patreon.com/uvr> <https://boosty.to/uvr> to visit <https://x-minus.pro/> to get an online version of MDX there as well (with exclusive paid models).

At least paid x-minus subscription allows you to use MDX HQ_2 498 (or HQ_3 already) instrumental model and for VR arch - 2_HP-UVR (HP-4BAND-V2_arch-124m), and Demucs 6s on their website. Feel free to listen and download lots of uploaded instrumentals on x-minus already. Dozens of instrumentals available.

Outdated

Alternatively you can experiment with 9662 model and ensemble it with the latest UVR 5's 4 band V2 with -a min_mag as Anjok suggested (but it was when new models weren't released yet).

Remotely I only know about old Colab which ensembles any two audio files, but it uses old algorithm if I'm not mistaken, so it is not as good (better use the ensemble Colab linked at the very top of the document):

<https://colab.research.google.com/drive/1eK4h-13SmbjwYPecW2-PdMoEbJcpqzDt?usp=sharing>

Note

Don't disable invert_vocals in Colab even if you only need vocal instead of instrumental, otherwise the Colab will end up with error.

MDX noise

There is a noise using all MDX-UVR inst/vocal models, and it's model dependent (irc 4 stems don't have it). It's fixed in Colabs using denoiser "however by using my method, conversions will be 2x slower as it needs to predict twice.

I see no quality degradation at all, and I can't believe it actually worked rofl" -HV

Also, UVR 5 GUI has the same noise filtering implemented (if not better, also with alternative model).

Current MDX Colab has normalization feature "normalizes all input at first and then changes the wave peak back to original. This makes the separation process better, also less noise. IDK if you guys have tried this, but if you split a quiet track, and normalize it after MDX inference the noise sounds more audible than normalizing it and changing the peak back to original."

If you want to experiment with MDX sound, the Colab from before that change is below:
<https://colab.research.google.com/drive/1EXlh--o34-rzAFNEKn8dAkqYqBvhVDsH?usp=sharing> (might no longer work due to changes made by Google to Colab environment, the last maintained are kae and HV (new) Colabs)

Furthermore, you can also try manually mixing vocal with original track using phase inversion and add specific gain on vocal track (+1.03597672895 or +3.07) for 9.7 model (or other ones with different values), using both this and below Colab and save result as 32 bit float (but this might have more bleeding, but it uses 32 bit while chunking):
https://colab.research.google.com/drive/1R32s9M50tn_TRUGIkfnNPYdbUvQOcfh?usp=sharing#scrollTo=IkTLtOvyBuxc

(for e.g. the best compensation value for 464 MDX-UVR inst. model is 1.0568175092136585 and it's not constant)

Also be aware that MVSEP uses 32 bit for MDX-UVR models for ready inversion of any model too.

If you look for eliminating the noise from MDX-UVR instrumentals, also the method described in Zero Shot below might work.

"I just run the MDX vocals thru UVR to remove any remaining buzz noises and synths, it works great so far" (probably meant one of VR models)

Average track in Colab is being processed in 1:00-1:30 minute using slower Tesla K80 (much faster than even UVR's HP-4BAND-V2_arch-124m model).

If you want to get rid of some artifacts, you can further process output vocal track from MDX through Demucs 3.

Options in the old HV MDX Colab/or kae fork Colab (from the very top)

Demucs model in the older MDX-Net Colab

When it's enabled, it sounds better to me, used with the old narrowband 9.X and newer vocal models, as Demucs 2 model is fullband, but opinions on superiority of this option are divided, and MVSEP dev made some SDR calculation where it achieved worse results with Demucs enabled. But be aware, that inverted results from narrowband are still fullband despite the narrowband training frequency, as there's no cutoff matching present in Colab, as it's implemented in UVR GUI as a separate option. Using such cutoff matching training frequency (which can be observed in non-inverted stem) might lead to less noise and residues in the results. Demucs model will work correctly only with vocal models in Colabs (we didn't have any MDX instrumental models back then, so naming scheme is reversed for these models, hence Demucs model with instrumental model produces distorted sound, it mixes vocals with instrumental in a weird way).

"The --shifts=SHIFTS performs multiple predictions with random shifts (a.k.a. the shift trick) of the input and average them. This makes prediction SHIFTS times slower but improves the accuracy of Demucs by 0.2 points of SDR. It has limited impact on Conv-Tasnet as the model is by nature almost time equivariant. The value of 10 was used on the original paper, although 5 yields mostly the same gain. It is deactivated by default, but it does make vocals a bit smoother.

The --overlap option controls the amount of overlap between prediction windows (for Demucs one window is 10 seconds). Default is 0.25 (i.e. 25%) which is probably fine." You can even try out 0.1, but for Demucs 4 it decreases SDR in ensemble if you're trying to separate a track containing vocals. If it's instrumental, then 0.1 is the best (e.g. for drums).

(outdated/for offline use/added to Colab)

Here's the new MDX-B Karokee model!

<https://mega.nz/file/iZgiURwL#jDKiAkGyG1Ru6sn21MklwF90C-fGD0o-Ws58Mn3O7y8>

The archive contains two versions: normal and aggressive. The second removes the lead vocals more. The model was trained using a dataset that I completely created from scratch. There are 610 songs in total. We ask that you please credit us if you decide to use these models in your projects (Anjok, aufr33).

Demucs 3

for 4 stems

(SDR 7.7 for 4 stems, it's better than Spleeter (which is SDR 6.5-7), or better than MDX 4 stem. In most cases, it's even better than Audioshake - at least on tracks without leading guitar)

Accompanied by MDX-UVR 9.7 vocal model, it gives very good 4 stem separation results
(For Demucs 4 a.k.a "htdemucs" check below)

https://colab.research.google.com/drive/1yyEe0m8t5b3i9FQkCI_iy6c9maF2brGx?usp=sharing (by txmutt), alternatively with [float32](#) here

Or <https://huggingface.co/spaces/akhaliq/demucs>

Or <https://mvsep.com/>

Pick up from the list Demucs Model B there.

You can export result files in MP3 320kbps, WAV and FLAC. File limit is 100MB and has a 10 minute audio length limit.

To use Demucs 3 locally:

<https://discord.com/channels/708579735583588363/777727772008251433/909145349426384917>

Currently, all the code uses now main branch which is Demucs 4 (previously HT) but these Colabs use old mdx_extra model.

Demucs 3 UVR models 2 stem only available on MVSEP.com or in UVR5 GUI (nice results in cases when you suffer vocal bleeding i regular UVR5, GSEP, MDX 9.7 - model 1 less aggressive, model 2 more destructive, model bag has more bleeding of all three).

In Colab, judging by quality of drums track, I prefer using overlap 0.1 (only for instrumentals), but default set by the author is 0.25 and is better for sound of instrumental as a whole. But it still provides decent results with instrumentals.

Also, HV had overall better separation quality results using shifts=10, but it increases separation time (it's also reflected by MVSEP's SDR calculations). Later we found out it can be further increased to 20.

Also, I have a report that you may get better results in Demucs using previously separated instrumental from e.g. UVR.

Anjok's tip for better instrumentals: "I recommend removing the drums with the Demucs, then removing the vocals and then mixing the drums back in". Yields much better results than simple ensemble.

It works the best in cases when drums get muffled after isolation, e.g. in hip-hop. You need to ensure that tracks are aligned correctly. E.g. if you isolate drumless UVR track, isolate also regular track to align drumless UVR track easier with drums track from Demucs, otherwise there will be hard to find the same peaks. Then simply align drumless UVR the same as regular track is aligned and mute/delete UVR regular (instrumental) track.

Be aware! This is not a universal solution for the best isolation in every case. E.g. in tracks with busy mix like Eminem - Almost Famous, the guitar in the background can get impaired, and so even drums (UVR tends to impair guitars in general, but on drumless track it was even more prevalent - in that case normal UVR separation did better job).

Also, if you slow down the input file, it may allow you to separate more elements in the "other" stem.

It works either when you need an improvement in such instruments like snaps, human claps, etc.

Normally, the instrumental sounds choppy when you revert it to normal speed. The trick is - "do it in Audacity by changing sample rate of a track, and track only (track menu > rate), it won't resample, so there won't be any loss of quality, just remember to calculate your numbers

44100 > 33075 > 58800

48000 > 36000 > 64000

(both would result in x 0.75 speed)

etc.".

Also, there's dithering enabled in Audacity by default. Might be worth disabling it in some cases. Maybe not, but still, worth trying out. There should be less noise.

BTW. If you have some remains of drums in acapella using UVR or MDX, simply use Demucs, and invert drums track.

"The output will be a wave file encoded as int16. You can save as float32 wav files with --float32, or 24 bits integer wav with --int24" it doesn't seem to work in Colab.

Demucs 4 (+ Colab) (4, 6 stem)

4 stem, SDR 9 for vocals on MUSDB HQ test, and SDR 9 for mixdowned instrumentals (5, 6 stem - experimental piano [bad] and guitar)

<https://github.com/facebookresearch/demucs> (all these models available in UVR 5 GUI or MVSEP [just x-minus doesn't have ft model for at least free users, it was mmi model at some point, but then got replace by MDX-B which "turned out to be not only higher quality, but also faster"])

Google Colab (all 4-6 stem models available, 16-32 bit output)

https://colab.research.google.com/drive/117SWWC0k9N2MBj7biagHjkRZpmd_ozu1

or Colab with upload script without Google Drive necessity:

https://colab.research.google.com/drive/1dC9nVxk3V_VPjUADsnFu8EiT-xnU1tGH?usp=sharing

or Colab by Bezio with batch processing, (only mp3 output and no overlap/shifts parameters beside model choice - choose demucs_ft for 4 stems):

https://colab.research.google.com/drive/15lscSKj8u6OrooR-B5GHxlvKE5YXyG_5?usp=sharing

or Colab with batch processing by jarredou (less friendly GUI, but should be usable too, lossless):

https://colab.research.google.com/drive/1KTkiBI21-07JTYcTdhlj_muSh_p7dP1d?usp=sharing

"I'd recommend using the "htdemucs_ft" model over normal "htdemucs" since IMHO it's a bit better", also SDR measurements confirm that. 6s might have more vocal residues than both, but will be a good choice in some cases (possibly songs with guitar).

All the best stock models:

- htdemucs_ft (f7e0c4bc-ba3fe64a.th, d12395a8-e57c48e6.th, 92fcf3b6-ef3bcb9c.th, 04573f0d-f3cf25b2.th [drums, bass, other, vocals])

“fine-tuned version of htdemucs, separation will take 4 times more time but might be a bit better. Same training set as htdemucs”.

Can be obtained with UVR5 in download center (04573f0d-f3cf25b2.th, 04573f0d-f3cf25b2.th, d12395a8-e57c48e6.th, f7e0c4bc-ba3fe64a.th; not in order)

- htdemucs - “first version of Hybrid Transformer Demucs. Trained on MusDB + 800 songs.” Default Demucs model in e.g. UVR5 (955717e8-8726e21a.th)

- htdemucs_mmi = Hybrid Demucs v3, retrained on MusDB + 800 songs
htdemucs_6s = 6 sources version of htdemucs, with piano and guitar being added as sources. Note that the piano source is not working great at the moment.”
“nowhere near Logic Pro” from May 2025 update.

- mdx_extra: The best Demucs 3 model from MDX 2021 challenge. Trained with extra training data (including MusDB test set), ranked 2nd on the track B of the MDX 2021 challenge.

- mdx_extra_q: a bit worse quantized version of the above (a bit faster)

Be aware that also UVR team and also ZFTurbo [available on MVSEP and GitHub] trained their own Demucs models (respectively instrumental and vocal ones), but there are some issues with ZFTurbo model using inference other than provided on his GitHub (so it's so far not compatible with e.g. UVR giving “KeyError: "models" for ckpt Demucs models instead of th).

To use the best Demucs 4 model in the official Colab (the 2nd link) rename model to e.g. “htdemucs_ft”. It can behave better than 6 stems if you don't need extra stems.

In other cases, extra stems will sound better in the mix, although using 6s model, vocal residues are usually louder than in ft model (but that might depend on a song or genre).

Despite the fact that 6s is an electric guitar model, it can also pick up acoustic guitar very well in some songs.

The problem with 6s models is that “when a song has a piano because not only the piano model is not the best, but it also makes the sound itself worse rather than just very filtered piano, it sounds like distorted filtered piano”

Sometime Gsep can be “still better because each stem has its dedicated model” but it depends on a song (other stem in GSep can be better more frequently, but now MDX23 jarredou fork or Ensemble models on MVSEP returns good other stems as well)
Gsep instead of inverting the whole result among stems like Demucs, won't preserve all the instruments occasionally.

"htdemucs (demucs 4) comes a bit closer [vs Gsep], most of the time the bass is better and there are few instances where demucs picks up drums better"

"From my experience and testing: If you decide to process an isolated track through Demucs, it has no trouble identifying what is bass guitar and what isn't bass guitar [does not matter if it's finger/pick/slap, it works on all of them for me, except distorted wah-wah bass]. The leftover noise [the part's that demucs did not pick up, and left it in the (No Bass) stem] is usually lower than minus 40 - 45 DB, and it's either noise, or hisses usually."

The problem comes when there are instruments besides the bass guitar that are playing beside it [a.k.a. music], since these are separation models, not identification models. It starts having trouble grabbing all the upper harmonics [which is the multiple of the root note frequency], and the transients, potentially starts mis-detecting, or in extreme cases, it does not pick up the bass at all."

"When used with "--shifts" > 0, demucs gives slightly different results each time you use it, that can also explain some little score differences"

<https://github.com/facebookresearch/demucs/issues/381#issuecomment-1262848601>

Initially, Shifts 10 was considered as max, but it turned out 20 can be used.
Overlap 0.75 is max before it gets very slow (and 0.95 when it becomes overkill).

While we also thought overlap 0.99 is max, it turned out you can use 0.99999 in UVR, and 0.999999 in CLI mode, but both make separations tremendously long, even 0.999 much longer than 0.99.

On GTX 1080 Ti on 1 minute song:

`0.99` = Time Elapsed: `00:09:45`
`0.999` = Time Elapsed: `01:36:45`

Also, shifts can be set to 0.

With htdemucs_ft, shifts doesn't matter nearly as much as overlap, I recommend keeping (shifts) at 2 [for weaker GPUs].

The drum SDR with 1 and 10 shifts difference is about 0.005

So overlap impacts SDR a bit more than shifts.

"The best way to judge optimum settings is to take a 10-second sample of a vocal extraction where there's evident bleeding and just keep trying higher overlaps etc until you're happy, or you lose patience, then you'll arrive at what I call the 'Patience Ratio'. For me, it's 2x song length."

Installation of only Demucs for Windows

Use UVR, or:

Download the git repo, extract it, then open PowerShell and write
"pip install *insert the directory of the extracted repo here*"

<https://github.com/facebookresearch/demucs#egg=demucs>

Alternatively, execute this command:

pip install git+<https://github.com/facebookresearch/demucs#egg=demucs>

or download the git repo first and then

"pip install *insert the directory of the extracted repo here*"

In case of "norm_first_error run this line or update torch to 1.13.1
python.exe pip install -U torch torchaudio

In Colab, judging by quality of drums track, I prefer using overlap 0.1 (better only for instrumentals) with shifts 10 (actually can be set to even 20), but default set by the author is 0.25 and is better for sound of instrumental as a whole.

Also, we have overall better separation quality results using shifts=10, but it increases separation time (it's also reflected by MVSEP's SDR calculations). Overlaps also increase general separation quality for instrumentals/vocals, at least up to 0.75, but everything above starts being tremendously slow (few hours for 0.99 max setting).

If you use particularly high overlap like 0.96 for a full length song, you can run out of Colab time limit if it's not your first file being processed during this session (for cases when processing takes more than 1 hour). If you exceed the limit, you can change Google account in the right top (don't use other account during mounting, or you'll end up with error). The limit is reset after 12 hours (maybe sooner). It's capable of processing one file for two hours, at least only if it's the first file being processed for a longer time during this day. Also, rarely, it can happen that your file is being processed faster than usual despite the same T4 GPU. If you have "*something has gone terribly wrong*" error right on the separation start, simply retry. If in the end of long separation - ignore it, and don't retry - your result is in the folder.

- **clipclamp** - uncheck it to disable hard limiter, but it may cause separation artifacts on some loud input files or will change volume proportions of the stems. I like it enabled somehow.

- Q: How to stop Demucs from rescaling the volume of stems after they're extracted (without adjusting the volume of the input mixture and passing --clip-mode=clamp)?

A: Set "--clip-mode none" argument coupled with export to --float32" (jarredou)

Picture

Demucs parameters explained by jarredou

- "Overlap is the percentage of the audio chunk that will be overlapped by the next audio chunk. So it's basically merging and averaging different audio chunk that have different start (& end) points.

For example, if audio chunk is '|---|' with overlap=0.5, each audio chunk will be half overlapped by next audio chunk:

-shifts is a random value between 0 and 0.5 seconds that will be used to pad the full audio track, changing its start(&end) point. When all "shifts" are processed, they are merged and average. (...)

It's to pad the full song with a silent of a random length between 0 and 0.5 sec. Each shift add a pass with a different random length of silence added before the song. When all shifts are done (and silences removed), the results are merged and averaged.

Shifts is performing lower than overlap because it is limited to that 0.5 seconds max value of shifting, when overlap is shifting progressively across the whole song. Both works because they are shifting the starting point of the separations. (Don't ask me why that works!)

But overlap with high values is kinda biased towards the end of the audio, it's caricatural here but first (chunk - overlap) will be 1 pass, 2nd (chunk - overlap) will be 2 passes, 3rd (chunk - overlap) will be 3 passes, etc..."

So Overlap has more impact on the results than shift.

"Side-note: Demucs overlap and MVSEP-MDX23 by ZFTurbo overlap features are not working in the same way. (...)

Demucs is kinda crossfading the chunks in their overlapping regions, while MVSep-MDX23 is doing avg/avg to mix them together"

Why is overlapping advantageous?

Because changing the starting point of the separation give slightly different results (I can't explain why!). The more you move the starting point, the more different the results are. That's why overlap performs better than shifts limited to 0-0.5sec range, like I said before. Overlap in Demucs (and now UVR) is also crossfading overlapping chunks, that is probably also reducing the artifacts at audio chunks/segments boundaries.

[So technically, if you could load the entire track in at once, you wouldn't need overlap]

Shifts=10 vs 2 gives +0.2 SDR with overlap=0.25 (the setting they've used in their original paper), if you use higher value for overlap, the gain will be lower, as they both rely on the same "trick" to work.

Shifts=X can give little extra SDR as it's doing multiple passes, but will not degrade "baseline" quality (even with shifts=0)

Lower than recommended values for segment will degrade "baseline" quality.

So in theory, you can equally set shifts to 0 and max out overlap.

Segments optimum (in UVR beta/new) is 256.

Gsep (2, 4, 5, 6 stem, karaoke)

<https://studio.gaudiolab.io/>

Paid (20 minutes free in mp3 - no credit card required)

7\$/60 minutes

16\$/240 minutes

50\$/1200 minutes

Electric guitar (occasionally bad), good piano, output: mp3 320kbps (20kHz cutoff), wav only for paid users, accepted input: wav 16-32, flac 16, mp3, m4a, mp4, don't upload files over 100MB (and also 11 minutes may fail on some devices with Chrome "aw snap" error), capable of isolating crowd in some cases, and sound effects. Ideally, upload 44kHz files with min. 320kbps bitrate to have always maximum mp3 320kbps output for free.

2025 metrics for 2 stems

https://mvsep.com/quality_checker/entry/9095

(outdated) About its SDR

[10.02 SDR](#) for vocal model (vs Byte Dance 8.079) on seemingly MDX21 chart, but non-SDR rated newer model(s) were available from 09.06.22, and later by the end of July, and now new model is released since 6 September (there were 4 or 5 different vocal/instrumental

models in total so far, the last introduced somewhere in September and no models update was performed with later UI update). [MVSEP SDR comparison](#) chart on their dataset, shows it's currently around SDR 9 for both instrumental and vocals, but I think evaluation done on [demixing challenge](#) (first model) was more precise. Be aware that GSEP causes issue of cancelling different sounds which cannot be found in any stem.

Since May 2024 update there was an average of 0.13 SDR increase for mp3 output and first 19 songs from multisong dataset evaluation, but judging by no audible difference for most people, they could simply change some parameters of inference. Actually, it's more muddy now, but in some songs there are a bit less of vocal residues, and in other songs, noticeably more. Inverting the mixture with vocals in WAV will muffle the sound in overall, e.g. snares, esp. in places of these residues, but the residues will disappear as well.

Uncheck vocals to download WAV file if WAV download doesn't work,
and uncheck instrumental to download vocals in WAV -
don't check all stems if you can't download WAV at all and download window simply
disappears.

If you still can't download your WAV files, go to Chrome DevTools>Network before starting downloading, and press CTR+R, now start download. Now both stems should be shown in DevTools>Network, starting with input file name, e.g. instrumental with ending name "result_accom.wav" (usually marked as "fail" in State column and xhr as type), click the entry with right mouse button and choose Open in new tab.

The download may fail frequently, forcing you to resume the download multiple times in browser manually, or wait a bit on the attempt to download the file at the start.

Free option of separating has been removed since the May 2024 update. There's only a 20-minute free trial with mp3 output.

Vocals and all other stems (including instrumentals/others) are paid, and length for each stem is taken from your account separately for each model.

No credit is not required for the trial.

For free, only mp3 output and 10 minutes input limit.

For paid users there's a 20 minutes limit, and mp3/wav output, plus paid users have faster queue, shareable links, and long term results storage.

Seems like there weren't any changes in the model
<https://www.youtube.com/watch?v=OGWaoBOKiMg>

The old files from previous separations on your account didn't get deleted so far if you have premium.

<https://studio.gaudiolab.io/pricing>

There was also added a new option for vocals called “Vocal remover” - good “for conservative vocals, it’s fine it even has 15 best scoring on SDR.” and 10.85 in vocals on multisong dataset.

Instruction

Log in, and re-enter into the link above if you feel lost on the landing page.

For instrumental with vocals, simply uncheck drums, choose vocal, and two stems will be available for download.

As for using 4/5 stem option for instrumental after mixing if you save the tracks mixed in 24 bit in DAW like Audacity, it currently produces less voice leftovers, but the instrumental have worse quality and spectrum probably due to noise cancellation (which is a possible cause of [missing sounds](#) in other stem). Use 5 stem, but cut silence in places when there is no guitar in the stem to get comparable quality to 4 stem in such places.

For 3-6 stem, you better don’t use dedicated stems mixing option - yes, it respects muting stems to get instrumental as well, but the output is always mp3 128kbps while you can perform mixdown from mp3s to even lossless 64 bit in free DAWs like Audacity or Cakewalk.

In some very specific cases you can get a bit better results for some songs by converting your input FLAC/WAV 16 to WAV 32 in e.g. Foobar2000.

Troubleshooting

- (fixed for me) Sometimes very long “**Waiting**” or recently “**Waiting**” - can disappear after refreshing the site after some time (July 2023) - e.g. if you see “SSG complete” message, you can refresh the site to change from waiting to waveform view immediately. I had that on a fresh account once when uploading the very first file on that account, and then it stopped happening (later it happened for me on an old account as well).

- (might be fixed too) **If you don’t see all stems after separation** (e.g. while choosing 2 stems, only vocals or only instrumental is shown) and only one stem can be downloaded (can’t be done on mobile browser) - workaround:

- “Aw snap” error on mobile Chrome can happen on regular FLACs as well as an attempt to download a song. Simply go back to the main page and try to load the song again and download it.

- If nothing happens when you press download button on PC, also go to Chrome DevTools>Network>All and click download again. Then new files will appear on the list. Right

click and open mp3 file in a new tab to begin download. Alternatively, log into your account in incognito mode.

- If you have "An error has occurred. Please reload the page and try again." try deleting Chrome on mobile (cleaning cache wasn't enough in one case).
- (rather fixed) If you have "**no audio**" error all the time when separation is done, or preview loading is infinite, or you have only one stem, also -
In PC Chrome go to DevTools>Network>[All](#) and refresh this audio preview site, and new entries will show up on the right, which among others will list filenames with your input file name with stems names e.g. "rest of targets" in the end.
Double click it or click RBM on it and press open on new tab, and download will start.
If no filenames to download appear on the list, press CTRL+R to refresh the site, and now they should appear.
In specific cases, files in the list won't show up, and you will be forced to log in to GSEP using incognito mode (the same account and result can be used). Also, make sure you have enough of disk space on C:.
Alternatively, clean site/browser cache (but the latter didn't help me at some point in the past, don't know how now).
If still the same, use VPN and/or new account (all three at the same time only in very specific cases when everything fails). You can also use different browser.
- When you see loop of redirections when you just logged, and you see Sign In (?~go to main page) simply enter the main link <https://studio.gaudiolab.io/gsep>
- If you're getting mp3 with **bitrate lower than 320kbps** which is base maximum quality in this service (but you get 112/128/224 output mp3 instead)
> Probably your input file is lossy 48kHz or/and in lower bitrate than 320kbps > your file must be at least mp3 320kbps 44kHz (and not 48kHz). The same issue exists for URL option and for Opus file downloaded from YouTube when you rename it to m4a to process it in GSEP.
To sum up - GSEP will always match bitrate of the input file to the output file if it's lower than 320kbps. To avoid this, use lossless 44kHz file or if you can't, convert your lossy file to WAV 32 bit (resample Opus to 44kHz as well - it's always 48kHz, for YT files, don't download AAC/m4a files - they have cutoff at 16kHz while Opus at 20kHz). Now you should get 320kbps mp3 as usual without any worse cutoff than 20kHz for mp3 320kbps.
If you still not get 320kbps, try using incognito mode/VPN/new account (at best all three at the same time).
You can use Foobar2000 for resampling e.g. Opus file (RBM on file in playlist>convert>processing>resampler>44100. And in output file format>WAV>32 bit). Don't download from YT in any other audio than Opus, otherwise it will have 16kHz cutoff and separation result will be worse.

- (fixed) Also on mobile, the file may not appear on your list after upload, and you need to refresh the site.
 - If FLAC persists to be stuck in the "Uploading" screen, try converting it to WAV (32-bit float at best)
 - Check [this](#) video for fixing issues in missing sounds in stems (known issue with GSEP)
 - GSEP separation results don't begin at the same time signature like UVR results.
- > In order to fix it, convert mp3 to WAV or align stems manually if you need it for some comparisons or manual ensemble. Also some DAWs can correct it automatically on import.

Eventually hit their [Discord](#) server and report any issues (but they're pretty much inactive lately).

Remarks about quality of separation

"The main difference (vs old model) is the vocals. I can't say for sure if they're better than before, but there is a difference, the "others" and "bass" are also different. Only the drums remain the same. Generally better, but the difference is not massive, depends on the song" (begruijly)

GSEP is generally good for tracks where using all the previous methods you had bleeding (e.g. low-pitched hip-hop vocals) or got flute sounds removed, although it struggles with "cuts" and heavily processed vocals in e.g. choruses. Though, it has more bleeding in some cases when the very first model didn't, so new MDX-UVR models can achieve generally better results now.

"GSEP is good at piano extraction, but it still lacks in vocal separation, in many times the instruments come out together with the voices, this is annoying sometimes."

Electric guitar model got worse in the last update in some cases. Also, bass & drums also not so loud since the first release of gsep.

"Electric guitar model barely picks up guitars, it doesn't compare to Demix/lalal/Audioshake".
"I kinda like it. When it works (that's maybe 50-60% of time), it's got merit."

The issue happens (also?) when you process (GSEP) instrumental via 5 stems. If you process a regular song with vocals - it picks up guitar correctly. It happens only in a place where previously was vocal removed by GSEP 2 stem.

I only tested GSEP instrumental so far, I don't know whether it happens on official instrumentals too (maybe not).

The cool thing is that when the guitar model works (and it grabs the electric), the remaining 'other' stem often is a great way to hear acoustic guitar layers that are otherwise hidden.

The biggest thing I'd like to see work done on is the bass training. At present, it can't detect the higher notes played up high... whereas Demucs3/B can do it extremely well."

It has "much superior" other stem than Demucs or even better than Audioshake. It has changed since 6 September 2022, but probably got updated since then and is probably fine.

As for 14.10.22 piano model sounds "very impressive".

As for the first version of the model comparable vocal stem to MDX-UVR 9.7, but with current limitation to mp3 320kbps and worse drums and bass than Demucs (not in all cases). Usually less bleeding in instrumentals than VR architecture models.

"Gsep sounds like a mix between Demucs 3 and Spleeter/lalal, because the drums are kind of muffled, but it's so confident when removing vocals, there aren't as many noticeable dips like other filtered instrumentals, and it picks up drums more robustly than Demucs. [it can be better in isolating hihats than Demucs 4 ft model too]

It removes vocals more steadily and takes away some song's atmospheres, rather than UVR approach which tries to preserve the atmosphere, but [in UVR] you end up with vocal artefacts"

As for tracks with more complicated drums sections: "GSEP sounds much fuller, Demucs 3 still has this "issue" with not preserving complex drums' dynamics" it refers to e.g. not cancelling some hi-hats even in instrumentals.

It happens that some instruments can be deleted from all stems. "From what I've heard, [it] gets the results by separating each stem individually (rather than subtractive / inverting etc.), but this means some sounds get lost in between the cracks you can get those bits by inverting the gsep stems and lining up with the original source, you should then be left with all the stuff gsep didn't catch".

Also, I'd experiment with the result achieved with Demucs ft model, and apply inversion for just the specific stem you have your sounds missing.

As for June 2023 gsep is still the best in most cases for stems, not anywhere close to being dead

gsep loves to show off with loud synths and orchestra elements, every other mdx/demucs model fail with those types of things

Processing

After your track is uploaded (when 5 moving bars disappear) it's very fast, and it takes 3-4 minutes for one track to be separated using 2 stem option (processing takes around 20

seconds). If 5 bars are moving longer than expected track upload time, and you see that nothing uses your internet upload, simply press CTRL+R and retry, if still the same, log off and log in again. It can rarely happen that the upload stuck (e.g. when you minimize the browser on mobile or switch tabs).

Generally it's very fast and long after the very first GSEP days, I needed to wait briefly in queue twice at 6-9 PM CEST, and I think once on Sunday in weekend of adding new model once in my whole life I waited around 7 minutes. Usually you wait in a queue longer than processing takes, so it's bloody fast.

(*outdated*)

If your stems can't be downloaded after you click the download button, go to Tools for Developers in your browser and open the console and retry. Now you should see an error with file address and your file name in it. You can simply copy the address to the address bar and start downloading it.

(Outdated - 3rd model changes) The quality of hi-hats is enhanced, sometimes at the cost of less vivid snare in less busy mix, while it's usually better in busy mix now, but it sometimes confuses snare in tracks when it sounds similar to hi hat making it worse than it was. So a trap with lots of repetitive hi-hats and also tracks with a busy mix should sound better now.

dango.ai

([2](#) or [more](#) [up to 6+] stems, paid only, 30 seconds free preview of mp3 320 output, 20kHz cutoff)

drums, vocal, bass guitar, electric guitar, acoustic guitar, violin, erhu

"10 tracks = €6.33 + needs Alipay or WeChat Pay"

max 12 minutes input files allowed

Now the site has English interface

Currently, one of the best instrumental results (if not the best). Not so good vocals.

(for older models) The combination of 3 different aggression settings (mostly the most aggressive in busy mix parts) gives the best results for Childish Gambino - Algorithm vs our top ensemble settings so far. But it's still far from ideal (and [not only] the most aggressive one makes instruments very muffled [but vocals are better cancelled too], although our separation makes it even a bit worse in more busy mix fragment).

As for drums - better than GSEP, worse than Demucs 4 ft 32, although a bit better hihat. Not too easy track and already shows some differences between just GSEP and Demucs when the latter has more muffled hi-hats, but better snare, and it rather happens a lot of times

(old) Samples:

Instrumental Drums

Also, it automatically picks the first fragment for preview when vocal appears, so it is difficult to write something like AS Tool for that (probably manipulations by manual mixing of fake vocals would be needed). Actually, smudge wrote one.

Very promising results even for earlier version.

They wrote once somewhere about limited previews for stem mode (for more than 2 mode) and free credits, but haven't encountered it yet.

They're accused by aufr33 to use some of UVR models for 2 stems in the past, without crediting the source (and taking money for that).

Now new, better models are released. Better instrumentals than in UVR/MVSep, and not the same models.

It used to be possible to get free 30 seconds samples on dango.ai, but recently 5 samples are available for free (?also) here:

<https://tuanziai.com/vocal-remover/upload>

You must use the built-in site translate option in e.g. Google Chrome, because it's Chinese only. You are able to pay for it using Alipay outside China.

music.ai

Paid - \$25 per month or pay as you go ([pricing chart](#)). In fact, no free trial.
Good [selection](#) of models and interesting [module stacking](#) feature.

To upload files instead of using URLs "you make the workflow, and you start a job from the main page using that custom workflow" [~ D I O ~].

Allegedly it's made by Moises team, but the results seem to be better than those on Moises.

"Bass was a fair bit better than Demucs HT, Drums about the same. Guitars were very good though. Vocal was almost the same as my cleaned up work. (...) An engineer I've worked with demixed to almost the same results, it took me a few hours and achieve it 39 seconds" (...) I'd say a little clearer than MVSEP 4 Ensemble. It seems to get the instrument bleed out quite well,"

"Beware, I've experienced some very weird phase issues with music.ai. I use it for bass, but vocals are too filtered / denoised imo and you can't choose to not filter it all so heavily."

Sam Hocking

MDX23 by ZFTurbo (jarredou fork) - 2, 4 stems

(2-4 stems, max 32-bit float output)

As of October 2025, the following Colabs are defunct due to Google's runtime changes (possible [fix](#)).

[v2.5](#), [v2.5 /w HQ_5](#) (experimental - muddiness, residues - set HQ_5 weight to 2.5 or lower),
[/w SCNet XL](#) (weights not measured), [2.4](#) (added BS-Roformer model), [2.3](#) (Kubinka fork of jarredou's Colab /w FLAC conversion, ZIP unpacking, new fullband preservation), [2.1](#) (voc_ft instead of Kim Vocal 2, a bit better SDR over 2.0 in overall), [2.2](#) (with MDX23C model, may have more vocal residues vs 2.1), org. [2.3](#) (with VitLarge model instead instr-HQ3), [GUI/CML](#) (GUI only for older original 1.x release by ZFTurbo), instructions for local installation at the button

The ZFTurbo 1.0 Colab further modified by jarredou to alleviate vocal residues. It adds better models and volume compensation, fullband trick for narrowband vocal models, higher frequency bleeding fix and much more. Currently, it achieves not much worse SDR as current "Ensemble 4 models" on MVSEP utilizing some newer private models available only on MVSEP already. Initially released 1.x code by ZFTurbo received 3rd place in the latest MDX 2023 challenge.

"I have successfully processed a ~30min track with vocals_instru_only mode [on Colab] while I was working on that 2.3 version, but it was probably with minimal settings. [Errors/freezes are] already happening during Demucs separation when you do 4-stem separation with files longer than ~10-15 min" jarredou
With v. 2.4, 30 minute file was too long, and the Colab hung on Roformer model separation.

The Colab combines results of then the best public models of different architectures using custom weights for every model (like a manually set volume for every stem, then mixed with others together), instead of usual methods of ensembling as in UVR, which in e.g. "avg" averages results of all models (so there the same volume is used for every stem). More tricks in the Colab explained further below.

As of v. 2.5 "Baseline ensemble is made with Kim Melband Rofo, InstVocHQ and selected 1296 or 1297 BS Rofo" (so Kim Rofo was added, and VitLarge is no longer default).

~"Free Google Colab gives you 3h per day, then you need to wait 24h, and next day it gives you 2 free hours, after 24h wait you'll get only 1h, and 24h later, 2h of free credits, the day after 1h of free credits, etc... and once in that pattern, you have to wait 48h to recover the 3h back." You can just change Google account when GPU limit is reached, but remember to use the same new account during mounting GDrive, otherwise you may get an error.

"I've opened a donation account for those who would want to support me:
<https://ko-fi.com/jarredou>"

Troubleshooting

- "PytorchStreamReader error"
simply restart the environment, it's a random issue occurring in the Colab.
- "usage: inference.py [-h] --input_audio INPUT_AUDIO [INPUT_AUDIO ...] --output_folder"
(and the whole list of arguments is shown below)
launch mount to GDrive cell (it's not being done automatically) or change file input and output path
- "ValueError: Mountpoint must not already contain files"
(on attempt of mounting GDrive), go to file manager on the left, and you probably have GDrive folder with empty folders you need to delete from there first, and retry (might happen when you use GDrive on this account while it's not mounted yet, but Colab works).
- "no such file"
(error in v2.3 while batch processing)
"it's square brackets []"
when it sees a [in the filename, it then thinks there's two additional [] in the name
changing to regular parentheses does work"
- "If I input more than a single song it just starts building up on model data without clearing the old one, so it slowly starts running out of VRAM and then gets stuck"

Experimenting with settings

- Default settings of the Colab are a good starting point in general
- Some people like to increase BigShifts to 20 or even 30 with all other default settings (some songs might be less muddy that way), but default 3 is already balanced value, but exceeding 5 or 7 may not give a noticeable difference, while increasing separation time severely.
- Switching from 1296 to 1297 model produces more muddy/worse instrumentals in this Colab (more sudden jumps of dynamics from residues). Similar situation with decreasing BigShifts to 1.
- voc_ft enabled might give less muddy results, but with more residues in instrumentals

- In 2.5 you can try out the following settings by mesk:
 "Set the weights of BS-RoFormer & MDX23C to 0, enable VitLarge, and set the weights of Mel-RoFormer & VitLarge to 8 & 3 respectively.
 You can set BigShifts to whatever you'd like, I think 5 or 7 is optimal" but mesk uses even 9.
 VitLarge overlap can no longer be changed in v2.5 of the Colab, probably only in CLI version.
- Or you can test ensemble of only Kim weight 10 + Vit weight 5, BigShifts e.g. 9
- Or BS-Roformer with MDX23C
 - Experimentally set "Separation_mode:" to 4 stems (slower) and "filename_instrument2" will be the sum of the Drums + Bass + Other stems that are obtained by processing "instrument" with multiple Demucs models. It might have a bit less vocal residues or be a bit muddier. Vs 2.1 denoiser is less aggressive as its disabled for some stems to save on VRAM.
 - Increasing overlap might give muddier results, but potentially better if you hear some vocal residues
 - In e.g. older v. 2.4 you might want to disable VitLarge to experiment (it's disabled in 2.5) - the model increases some noise at times
 - Older versions than 2.4 have very clean results for instrumentals, although it can rarely fail in getting rid of some vocals in quiet fragments of a track, but it has bigger SDR than the best ensembles in UVR. Versions 2.4 and newer started to utilize BS-Roformer arch, which is pretty muddy itself, but deprived of the majority of vocal residues.
 - For instrumentals, I'd rather stick to instrument2 results (so sum of all 3 stems instead of inversion with e.g. inst only enabled) but some fragments can sound better in instrument and it also slightly better SDR, so e.g. instrument can give louder snares at times, while instrument2 is muddier but sometimes less noisy/harsh. It can all depend on a song. Most people can't tell a difference between both.
 - If you suffer from some vocal residues in v. 2.2.2, try out these settings
 BigShifts_MDX: 0
 overlap_MDX: 0.65
 overlap_MDXv3: 10
 overlap_demucs: 0.96
 output_format: float
 vocals_instru_only: disabled (it will additionally give instrument2 output file for less vocal residues in some cases)
 - You can manipulate with weights.

E.g. different weight balance, in 2.2 with less MDXv3 and more VOC-FT.

- For vocals in [2.2](#) you can test out [these](#) (dead link) settings (21, 0, 20, 6, 5, 2, 0.8)
- In older versions of the Colab Overlap large and small control overlap of song during processing. The larger value, the slower processing but better quality (for both), but bad setting will crash your separation at least on certain songs.

Q: is it possible to use v.2.5 for Melband inference without the need to run the BS model?

A: You can comment out the model(s) you don't want to disable them L621-627 in inference.py [in the line called "vocals_model_names"]

Probably, you could also set BS weight to 0, but it might trigger separation of that model anyway, making it slower.]

- To experiment with parameters for just 4 stems separation, you can use:
 - a) "overlap_demucs" in the Colab (not sure how in this Colab, but for demucs_ft, shifts 10 and overlap 0.1 worked the best for original instrumentals as input)
 - b) shifts for demucs are in line probably 511 (formerly 618 in some other versions iirc):
https://github.com/jarredou/MVSEP-MDX23-Colab_v2/blob/v2.5/inference.py
- In order to bypass models for 2 stem separation to use just instrumentals as input for 4 stem separation, "comment out/delete the name of the models you want to bypass" in the line 621 ([screen](#)). "If you want to use only VOCFT, you have to activate InstVoc too, else it will crash (as it's using InstVoc to fill the spectrum part that is missing because of VOCFT cutoff)" - jarredou

Using other models not included in the Colab

- https://github.com/jarredou/MVSEP-MDX23-Colab_v2/blob/v2.5/inference.py

E.g. in line 452 you can replace Kim model by any other vocal model, and replace that edited in file manager once Colab has executed initialization cell or fork the repo. As for using instrumental Roformer models instead of vocal models, I can't guarantee it will work correctly.

- "Easiest way [to replace MDX HQ model] should be to replace Inst-HQ4 link with HQ5 link line 480

https://github.com/jarredou/MVSEP-MDX23-Colab_v2/blob/36909309efd4a75dab9f1d093a112785a8f560fb/inference.py#L480

If models parameters are same (iirc they are), drop-in replacement should work (and then you control HQ5 with HQ4 settings in colab GUI)"

- Change the args awaited by inference.py accordingly to the ones you've changed in the Colab notebook [if you decide to change models names in he Colab], it's at bottom of inference.py (line 874 and so on)
- Adding e.g. SCNet is not that easy task, it will also require to really add SCNet arch to the script, not only words (add its core files to "modules" folder, import them in main script, check if that work with existing "demix" functions, etc... else it can't work).

<https://github.com/ZFTurbo/Music-Source-Separation-Training/tree/main/models/scnet>

You can study how ZFTurbo is doing it with his script and then try to adapt it to MDX23
Colab. ~jarredou

- What weight you should use for your custom model?

"You must process an evaluation dataset with each model individually, download the separated audio and then use my "weight finder" script ([here \[mirrored\]](#)) with all the separated audios from each model. It will try many different weights until it find the best ones for the given model inputs.

Else you can set "random" weights, process the multisong dataset from MVSEP and upload the separated audios to the quality checker to get the evaluation scores

https://mvsep.com/en/quality_checker (and repeat until you're satisfied)

Download the multisong eval dataset provided on the quality checker link I've shared above. Process the 100 tracks with the model/ensemble you want to evaluate. Download the separated audio.

Rename the files accordingly to guidelines provided in quality checker link, zip them, upload them and wait for the results

All in same folder, and named:

song_000_instrum.wav

song_000_vocals.wav

song_001_instrum.wav

song_001_vocals.wav

song_002_instrum.wav

song_002_vocals.wav

etc...

Software like <https://www.advancedrenamer.com/> can be useful for this"

About

The Colab produces one of the best SDR scores for 4 stems (maybe with slightly better implementation on MVSEP as "Ensemble" 4/5 or more models, although it could be 24 or 32 bit output used for that evaluation which increases SDR (jarredou's v2.3 evaluation was made using 16 bit).

In version 2.4, for 2 stems, UVR/ZFTurbo/Viperx following models are used:

MDX23C Inst Voc HQ/MDX-Net HQ_4 and voc_ft (optionally)/VitLarge/BS-Roformer and for 4 stems:

How MDX23 Colab works under the hood in 2.3 iirc (more or less)

- MDX models vocal outputs (so inversion of one inst model there) + Demucs only vocals>inversion of these to get instrumental>demucs_ft+demucs 6s+demucs+mmi to get remaining 3 stems (weighted) to get remaining 3 stems (all steps weighted). Something in this recipe could be changed since then.

Or differently - "The process is:

1. Separate vocals independently with InstVocHQ, VitLarge (and VOC-FT as opt)
2. Mix the vocals stems together as a weighted ensemble to create final vocals stem
3. Create instrumental by inverting vocals stem against source
4. Save vocals & instrumental stems
- 5 (if 5). Take the instrumental to create the 3 others stems with the multiple demucs models weighted ensembles + phase inversion trick and save them."

Modified inference will probably work locally too, e.g. if you use [that](#) 2.1 repo locally (and probably newer too), but the modified inferences from jarredou crashes the GUI, so you can only use CML version locally in that case.

Usage:

```
python inference.py --input_audio mixture1.wav mixture2.wav --output_folder ./results/
```

To separate locally, it generally requires a 8GB VRAM Nvidia card. 6GB VRAM is rather not enough but lowering overlaps (e.g. 500000 instead of 1000000) or chunking track manually might be necessary in this case. Also, now you can control everything from options: so you can set chunk_size 200000 and single ONNX. It can possibly work with 6GB VRAM that way.

If you have fail to allocate memory error, use --large_gpu parameter.

Chunks option have been deleted from newer Colab options.

Jarredou made some fixes in 2.2.2.x version in order to handle memory better with MDX23C fullband model.

"I've only removed the denoise double pass for demucs_6s, it's activated for other demucs models."

jarredou:

"you can use a workaround to have MDX23C InstVoc-HQ results only with ([dead](#)) settings: (all weights beside MDXv3 set to 0, BigShifts_MDX set to min. of 1, and demucs overlap 0 [at least for vocal_instru_only])

You can use a higher "overlap_MDXv3" value than in the screenshot to get slightly better results.

(and also, as it's only a workaround, it will still process the audio with other models, but they will not be used for final result as their weights = 0)

(MDX23C InstVoc-HQ = MDXv3 here)

You can also use the defaults settings & weights, as it scores a bit higher SDR than InstVoc alone "

Be aware that [2.0](#) version wasn't updated with:

```
!python -m pip install ort-nightly-gpu  
--index-url=https://aiinfra.pkgs.visualstudio.com/PublicPackages/_packaging/ort-cuda-12-nightly/pypi/simple/
```

Or in case of credential issues, you can try out this instead:

```
!python -m pip -q install onnxruntime-gpu --extra-index-url  
https://aiinfra.pkgs.visualstudio.com/PublicPackages/_packaging/onnxruntime-cuda-12/pypi/  
simple/
```

Hence, it's slow (so use 2.1-2.3 instead as they work as intended or add the line at the end of the first cell yourself)

Explanations on features added in [2.2 Colab](#) (2.2 might have more residues vs 2.1) by jarredou

What are BigShifts?

It's based on Demucs' shift trick, but for Demucs it is limited to 0.5 second shifting max (with a randomly chosen value).

Each BigShifts here shifts the audio by 1 second, no more random shifting.

f.e. bigshifts=2, it will do 1 pass with 0 shifting, and a second pass with 1 second shifting, then merge the results

bigshifts=3 means 1 pass with 0 shifting + 1 pass with 1 sec shift + 1 pass with 2 sec shift, etc...

Overlap is doing almost the same thing but at audio chunks level, instead of full audio, and the way overlap is implemented (in MVSEP-MDX23), f.e. with overlap=0.99, first audio chunk will have 1 pass, 2nd audio chunk will have 2 passes, etc... until 99th audio chunk and following ones will have 99 passes. With BigShifts, the whole audio is processed with the same number of passes.

So BigShifts shifts the audio forward one second each time.

Overlap computing is different between MDXv2 models and the other ones in the fork:

For MDXv2 models (like VOC-FT), it uses the new code from UVR and goes from 0 to 0.99.

For MDXv3 (InstVoc) & VitLarge models [introduced in v2.3] it uses code from ZFTurbo (based on MDXv3 code from KUIELab, <https://arxiv.org/abs/2306.09382>) and it goes from 1 to whatever.

I'm using low overlap values in the fork because it's kinda redundant with the BigShifts experimental feature I've added and which is based on Demucs' "shift trick" (described here <https://arxiv.org/pdf/1911.13254.pdf>, chapter 4.4). But instead of doing shifts between 0 and 0.5 sec like Demucs by adding silence before input, BigShifts are much larger (and related to input length). Having larger time shifting gives more amplitude in possible results.

Instead of adding silence before input to shift it, which would be a waste of time & resources as BigShifts can be above 30s or 1 min of shifting, instead, it changes the shifted part position in audio input (like move the 1st minutes of audio at the end of the file before processing and restores it after processing).

Then like Demucs original trick all shifted & restored results are merged together and averaged.

From my tests, it can influence results from -2 SDR to +2 SDR for each shifted results, depending on input and BigShifts value. It's not linear!

Using BigShifts=1 (disabled) and high overlap value probably gives more stable results, in the other end, but maybe not always as high and/or fast as what BigShifts can give.

Weights have been indeed evaluated on MVSep's multisong dataset. I haven't tried every possible settings, but default values should be not far away from optimal settings, if not optimal [already].

Q: Wasn't the BigShifts trick in the MDX23 Colab relying on a slowed-down and sped-up separation ensembling?

I think increasing the parameter too much rather tends to increase bleeding.

A: It's unrelated to bigshifts, but it was doing that for MDX2 models with a cutoff around 16-17khz (to get fullband results from them) but since it's using only fullband models, I've removed that part (in v2.2 iirc)

There are a few other "tricks" used in the fork:

The phase inversion denoise trick (was already in original code from ZFTurbo, also used in UVR):

Some archs (MDXv2 mostly, so VOC-FT here) are adding noise to output signals. So to attenuate it, we process the input 2 times, including one time with phase polarity inverted before processing, and restored after processing. So, only the model noise is phase cancelled when the 2 passes are mixed together. (It doesn't cancel 100%, but it's attenuated). This is also applied to Demucs processing (since original code).

MDXv3 & VitLarge don't seem to add noise (or at insignificant volume) so this trick is not used with these models.

Segment_size (dim_t) original model value is doubled since v2.3 of the fork.

Some benchmarks done by Bas Curtiz showed that it gives a little bit better results ([here](#) with VOC-FT, there's the same benchmark with InstVocHQ model [here](#)).

Multiband ensembling:

I'm using a 2-band ensemble, with different ensemble in frequencies below 10 kHz and above. This is a workaround to get fullband final results even when not fullband models are

part of the ensemble (like VOC-FT). Without it, the instrumental stem, obtained by vocals phase inversion against the input audio would have small permanent vocals bleeding above VOC-FT's cutoff, as phase cancellation would be biased there.

It was a really more essential feature in previous versions when most of the models were not fullband.

VitLarge is not used too in high freq band, but it's a more personal taste (so in the end there's only InstVoc model results above the crossover region)

In fact, alternatively you could separate your instrumental with Demucs single models used by the Colab (demucs_ft, demucs_6s, demucs_mmi) and use SCNet XL and BS-Roformer models from [here](#).

As, along with demucs_ft, they have the best overall SDR for 4 stems separation (actually MDX23C model1 can give interesting results too compared to demucs_ft).

And then perform manual weighted ensemble in DAW by setting volume of the stems manually to your liking after importing and aligning lossless stems.

Because rarely ensembling of more than 4 stems gives good results, IG you could get rid of some demucs models with lower SDR for it (I think the mmi has the lowest SDR, and then 6s).

If it's too much of a hassle, you could change the volume of the stems from a specific model by the same volume.

Guide how to use Colab v 2.5

(reworked Infisrael text)

0. If you plan to use your GDrive for input files, go there now, and create folder called "input" and upload your files there. Create also output folder (not sure if the Colab creates both already). That way you may decrease the time till timeout when the Colab is initialized (esp. for people with slower connection).

Now open the Colab

1. Click the "play" button on the Installation cell and wait until it's finished (should show a green checkmark on the side)

2. Click the "play" button on the GDrive cell.

It will ask you for permission for this notebook to access your Google Drive files, you can either accept or deny it (it is recommended to accept it if you want to use Google Drive as i/o for your files).

After you've done installing it, go to the Separation section below.

Default settings are already balanced in terms of SDR, and too resource-intensive.

3. Click on the play button to start the Separation, **make sure** you uploaded the audio file in the `folder_path`.

After it's done, it will output the stems in the `output_folder`.

Also note, "filename_instrum" is the inversion of the separated vocals stems against the original audio.

"filename_instrum2" when "Separation_mode:" is set to 4 stems (slower) is the sum of the Drums + Bass + Other stems that are obtained by processing "instrum" with multiple Demucs models.

So "instrum" is the most untouched and "instrum2" can have fewer vocals residues or sound a bit muddier.

Experimenting on settings you can set BigShifts to 5 or 7, although it may not give a noticeable difference vs default 3, while increasing separation time severely, but some people use 20 or even 30.

Comparisons of MDX23 (probably v. 2.0) vs single demucs_ft model by A5

The Beatles - She Loves You - 2009 Remaster (24-bit - 44.1kHz)

So I tried out the MDX23 Colab with She Loves You, which is easily the most ratty sounding of all the Beatles recordings, as it is pure mono and the current master was derived from a clean vinyl copy of the single circa 1980. So if it can handle that, it can handle anything. And well, MDX23 is very nice, certainly on par with htdemucs_ft, and maybe even better. I'm surprised. You can hear the air around the drums. Something that is relatively rare with demucs. And the bass is solid, some bleed but the tone and the air, the plucking etc is all there. Plus, the vocals are nicer, less drift into the 'other' stem.

John Lennon - Now and Then (Demo) - Source unknown (16-bit - 44.1kHz)

OK, another test, this time on a John Lennon demo, Now and Then. The vocals are solid, MDX23 at 0.95 overlap is catching vocals that were previously in htdemucs_ft being lost to the piano. So, yeah, it's pretty good. MDX23 is now my favored model. In fact, upon listening to the vocals, it's picking up, from a demo, from a poor recording, on a compact cassette, lip smacks, breathing and other little non-singing quirks. It's like literally going back and having John record in multitrack.

Queen - Innuendo - CD Edition TOCP-6480 (16-bit 44.1kHz)

Every single model fell down with Freddie Mercury's vocals, not anymore. (...) I've heard true vocal stems from his vocals and the MDX23 separation sounds essentially like that. We're now approaching the 'transparent' era of audio extraction.

NOTE: [voc_ft not tested] for Innuendo, will be tested by 07/07/2023

Colab instruction by Infisrael for old versions

Install it, click on the play button and wait until it's finished (should show a green checkmark in the side).

It will ask you for permission for this notebook to access your Google Drive files, you can either accept or deny it (it is recommended to accept it if you want to use Google Drive as i/o for your files).

After you've done installing it, go to the configuration, it's below the 'Separation' tab.

<https://i.imgur.com/qD9jsYG.png> (dead)

(Recommended settings)

Input ``overlap_large`` & ``overlap_slow`` with what you desire, at the highest (1.0), it will process slower but will give you a better quality. The default values for large are (0.6), and for small (0.5) [with 0.8 still being balanced in terms of speed and quality].

Input ``folder_path`` with the folder destination where you have uploaded the audio file you'd like to separate

Input ``output_folder`` with the folder you'd like the stems to be separated

Change your desired path after `/content/drive/MyDrive/`, so for example:

```
> `folder_path: /content/drive/MyDrive/input`  
> `output_folder: /content/drive/MyDrive/output`
```

You can also make a use of ``chunk_size`` and put it in a higher value by a little, but if you experience memory issues, lower it, default value for it is 500000.

Afterwards, click on the play button to start the separation, **make sure** you uploaded the audio file in the `folder_path` you provided.

After it's done, it will output the stems in the `output_folder`.

Also note, ``filename_instrum`` is the inversion of the separated vocals stems against the original audio.

``filename_instrum2`` is the sum of the Drums + Bass + Other stems that are obtained by processing ``instrum`` with multiple Demucs models.

So ``instrum`` is the most untouched and ``instrum2`` can have fewer vocals residues.

Installing the Colab locally

NVIDIA 12GB VRAM GPU recommended

"I think it's possible to use Colab notebook .ipynb files locally with anaconda and jupyter, but I've never tried. [Or:]

You can git clone the repo, install requirements and use the inference.py script, but the command line can be really long to type manually (on Colab it's managed with the GUI):

```
python inference.py \
--input_audio "{file_path}" \
--large_gpu \
--BSRoformer_model {BSRoformer_model} \
--weight_BSRoformer {weight_BSRoformer} \
--weight_Kim_MelRoformer {weight_Kim_MelRoformer} \
--weight_InstVoc {weight_InstVoc} \
--weight_InstHQ4 {weight_InstHQ4} \
--weight_VOCFT {weight_VOCFT} \
--weight_VitLarge {weight_VitLarge} \
--overlap_demucs {overlap_demucs} \
--overlap_VOCFT {overlap_VOCFT} \
--overlap_InstHQ4 {overlap_InstHQ4} \
--output_format {output_format} \
--BigShifts {BigShifts} \
--output_folder "{output_folder}" \
--input_gain {input_gain} \
{filter_vocals} \
{restore_gain} \
{vocals_only} \
{use_VitLarge_} \
{use_VOCFT_} \
{use_InstHQ4_} \
{use_InstVoc_} \
{use_BSRoformer_} \
{use_Kim_MelRoformer_}
```

Q: How do you use the example {useVitLarge}
like the other stuff ik how to use

A: These last arguments are boolean based, there are generated before the command line and depending on the option selected in the GUI with:

```
use_InstVoc_ = '--use_InstVoc' #forced use
use_BSRoformer_ = '--use_BSRoformer' #forced use
use_Kim_MelRoformer_ = '--use_Kim_MelRoformer' #forced use

use_VOCFT_ = '--use_VOCFT' if use_VOCFT is True else "
use_VitLarge_ = '--use_VitLarge' if use_VitLarge is True else "
use_InstHQ4_ = '--use_InstHQ4' if use_InstHQ4 is True else "
restore_gain = '--restore_gain' if restore_gain_after_separation is True else "
```

```
vocals_only = '--vocals_only' if Separation_mode == 'Vocals/Instrumental' else "
filter_vocals = '--filter_vocals' if filter_vocals_below_50hz is True else "
```

Q: So you don't need to use them?

Only using the ones with the two -- before right

A: For example, if you want to activate vocals filtering below 50hz, you add "--filter_vocals" to the command line

Q: How do you do this

A: ([click](#))

Q: oh yeah I just have to change the default number then right

It works [#general](#)

A: If you have multiple GPUs and the CUDA one is not labelled device "0", maybe that can be the cause too, it's hardcoded for Colab, but you can change it in first lines of inference.py file gpu_use = "0"

If your GPU is not detected in Anaconda, use Python (can be 3.12). If it's the same:

<https://pytorch.org/get-started/locally/#start-locally>

Where it says "run this command" I basically uninstalled the modules it had in there so I did pip uninstall torch torchvision torchaudio
then ran that command to install it
and it fucking fixed it (knock)

1.0 original code used kim vocal 1 (later 2), kim inst and (at least for 4 stems) Demucs models.

KaraFan by Captain FLAM

(2 stems)

[Colab](#) w/ more models (AI Hub fork, also fixed), fixed org. [Colab](#), org. [Colab](#) (slow), [GUI](#), GH documentation

[How to install it locally \(advanced\)](#), alt. [tutorial](#),
or [easy](#) instruction

Should work on Mac with Silicon or AMD GPU (although not for everyone)
& Linux with Nvidia or AMD GPU
& Windows probably with at least Nvidia GPU, or with CPU (v. slow)

- For Colab users - create “Music” in the main GDrive directory and upload your files for separation there (the code won’t create the folder on the first launch).
- Sometimes you’ll encounter soundfile errors during separation. Just retry, and it will work

KaraFan (don’t confuse with KaraFun) is a direct derivative of ZFTurbo’s MDX23 code forked by jarredou, but with further tweaks and tricks in order to get the best quality of instrumentals and vocals sonically, but without overfocusing on SDR only, but the overall sound.

Its aim is to not increase vocal residues without making instrumentals too muddy like e.g. sometimes HQ_3 model does, but without having so many vocal residues as MDX23C fullband model (but it depends on chosen preset).

Since v. 4.4 and 5.x you have five presets to test out.

Presets 3 and 4 are more aggressive in canceling vocal residues (P4 can be good for vocals).

Preset 5 (takes 12 minutes+ on the slowest setting for 3:25 track on T4) has more clarity of instrumentals over presets 3 and 4, but also more vocal residues (although less than P1 and P2 (takes 8 minutes for 3:24 track on the slowest setting)).

On 23.11.24 “Preset 5 was corrected to be less aggressive as possible”. All the below Preset 5 descriptions refer to the old P5. The original preset 5 is [here](#), and is less muddy, but has more vocal residues (at least the original preset contains more models and is slower).

Speed and chunks affect quality. The slower, the muddier, but also slightly less vocal residues, although they’ll be still there (just slightly quieter). I’d recommend the “fastest” Speed setting and 400K chunks for the current P5 (tested on 4:07 song, may not work for longer tracks).

- If you replace Inst Voc HQ1 model by HQ2 using AI Hub fork in current P5, the instrumental will be muddier.
- To preserve instruments which are counted as vocals by other MDXv2 models, use [these](#) preset’s 5 modified settings - they have more clarity than P5 and preserve hi-hats better. But to preserve the same processing time as in P5, but setting “Speed” slider to medium, in this case will result in more constant vocal residues vs P5 with the slowest setting (too much at times, but it might serve well for specific song fragments). It will take 12 minutes+ for 3:24 track on medium. Debug and God mode on the screenshot are unrelated and optional.
- To fix issues with saxophone in P5 use [these](#) settings. They even have more clarity than the one above, but also more hearable vocal residues. It helps to preserve instruments better than the setting from the above. It can be better than P2 - less hearable consistent

vocal residues, but in similar amount, while on other artists sax preset even gives more vocal residues than P2. Sax setting is worse in preserving piano than the setting above.

- Using the slowest setting here in sax fix preset will result in disconnection of runtime with free T4 after 28 minutes of processing, but it should succeed anyway (result files might be uploaded on GDrive after some time anyway).

Vs medium, the slowest setting gives more muffled sound, but not always less vocal residues. It can be heard the best in short parts with only vocals. 18 minutes for 4:07 track on Fast setting (God Mode and Debug Mode are disabled in KaraFan by default).

After 3-4 ~18 minutes separations (in this case not made in batch, but with manually changed parameters in the middle), when you terminate and delete environment, you might be not able to connect with GPUs again as the limit will be reached unless you switch Colab account (mount the same GDrive account as Colab to avoid errors)

- Preset 5 provides more muffled results than the two settings above, but with good balance of clarity and vocal residues. Sometimes this one has less vocal residues, sometimes 16.66 MDX23C model on MVSEP (or possibly a bit older HQ_1 model in UVR), it can even depend on a song fragment. Using newer MDX23C HQ 2 in P5 instead of MDX23C HQ doesn't seem to produce better results

After 5th separation (not in batch) you must start your next separation very fast because or you'll run out of Colab free limit when GUI is in idle state. In such case, switch Colab account, and use the same account to mount GDrive (or you might encounter error).

Comparisons above made with normalization disabled and 32-bit float setting

The code handles mono and 48kHz files too, 6:16 (preset 3) tracks, and possibly 9 minutes tracks too (but can't tell if with all presets). It stores models on GDrive, which takes 0,8-1,1GB (depending on how many models you'll use). One 4:07 song in 32-bit float with debug mode enabled (all intermediate files will be kept) will take 1,1GB on GDrive. Instrumentals will be stored in files marked as Final (in the end), Music Sub (can sound a bit cleaner at times, but with more residues), and Music Extract (from specific models).

Older 1.3 version Colab fork by Kubinka was deleted.

Colab fork made by AI HUB server members also includes MDX23C Inst Voc HQ 2 and HQ_4 models, and contains slow separation fix from the "fixed Colab".

KaraFan used to have lots of versions which differ in these aspects with an aim to have the best result in the recent Colab/GUI version. E.g. v.3.1 used to have more vocal residues than in 1.3 version and even more than in HQ_3 model on its own, and it got partially fixed in 3.2 (if not entirely). But 1.3 irc, had some overlapped frequency issue with SRS disabled, which makes the instrumentals brighter, but it got fixed later. The current version at the time of writing this excerpt is 4.2, with pretty good opinions for v.4.1 shortly before.

Colab troubleshooting

- (no longer necessary in the fixed [Colab](#)) If you suffer from very slow or unfinishable separations in the Colab using non-MDX23C models (e.g. stuck on voc_ft without any progress), use fixed Colab (the onnxruntime-gpu line added in the end of the first cell)
 - Contrary to every other Colab in this document, KaraFan uses a GUI which launches after executing inference cell. It triggers Google's timeout security checks frequently esp. in free Colab users, because Google behaves like the separation is not being executed where you do it in GUI, and it's generally against their policies to execute such code instead pasting commands to execute in Colab cells directly. The same way many RVC Colabs got blocked by Google, but this one is generally not directly for voice cloning, and is not very popular yet, so it wasn't targeted by Google yet.
 - Once you start separation, it can get you disconnected from runtime quickly, especially if you miss some multiple captcha prompts (in 2024 captchas stopped appearing at all, so the user inactivity during separation process seems to be no longer checked).
 - After runtime disconnection error, output folder on e.g. GDrive can be still constantly populated with new files, while progress bar is not being refreshed after clicking close or even after closing your tab with Colab opened. At certain point it can interrupt the process, leaving you with not all output files. Be aware that final files always have "Final" in their names.
 - It can consume free "credits" till you click Environment>Terminate session. It happens even if you close the Colab tab. You can check "This is the end" option so the GUI will terminate the session after separation is done to not drain your free limit.
 - (rather fixed) As for 4.2 version, session crashes for free Colab users can occur, due to running out of memory. You can try out shorter files.
Currently, if you rename your output folder with separation, and retry separation, it will look for the old folder with separation to delete, and return the error, and running the GUI cell again may cause disappearing of GUI elements.
it's a default behavior of Colab and IPython core : Sync of files the Colab sees is not real time
- Two possible solutions:
- wait until sync with Google Drive is done
 - restart & run Colab
- Sometimes shutting down your environment in Environment options and starting over might do the trick if something doesn't work. E.g. (if it wasn't fixed), when you manipulate input files on GDrive when GUI is still opened, and you just finished separation, you might run into an error when you start separating another file with input folder content changed.

In order to avoid it, you need to run the GUI cell again after you've changed the input folder content (IRC it's "Music" folder by default). Maybe too low chunks (below 500k for too long tracks if something hasn't changed in the code). Also, check with some other input file you used before and worked before first.

Also, be more specific about what doesn't work. Provide screenshot and/or paste the error.

- You can be logged to a maximum of 10 Google accounts at the same time. You can't log out of any of these single accounts on PC in browser. The only way is to do it on your Android phone, but it might not fix the problem, as it will tell "logged out" on that account on PC, and logging into other one might not work and the limit will be still exceeded. At this situation you can only log out from all accounts (but it will break accounts order, so any authorizations set to specific accounts in your bookmarked links will be messed up - e.g. those to Colab, GDrive, Gmail, etc. I mean: /u/0 and in Colab authuser= in links. Easier way to access to extra Google account will be to log into it from Incognito mode.

If you possess lots of accounts and you don't log for some for 2 years, Google can delete it. To avoid it, create YT channel on it, and upload at least one video, and the account won't be deleted.

Tests of four presets of KF 4.4 vs MDX-UVR HQ_3 and MDX23C HQ (1648)

(noise gate enabled a.k.a. "Silent" option)

Not really demanding case, so without modern vocal chain in the mix, but probably enough to present the general idea of how different presets sound here. So, more forgiving song to MDX23C model this time, and less aggressive models with more clarity.

Genre: (older rap) Title: (O.S.T.R. - Tabasko [2002])

BEST Preset : 3

Music :

Versus P4, hi-hats are preserved better in P3.

Snare in P3 is not so muffled like in P4.

HQ_3 has even more muffled snares than in P4.

P3 still had less vocal residues than MDX23C HQ 1648 model, although the whole P3 result was more muffled, but residues are smartly muffled too.

MDX23C had like more faithfully sounding snares than P3, to the extent that they can be perceived brighter (but vocal residues, even on a more forgiving song like this, are more persistent in MDX23C than in P3).

Sometimes it depended on specific fragment where P4 and where P3 has more vocal residues in that specific case, so P3 turned out being pretty much balanced, although P4 had less consistent vocal residues, although still not so few like HQ_3, but it's not that much of a problem (HQ_3 is really muffled). If it was 4 stems, then I'd describe P3/4 as having very good "other" stem but drums too as I mentioned.

WORST Preset (in that case) : 1

Music : Too much consistent vocal residues

There's a similar situation in P2, but at least P2 has brighter snares than even MDX23C.

In other songs, P1 can be better than P2, leaving less vocal residues in specific fragments for a specific artist, but noticeably more for others.

Preset 4 with setting slow (but not the slowest) takes 16 minutes for 5 minutes song on T4 in free Colab (performance of ~GTX 3050). For 3:30 track, it takes 13:30 for the slowest setting. In KF 5.1 with default chunks 500K and slowest setting, for 4:50 song and preset 2 it took <10 minutes, preset 3, 12 minutes.

VS preset 3, the one from the [Screenshot](#) (now added as preset 5) is more noisy and has more vocal residues, mainly in quiet places or when there is no instrumental. Processing time for 6:16 track on medium setting is 22:19 minutes. But it definitely has more clarity over preset 3. And there is still less vocal residues than in Preset 1 and 2, which have more clarity, but tend to have too many vocal residues in some tracks. Hence, preset 5 is the most universal for now.

For future: "To add or remove some models u need to edit the .csv file

<https://github.com/Eddycrack864/KaraFan/blob/master/Data/Models.csv>

with the model info (Only MDX23C or MDX-NET) u can found the model info on the model_data_new.json:

https://raw.githubusercontent.com/TRvlvr/application_data/main/mdx_model_data/model_data_new.json u need to find the hash of the model. And.... that's it! (Not Eddie)

Ripple/Capcut/SAMI-Bytedance/Volcengine/BS-RoFormer (2-4 stem)

Output quality in Ripple is: 256kbps M4A (320kbps max) and lossless (introduced later).

50MB upload limit, 4 stems

Min. iOS version: 14.1

Ripple is only for US region (which you can change, more below)

Ripple no longer separates stems (there's an error "couldn't complete processing please try again")

Ripple for iOS: <https://apps.apple.com/us/app/ripple-music-creation-tool/id6447522624>

Capcut for Android: <https://play.google.com/store/apps/details?id=com.lemon.lvoverseas>
(separation only for Pro, Indian users sometimes via VPN)

Capcut a.k.a. Jianying (2 stems) works also on Windows (only in Jianying Pro, separation option is available)

Can be used instead of Ripple if you're on unsupported iOS below 14.1 or don't have iOS. To get Ripple you can also use a virtual machine remotely instead (instructions below). Ripple can also be run on your M1 Mac using app sideloading (instructions below).

Ripple = better quality than CapCut as of now (and fullband)
with fixed the click/artifacts using cross-fade technique between the chunks.

Capcut = “the results are really low quality but if you export the instrumental and invert it with the lossless track, you will get the vocals with the noise which is easy to remove with mdx voc ft for example, then you can invert the lossless processed vocals with the original and have it in better quality.

The vocals are very clean from cap cut, almost no drum bleed”

Ripple and Capcut uses SAMI-Bytadance arch (later known as BS-Roformer), but it's a different model with worse SDR than on the leaderboard. It was developed by Bytedance (owner of TikTok) for MDX23 competition, and holds the top of our MVSEP leaderboard. It was published on iOS and for the US region as “Ripple - Music Creation Tool” app. Furthermore, it's a multifunctional app for audio editing, which also contains a 4 stem separation model. Similar situation with Capcut (which is 2 stems only IRC). The model itself is not the same as for MDX23 competition (SAMI ByteDance v1.0), as they said, models for apps were trained on 128kbps mp3 files to avoid copyright issues, but it's the same arch, just scores a bit lower (even when exported losslessly for evaluation). SDR for Ripple is naturally better than for Capcut.

Seems like there is no other Pro variant for Capcut Android app, so you need to unlock regular version to Pro.

At least the unlocked version on apkfile.me have a link to the regular version, so it doesn't seem to be Pro app behind any regional block. But - "Indian users - Use VPN for Pro" as they say, so similar situation like we had on PC Capcut before. Can't guarantee that unlocked version on apkfile.me is clean. I've never downloaded anything from there.

Bleeding

Bas Curtiz found out that decreasing volume of mixtures for Ripple by -3dB (sometimes -4dB) eliminates problems with vocal residues in instrumentals in Ripple. [Video](#)

This is the most balanced value, which still doesn't take too many details out of the song due to volume attenuation.

Other good values purely SDR-wise are -20dB>-8dB>-30dB>-6dB>-4dB> /wo vol. decr.

The method might be potentially beneficial for other models, and probably work best for the loudest tracks with brickwalled waveforms.

The other stem is gathered from inversion to speed up the separation process. The consequence is bleeding in instrumentals.

- If you suffer from bleeding in other stem of 4 stems Ripple, beside decreasing volume by e.g. 3/4dB also “when u throw the ‘other stem’ back into ripple 4 track split a second time, it works pretty well [to cancel the bleeding]”

The forte of the Ripple is currently vocals - the algo is very good at differentiating what is vocals and what is not, although they can sound “filtered” at times.

Currently, the best SDR for public model/AI, but it gives the best results for vocals in general. For instrumentals, it rather doesn’t beat paid Dango.ai (and rather not KaraFan and HQ_3 or 1648/MDX23C fullband too).

It’s good for vocals, also for cleaning vocal inverts, and surprisingly good for e.g. Christmas songs, (it handled hip-hop, e.g. Drake pretty well). It’s better for vocals than instrumentals due to residues in other stem - bass is very good, drums also decent, kicks even one if not the best out of all models, as they said some fine-tuning was applied to drums stem. Vocals can be used for inversion to get instrumentals, and it may sound clean, but rather not as good as what 2 stem option or 3 stem mixdown gives as output is lossy.

Capcut (2 stems only)

<https://www.capcut.cn/>

It is a new Windows and Android app which contains the same arch as Ripple inst/vocal, but lower quality model, and without an option of exporting 4 stems.

It normalizes the input, so you cannot use Bas’ trick to decrease volume by -3dB to workaround the issue of bleeding like in Ripple (unless you trick out the CapCut, possibly by adding some loud sound in the song with decreased volume).

“At the moment the separation is only available in Chinese version of Windows app which is jianyingpro, download available at capcut.cn [probably here - it’s where you’re redirected after you click “Alternate download link” on the main page, where download might not work at all]

Some people cannot find the settings on [this](#) screen in order to separate.

Separation doesn’t require sign up/login, but exporting does, and requires VIP, which is paid depending on whether you’re from rich or poor country, then it’s free.

- There’s a workaround for people not able to split using Capcut for Windows in various regions.

- Bas Curtiz’ new video on how to install and use Capcut for separation incl. exporting:
<https://www.youtube.com/watch?v=ppfyl91bJlw>

"It's a bit of a hassle to set it up, but do realize:

- This is the only way (besides Ripple on iOS) to run ByteDance's model (best based on SDR).
- Only the Chinese version has these VIP features; now u will have it in English
- Exporting is a paid feature (normally); now u get it for free

The instructions displayed in the video are also in the YouTube description."

- mitmproxy [script](#) allowing to save to FLAC instead of AAC (although it just reencodes from AAC 113kbps with 15.6kHz lowpass filter). It's a bit more than script. See the [full](#) tutorial.

- For some people using mitmproxy scripts for Capcut (but not everyone), they "changed their security to reject all incoming packet which was run through mitmproxy. I saw the mitmproxy log said the certificate for TLS not allowed to connect to their site to get their API. And there are some errors on mitmproxy such as events.py or bla bla bla... and Capcut always warning unstable network, then processing stop to 60% without finish."

~hendry.setiadi

"At 60% it looks like the progress isn't going up, but give it idk, 1 min tops, and it splits fine." - Bas

"in order to install pydub within mitmproxy, you additionally need to:

open up CMD

pip install mitmproxy

pip install pydub"

- IntroC created a [script](#) for mitmproxy for Capcut allowing fullband output, by slowing down the track. [Video](#)

Older Capcut instruction:

The [video](#) demonstration of below:

0. Go offline.

1. Install the Chinese version from [capcut.cn](#)

2. Use [these](#) files copied over your current Chinese installation in:

C:\Users\your account\AppData\Local\JianyingPro

Don't use English patch provided below (or the separation option will be gone)

3. Now open CapCut, go online after closing welcome screen, happy converting!

4. Before you close the app, go offline again (or the separation option will be gone later).

! Before reopening the app, go offline again, open the app, close welcome screen, go online, separate, go offline, close. If you happen to miss that step, you need to start from the beginning of the instruction.

(no longer works after 4.6 to 4.7 update, as it freezes the app) The only thing that seems to enable vocal separation without requiring replacing everything, is to replace that

[SettingsSDK](#) folder contents inside User Data. It's probably the settings_json file inside responsible for that.

FYI - the app doesn't separate files locally.

The quality of separation vs Capcut is not exactly the same as Ripple. Seeing by spectrograms, there is a bit more information in vocals in Capcut, while Ripple has a bit more information in spectrum in instrumentals.

Separated vocal file is encrypted and located in

C:\Users\yourusername\AppData\Local\JianyingPro\User Data\Cache\audioWave"

The unencrypted audio file in AAC format is located at \JianyingPro
Drafts\yourprojectname\Resources\audioAlg (ends with download.aac)

"To get the full playable audio in mp3 format, a trick that you can do is drag and drop the download.aac file into Capcut and then go to export and select mp3. It will output the original file without randomisation or skipping parts"

(although it resulted in VIP option disappearing but Bas somehow managed to integrate it in his new video tutorial, and it started to work, English translation isn't the culprit of the problem, but if you use both language pack and SettingsSDK folder from above)

You can replace the zh-Hans.po file with [English one](#) to have English language on Chinese version of the app possessing separation feature in:

jianyingpro/4.6.1.10576/Resources/po

While you can't use that language pack, you can always use Google Translate to transform Chinese into your own language on a screen of your smartphone.

https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQwk_qynMHMwquSfQZFrrn30F355lhta_GHQNo7vhnPUhfj-kUiqSRBiLQbPlgmB5Gqro&usqp=CAU

<https://support.google.com/translate/answer/6142483?hl=en&co=GENIE.Platform%3DDesktop>

"Trying out capcut, the quality seems the same as the Ripple app (low bitrate mp3 quality) at least the voice leftover bug is fixed, lol"

Random vocal pops from Ripple are fixed here.

Also, it still has the same clicks every 25 seconds as before in Ripple.

Capcut adds 1024 extra samples at the beginning, and 16 extra samples at the end of the file.

How to change region to US in order to make Ripple work on iOS

in Apple App Store to make "Ripple - Music Creation Tool" (SAMI-Bytedance) work.
<https://support.apple.com/en-gb/HT201389>

- Bas' [guide](#) to change region to US for Ripple on iOS

<https://www.bestrandoms.com/random-address-in-us>

Or use this Walmart address in Texas, the number belongs to an airport.
Do it in App Store (where you have the person-icon in top right).
You don't have to fill credit cards details, when you are rejected,
reboot, check region/country... and it can be set to the US already.
Although, it can happen for some users that it won't let you download anything forcing your
real country.

"I got an error because the zip code was wrong (I did enter random numbers) and it got stuck even after changing it.
So I started from the beginning, typed in all the correct info, and voilà"

If "you have a store credit balance; you must spend your balance before you can change stores".

It needs (an old?) a simcard to log your old account out if necessary

Ripple on Windows or MacOS

- Another way to use **Ripple** without Apple device -
virtual machine

Sideloaded of this mobile iOS app is possible on at least M1 Macs.

- Saucelabs

Sign up at <https://saucelabs.com/sign-up>

Verify your email, upload this as the IPA:

<https://decrypt.day/app/id6447522624/dl/cllm55sbo01nfoj7yifiyucaa>

Rotating puzzle captcha for TikTok account can be tasking due to low framerate. Some people can do it after two tries, others will sooner run out of credits, or completely unable to do it.

- <https://mobiledevice.cloud/>

Mobile device cloud

- Scaleway

"if you're desperate you can rent an M1 Mac on scaleway and run the app through that for \$0.11 an hour using this <https://github.com/PlayCover/PlayCover>"

IPA file:

https://www.dropbox.com/s/z766tfysix5gt04/com.ripple.ios.appstore_1.9.1_und3fined.ipa?dl=0

"been working like a dream for me on an M1 Pro... I've separated 20+ songs in the last hour"

More info:

- <https://cdn.discordapp.com/attachments/708579735583588366/1146136170342920302/image.png>

- "keep in mind that the vm has to be up for 24 hours before you can remove it, so it'll be a couple bucks in total to use it"

Fixing chunking artefacts (probably fixed)

- Every 8 seconds there is an artifact of chunking in Ripple. Heal feature in Adobe Audition works really well for it:

<https://www.youtube.com/watch?v=Qqd8Wjqtx-8>

-The same explained on RX10 example and its Declick feature:

<https://www.youtube.com/watch?v=pD3D7f3ungk>

Volcengine (a.k.a. The sami-api-bs-4track - 10.8696 SDR Vocals)

<https://www.volcengine.com/docs/6489/72011>

Ripple/SAMI Bytedance's API was found. If you're Chinese, you can go through it easier - you need to pass the Volcengine facial/document recognition, apparently only available to Chinese people

We already evaluated its [SDR](#), and it even scored a bit better than Ripple itself.

"API from volcengine only return 1 stem result from 1 request, and it offers vocal+inst only, other stems not provided. So making a quality checker result on vocal + instrument will cost 2x of its API charging."

Something good is that volcengine API offers 100 min free for new users"

API is paid 0.2 CNY per minute.

It takes around 30 seconds for one song.

It was 1.272 USD for separating 1 stem out MVSEP's multisong dataset (100 tracks x 1 minute).

"My only thought is trying an iOS Emulator, but every single free one I've tried isn't far-fetched where you can actually download apps, or import files that is"

So far, Ripple didn't beat voc_ft (although there might be cases when it's better) and Dango.

Samples we got months ago are very similar to those from the app, also *.models files have SAMI header and MSS in model files (which use their own encryption), although processing is probably fully reliable on external servers as the app doesn't work offline (also model files are suspiciously small - few megabytes, although it's specific for mobilenet models). It's probably not the final iteration of their model, as they allegedly told someone they were afraid that their model will leak, but better than the first iteration judging by SDR with even lossy input files.

Later they told that it's different model than the one they previously evaluated, and that time it was trained with lossy 128kbps files due to some "copyright issues".

"One thing you will notice is that in the Strings & Other stem there is a good chunk of residue/bleed from the other stems, the drum/vocal/bass stems all have very little to no residue/bleed" doesn't exist in all songs.

It's fully server-based, so they may be afraid of heavy traffic publishing Ripple worldwide, and it's not certain whether it will happen.

Thanks to Jorashii, Chris, Cyclrclicly, anvuew and Bas, Sahlofolina.

Press information:

<https://twitter.com/AppAdsai/status/1675692821603549187/photo/1>

<https://techcrunch.com/2023/06/30/tiktok-parent-bytedance-launches-music-creation-audio-editing-app/>

Site:

<https://www.ripple.club/>

BS-RoFormer

Used architecture in Capcut/Ripple (now defunct). Their paper was published and later reimplemented by lucidrains for training and inferencing:

<https://github.com/lucidrains/BS-RoFormer>

Later, [Mel-Band RoFormer](#) based on band split was released, which is faster, but doesn't provide such high SDR as BS. Mel variant might require some revision of the code, and its paper might lack some features need to keep up SDR-wise with extremely slow BS original variant. On paper, it should be better than BS-Roformer, but for some reason, models trained with Mel have worse results than with BS-Roformer (so probably problem with reimplementation from paper). Kim reworked her config, so the results with Mel models improved, but still are a tad lower than BS-Roformer. ZFTurbo includes training and inference of Roformers in his repository on GitHub.

For more information, check the [training](#) section.

About ByteDance

Winners of MDX23 competition. They said at the beginning, that it utilizes novel arch (so no weighting/ensembling of existing models). In times of v.0.1 seemingly the best vocals, not so good instrumentals, as it was once said by someone who heard samples, but they came a long way lately. It's all about their politics. It's a Chinese company responsible for TikTok, famous for d**k moves outside China - manipulating their algorithms - encourage of stupidity outside China, and greedy, wellness-centered attitudes for users in China (the app is currently banned in China), manipulating their algorithms to promote only black-white relationships in western countries, spying on users copying their clipboard, spying even on journalists to find their sources of information about the company, unauthorized remote access to TikTok user data from China, and also, a subject to ban in US and other countries for bad influence on children, data infringement by storing non-China users data directly on their servers which is against the law of many countries (there were some actions taken on it later). [Decompiling TikTok analysis](#) (tons of spying improper behavior of the app). Currently, Bytedance is only around 40% owned by founders, Chinese investors, and their employees and the rest (60%) state global investors (incl. lots of American) and is pushed to sale more stakes to US risking US ban on the app.

They said, the CEO, told them to hold this ByteDance arch for two years for themselves. Initially they had plans to release it in some kind of app, firstly at the end of June, later something was planned at the end of year, later they said something about two years (maybe more about open sourcing, but we can't have our hopes high). Previously, they said the case of open sourcing/releasing was stuck in their legal department. Later they told they used MUSDBHQ+500 songs for their dataset. These 500 songs could have been obtained illegally for training (although everyone does it), but they might be extremely cautious about it (or it's just an excuse). Eventually, they released Ripple and Capcut. Then they released the paper for the Bs/Mel archs, and it was implemented and coded by Lucidrains, so later could be used for training.

Later, they seemingly spread information among users privately, that despite the similarities in SDR, the 18.75 score is a result of a trolling, someone other than ByteDance. Some people favoring ByteDance were rumored for disruptive, trolling behavior on our server too, harassing other users, or just being unkind to others etc. Besides, the same person responsible, was also the most informed about ByteDance next moves, and was also changing nicknames or accounts frequently. Also possessed great ML knowledge. Many coincidences. In the end, the same user, zmis (if you see the details of the account above), was behind a lot of newly created, accounts, which were banned on our server.

The same day or in very similar period, a new account was created, conducting the same behavior, when previous was banned.

The main core of their activities, was spreading misinformation about SDR metrics, telling that is the most important thing in the world, because their own arch is good at it, hence the narration.

So don't bother, and do your good job not feeding troll from other company. They don't like competition, doing their own moves behind backdrops and become better.

It's not impossible to fake SDR results in the MVSEP leaderboard. For current public archs, you'd need to feed your dataset both by the songs in the evaluation dataset, keeping your regular big dataset in place, so you simply lose evaluation factor of this leaderboard, or you can simply mix your result stems with original stems. "SDR focuses more on lower frequencies, it can easily be fooled into giving a higher score if the lower frequencies are louder, Bas tested this theory and confirmed it". you can boost the bass, and it will score +1 or +2 sdr higher or something, that's why It's not always reliable" - becruily

Those results, which are not faked, are at least those, which were uploaded by various users evaluating the same public, available for offline use models, but usually uploaded with various parameters which affects SDR (so usually the better parameters, the higher SDR, but not always), remain consistent among various users evaluations with similar parameters and inference code, so scalability is correct and preserved, thus the results weren't faked, and can be reproduced with similar SDR. For the other scores from unpublic inferences/models/methods, we simply trust ZFTurbo and rather viperx too, as they're/were our trusted users for years. Also, the leaderboard in the current multisong dataset tends to give better SDR to the results with more residues on different occasions before, so the chart is simply not fully reliable for that, but rather not manipulated in its core either. It's more a nature of SDR measurement and/or used dataset.

ViperX trained the first community BS-Roformer model similar to SAMI v1.0 model, although 2 stems (and lower scoring Mel at the time). His BS model sounds similar to Ripple (although it's only 2 stem, while 4 stem Ripple variant scores a bit higher than the 2 stem variant, but still lower than ViperX and v1.0). Then there were a lot of community trained models like private one by ZFTurbo, and fine-tunes by various users (Kim, Unwa, Bas, ZFTurbo)

Bas tried to train a model purely on multisong dataset only, but failed to surpass the SDR score of a 1.1 Bytedance's model anyway. v1.1 has new arch enhancements to the arch, and will be presented on ISMIR2024 (white paper is already out; link in the [Training](#) section).

Drumsep - single percussion instruments separation

If you want to further separate single instruments from drums stem separated with e.g. MDX23 Colab, Mel-Roformer drums on x-minus.pro premium, MVSEP, or Demucs_ft (not necessarily BS-Roformer SW) into: hihat, cymbals, kick, snare and more, you might want to check below solutions. Sampling from such separated stems might be not the best idea due to the quality (see [here](#) for free Drumclone plugin allowing even different types of synthesised kicks from mixture; [video](#)). But e.g. it serves well for purposes of conducting new mixes/remasters of the same songs or separated instrumentals, e.g. when it's overlapped with better quality, previously separated drums stem. It might give interesting results when aligned with the original drums, and rebalanced with effects (drums stem might

end up louder in the mix than separated percussion, as most likely it will still have better quality). Check out these drum replacers.

To potentially increase drumsep models separation quality, “try using a small pitch shift up or down, like +/- 1 or 2 semitones (...) can sometimes help bring out the lows or highs if they seem weak.” (CZ-84)

Also, consider using good instrumental model before using 4 stem model for drums (if it's not instrumental already), to enhance drums stem, and then to enhance drumsep result.

Some drumsep models might have a bug where “a small, but relevant portion of audio is being lost when the [drumsep] model is being used”

“The solution is to invert the phase on all the drum stems into the original file and save that as its own file, making your own “other” file”. It has been fixed on MVSEP.

Mel-Roformer MVSEP drumsep models

1) 4 stems v2 (kick, snare, toms, cymbals) - “It gives the best metrics with a big gap for kick, snare and cymbals.” - ZFTurbo. The old v1 below was removed.

([metrics](#); only toms are worse SDR-wise vs previous SCNet Drumsep models below)

1) ~~4 stems v4~~ removed (kick, snare, toms, cymbals) - average SDR of hihat ride, crash is 11.52 (but in one stem) and so far it's the best SDR out of all models (even vs the previous ensemble consisting of three MDX23C and SCNet models).

2) 6 stems (kick, snare, toms, hihat, ride, crash) - average SDR of hihat ride, crash is 8.18 (but from separated stems), while

The snare in 1) has the best SDR out of all available models.

Kick and toms are still the best SDR-wise in the previous 3x MDX23C and SCNet ensemble (new ensemble with these new Mel-Roformers so far)

- The new models “are very great for ride/crash/hh. And overall, they have the best metrics for almost all stems.” - ZFTurbo

[SDR/L1 Freq/bleedless/fullness chart](#) of all models

[Evaluations on new dataset](#) (esp. check Log WMSE Results with ““bypass_filter” with torch_log_wms, ([good] at least for drums or anything rich in low frequency content)” - jarredou

Sometimes the newer jarredou’s drumsep 6 stems model below can serve to clean up “upper frequency range of snare hits” in the cymbals stems in “either MelRoFormer or SCNet-XL four-or-six stem DrumSep models” - Dyslexicon

“The core problem is that the main MVsep Drums model which is used by everything-including the Drumsep models- is not purely drums, it's mixed with other percussion which taints things.” - godzfire

SCNet MVSEP drumsep models

Better SDR than MDX23C and Demucs models above

- MVSEP 8 stems ensemble of all the 4 drumsep models below (along with MDX23C model, and besides Demucs model by Imagoy) [metrics](#)
- MVSEP's SCNet 4 stem (kick, snare, toms, cymbals) out of following models, the best SDR for kick and similar to 6 stem below for toms - only -0.01 SDR difference)
- MVSEP's SCNet 5 stem (cymbals, hi-hat, kick, snare, toms)
- MVSEP's SCNet 6 stem model (ride, crash, hi-hat, kick, snare, toms) worse snare SDR

(newer) MDX23C 5 stem drumsep by jarredou

[Download](#). All SDR metrics are better than the previous 6 stem model below:

SDR: kick: 16.66, snare: 11.54, toms 12.34, hihat: 4.04, cymbals: 6.36 ([all metrics](#)).

Metric fullness for snare: 25.0361, bleedless for hh: 12.3470, log_wmse for snare: 13.8959
“it’s more on the fullness side than bleedless” - from all the metrics, only bleedless for snare is worse than in the previous model:

26.8420 vs 30.4149

“Quite cleaner than the previous [6 stem] one”, “a lot noisier than other drumpsep models, but that’s not necessarily a bad thing.”

Possible “UnpicklingError: “invalid load key, ‘\x0a’.”” issue in UVR if you use the old 6 stem yaml.

Maybe if we separate just snare with the old MDX23C model below from an already separated drums stem, and mix/invert to get the rest, then pass it through the new model, the bleed would be gone.

For comparison, [metrics](#) of the old 6 stem jarredou/Aufr33 MDX23C model

(which has cymbals divided into ride and crash which are not in the evaluation dataset):

SDR: kick: 14.55, snare: 9.79, toms: 10.64, hihat: 3.20, cymbals: 6.08

Metric fullness for snare: 25.0361, bleedless for hh: 10.2765, log_wmse for snare: 12.4258
The model was trained with a lightweight config to train on a subpar T4 GPU on free Colabs and 10 accounts. The metrics do not surpass exclusive drumsep MelRoFormer and SCNet models on MVSEP, but at least you can use this one locally.

“Depending on the quality tier of input source material, it can sometimes yield more accurate stem-to-stem separations than either MelRoFormer or SCNet-XL four-or-six stem DrumSep models. (...)

For example, I often find that MelRoFormer DrumSep can leave the upper frequency range of Snare hits and mis-assign them to the Cymbals stem. This is a common issue I have encountered with separating AUD recordings with MelRoFormer DrumSep.

MDX23c 5-stem drumsep is trained in such a way that it separates these snare remainders out of the Cymbals stem, which is extremely useful. " Dyslexicon"

(older) MDX23C 6 stem drumsep by jarredou/Aufr33

Use it on already separated [drums](#).

Download

https://github.com/jarredou/models/releases/tag/aufr33-jarredou_MDX23C_DrumSep_model_v0.1

Use on Colab:

<https://github.com/jarredou/Music-Source-Separation-Training-Colab-Inference/>

". Added on MVSEP and [uvronline](#) too.

(jarredou) "Drums Separation model trained by aufr33

(on my not-that-clean drums dataset)

Stems:

kick, snare, toms, hh, ride, crash

MVSEP dataset evaluation:

SDR: Kick: 14.55, snare: 9.79, toms: 10.64, hihat: 3.20, cymbals: 6.08, hihat & cymbals: 6.77

[More metrics](#)

To get potentially better results with the model "try using a small pitch shift up or down, like +/- 1 or 2 semitones, in the settings you use to extract the drum stem from the instrumental stem. (...) can sometimes help bring out the lows or highs if they seem weak." (CZ-84)

It can already be used, but training is not fully finished yet.

The config allows training on not so big GPUs [n_fft 2048 instead of 8096], it's open to anyone to resume/fine-tune it.

For now, it's struggling a bit to differentiate ride/hh/crash correctly, kick/snare/toms are more clean.

[“and has the usual issues with mdx23 models, but it’s an improvement over drumsep I think”
- Dry Paint Dealer Undr]

- If you got an error while using jarredou's Drumsep Colab (object is not subscriptable):
change to this on line 144 in inference.py:

```
if type(args.device_ids) != int:  
    model = nn.DataParallel(model, device_ids = args.device_ids)  
(thx DJ NUO)
```

It works in UVR too. All models should be located in the following folder:
Ultimate Vocal Remover\models\MDX_Net_Models
Don't forget about copying the config file to: model_data\mdx_c_configs.
Once the model is detected, select the config in a new window, and that's all.

The model achieved much better SDR on private jarredou's small evaluation dataset compared to the previous drumsep model by Inagoy which was based on a worse dataset and older Demucs 3 arch.

The dataset for further training is available in the drums section of [Repository of stems/multitracks](#) - you can potentially clean it further and/or expand the dataset so the results might be better after resuming the training from checkpoint. Using the current dataset, the SDR might stall for quite some amount of epochs or even decrease, but it usually increases later, so potentially training it further to 300-500-1000 epochs might be beneficial.

Attached config also includes necessary training parameters for training further using ZFTurbo [repo](#).

Current model metrics (not MVSEP evaluation dataset):

"Instr SDR kick: 18.4312
Instr SDR snare: 13.6083
Instr SDR toms: 13.2693
Instr SDR hh: 6.6887
Instr SDR ride: 5.3227
Instr SDR crash: 7.5152
SDR Avg: 10.8059" Aufr33

And if evaluation dataset hasn't changed since then, the old Drumsep SDR:

"kick : 13.9216
snare : 8.2344
toms : 5.4471
(I can't compare cymbals score as it's different stem types)" - jarredou

After initial jarredou's training in Colab, Aufr33 decided to train the model for additional 7 days, to at least above epoch 113 (perhaps around 150, it wasn't said precisely), while using the same config, but on a faster GPU (rented 2x RTX 4090).

Even epoch 5 trained on jarredou's dataset casually in slow and troublesome free Colab (which uses Tesla T4 15GB with performance of RTX 3050, but with more VRAM) with multiple Colab accounts and very light and fast training settings, already achieved better SDR than Drumsep using smaller dataset and older architecture. Colab epochs metrics:

“epoch 5:

Instr SDR kick: 13.9763
Instr SDR snare: 8.4376
Instr SDR toms: 6.7399
Instr SDR hh: 0.7277
Instr SDR ride: 0.8014
Instr SDR crash: 4.4053
SDR Avg: 5.8480

epoch 15:

Instr SDR kick: 15.3523
Instr SDR snare: 10.8604
Instr SDR toms: 10.3834
Instr SDR hh: 4.0184
Instr SDR ride: 2.7248
Instr SDR crash: 6.1663
SDR Avg: 8.2509”

Don't forget to use already well separated drums (e.g. from Mel-Roformer for premium users on x-minus or MVSEP Drums ensemble) from well separated instrumental as input for that model, or jarredou's MDX23 Colab fork v. 2.5 or also for all stems - MVSEP 4/+ ensemble (premium).

Purely for drums separation from even instrumentals, the model might not give good results, hence it needs separated drums first. It was trained just on percussion sounds and not vocals or anything else.

Also, e.g. the kick and toms might have a bit weird looking spectrograms. It's due to: “mdx23c subbands splitting + unfinished training, these artifacts are [normally] reduced/removed along [further] training.” [Examples](#)

Older [drumsep](#) by Inagoy

Demucs 3 model. Just remember to use drums in one stem (e.g. with demucs_ft) from already good sounding instrumental or ensemble on MVSEP or MDX23 v. 2.4 Colab first, as use it as input (both are better for instrumental in most cases than just Demucs 4 - you can use various settings for ensembles to get better instrumentals, the better drums, the better results from drumsep)

- [Fixed Colab](#)
- or [Kubinka Colab](#) (you can provide direct links there)

- Available on MVSEP.com (but you can use more intensive parameters in Colab for a bit better quality)

(Use these solutions instead of GitHub Colab as the model's GDrive link from OG GitHub Colab is currently deleted, so drumsep won't work correctly, unless you replace GDrive link with model to the .th model reupload:

<https://drive.google.com/file/d/1S79T3XIPFosbhXgVO8h3GeBJSu43Sk-O/view>)

- Windows installation - execute the following:

```
demucs --repo "PATH_TO_DrumSep_MODEL_FOLDER" -n modelo_final  
"INPUT_FILE_PATH" -o "OUTPUT_FOLDER_PATH"
```

- You can also use drumsep in UVR 5 GUI
(so beside using fixed Colab or in CML):

Go to UVR settings and open application directory.

Find the folder "models" and go to "demucs models" then "v3_v4"

Copy and paste both the [.th](#) and [.yaml](#) files, and it's good to go.

Be aware that stems will be labelled wrong in the GUI using drumsep.

It's much more sensitive to shifts than overlap, where above 0.6-0.7 it can become placebo.
Consider testing it with shifts 20.

But some people find using shifts 10 and overlap 0.99 better than shifts 20 and overlap 0.75.
Just be aware, that if you're willing to wait, you can further increase shifts to 20 if you want the best of both worlds.

Also, consider testing it with -6 semitones e.g. in UVR 5.6/+ , or with 31183Hz sample rate with changed (decreased) tempo and pitch.

-12 semitones from 44100Hz is 22050 and should be rather less usable in most cases, the same for tempo preservation, it should be off.

Be aware that sometimes it can “consistently put hi hats in snare stem” and can contain some artefacts, and results might not null with the source.

“From what I've tested (on drums already extracted with demucs4_ft from a live band recording from the output of the soundboard... so shitty sounding!), It is quite good at separating cymbals from shells, and kick from snare, but there are parts of kick or snare sounds that can go into the toms stem (...it's easy to fix manually in a DAW)”

"Ok I did test it.

- You're right, Drumsep is good if shifts are applied, this makes a HUGE difference, first time i did test it with 0 or 1 shift and results were meh. Shifts (from about 5/6/10 depending on source) clean it nicely.

Minuses: only 4 outputs. Not enough for a lot of drumtracks (but hey you can Regroove results, and this is what i will be doing probably from now) - It takes a long time with a lot of shifts, - it doesn't null with original tracks

- Regroove allows me more separations, especially when used multiple times, so as a producer it allows me to remove parts of kicks, parts of snares etc, noises etc. More deep control. Plus it nulls easily (it always adds the same space in front) so I can work more transparently.

But you're right, I will use drumsep in the Colab with a lot of shifts as a starting point in most cases now."

"It's trained with 7 hours of drum tracks that I made using sample-based drum software like Adictive Drums, trying to get as many different-sounding drums as I could. As everything was controlled with MIDI, I could export the isolated bodies: kick, snare, toms (all on one track), and cymbals (including hi-hat). So every dataset example is composed of kick, snare, toms, cymbals, and the mixture (the sum of all of them)." - said the author - Inagoy

From paid solutions for separating drums' sections there is mainly a paid [FactorSynth](#) and other alternatives are more problematic or less perfect.

Use free zero shot for separating single other instruments from e.g. others stem from Demucs or GSEP.

Moises.ai drumsep

(only for Pro)

- Kick, snare, toms, hi-hat, cymbals, other

It's not well documented on their promotional materials, but the option is available after dragging your input file on the site, and then under drums button.

FactorSynth

Since version 3 available in a form of plugin for most DAWs. Demo runs for 20 minutes at a time. Exporting and component editing are disabled.

Till v. 2 it was Ableton-only compatible add-on. And (probably) could be used on free Ableton Live.

Also, not for separating drums from a full mix, but for separating your already separated drums into further layers like kick, snare, transients, cymbals, etc. from Demucs or GSEP (the latter usually has better shakers and at least hi-hats when they're in fast tempo).

[till v2 demo version limit was 8 seconds and no limit for full version]" "it's amazing". It works the same way as Regroover VST (which may have some problems with creating a trial account). It's comparable or better quality (both better than zero shot for at least drums).

"Factorsynth has more granularity, but drumsep is easier to work with and gets less confused between toms and kicks."

There's a freeware prototype 0.4-0.1 versions from 2017 for Mac available to download:
<https://www.jjburred.com/software/factorsynth/proto.html>

Regroover

Regroover is only for 30 seconds chunks, and they require manual align due to phasing issues - additional silence is added in the beginning and ending.

"Get your 30-second drum clip, then drag and drop it into Regroover.

Make sure to de-select the Sync option, as it will time stretch it by default.

On the right-hand side, I recommend changing the split to 6 layers instead of 4, simply for flexibility.

Once it has processed that, you can choose export -> layers."

There was a report that probably newer versions might not be feasible for this task anymore.

In other words:

It's much more hassle to use it than drumsep, but it's very good "if you need particular sound and not about pattern etc.

1. separate drums from whole track (demucs)
2. Cut drum track into max 30 second cuts [regroover limits] and ideally cut right on transient, some space before kick helps,
2. You use regroover for the first time and for example try to separate to 4 tracks, just so overall separation.
3. Those separation sums exactly to that is given, sometimes it just need to be realigned few ms.
4. And if for example kick still has some not needed parts, you just regroove it once again.

If are looking overall fast and for patterns, drumsep. Regroover for painfull but precise job. Also in most cases hihats are trash, but snare's and kicks you often can find perfetclu usable ones. I'm not sure about metal but overall."

UnMixingStation

"Very, very old and almost impossible to [find](#), but the separations are 95% close to Regroover". The software is 13 years old, and their site is down, and the tool doesn't seem to be available to buy anywhere.

LarsNet

Adden on MVSep. [Colab](#). Source: <https://github.com/polimi-ispl/larsnet>

It separates previously separated drums into 5 stems: kick, snare, cymbals, toms, hihat.

It's worse than Drumsep as it uses Spleeter-like architecture, but "at least they have an extra output, so they separate hihats and cymbals.". Colab

"Baseline models don't seem better quality than drumsep, but the provided checkpoints are trained with only 22 epochs, it doesn't seem much. (and STEMGMD dataset was limited by the only 10 drumkits), so it could probably be better with better dataset & training"

Similar situation as with Drumsep - you should provide drums separated from e.g. Demucs model.

There's also Zynaptiq Unmix Drums, but it's not exactly a separation tool, but to "Boost Or Attenuate Drums In Mixed Music".

- For only kick and hi hat separation now free -

VirtualDJ 2023/Stems 2.0 (kick, hi-hat)

Probably using drums from Demucs 4 or GSEP first, will give better results but, it's not perfect. In many cases it may leave bleeding of snare a little bit, in both hi-hat and kick track. Sadly it sometimes confuses these elements of a mix.

"If you are not using it professionally, and do not use any professional equipment like a DJ controller, or a DJ mixer, then VirtualDJ is (now) FREE".

RipX DeepAudio (-||-) (6 stems [piano, guitar])

Popular tool. Decent results for specific drums' sections separation (but as for vocal/instrumental/4 stems separation, all the tools mentioned in at the very top of the document outperforms RipX, so use it only for specific drums' section separation only, at best using Demucs 4 or GSEP for drums stem).

"It can separate a file into a buncha things into a lot more types of instruments than just the basic 4 stems (with varying degrees of success ofc).

Might be a case that old cracked versions of RipX don't allow separating drums sections well, or just the opposite - check both the newest version and Hit'n'Mix RipX DeepAudio v5.2.6, but probably the latter doesn't support separating single drums yet.

It's basically UVR but with their custom models + SFX single stem
It's good for guitar, but not in all cases (possibly Demucs for 4 stems).
Piano and guitar models were added recently (somewhere in the January 2023)

- Hit 'n' Mix RipX DAW Pro 7 released. For GPU acceleration, min. requirement is 8GB VRAM and 10XX card or newer (mentioned by the official document are: 1070, 1080, 2070, 2080, 3070, 3080, 3090, 40XX). Additionally, for GPU acceleration to work, exactly Nvidia CUDA Toolkit v.11.0 is necessary. Occasionally during transition from some older versions, separation quality of harmonies can increase. Separation time with GPU acceleration can decrease from even 40 minutes on CPU to 2 minutes on decent GPU.

They say it uses Demucs.

We have reports about crashes, at least on certain audio files. There are various RipX versions uploaded on archive.org, maybe one will work, but some keys work only on versions from 2 and up.

Spectralayers 10

Received an update of an AI, and they no longer use Spleeter, but Demucs 4 (6s), and they now also good kick, snare, cymbals separation too. Good opinions so far. Compared to drumsep sometimes it's better, sometimes it's not. Versus MDX23 Colab V2, instrumentals sometimes sound much worse, so rather don't bother for instrumentals.

USS-Bytedance (any; esp. SFX)

<https://github.com/bytedance/uss>

(COMMAND: "conda install -c intel icc_rt" SOLVES the LLVM ERROR)

You provide e.g. a sample of any instrument or SFX, and the AI separates it solo from a song or movie fragment you choose to separate.

It works in mono. You need to process right and left channel separately.

Update 29.04.25 (Python No such file or directory fix; thx epiphery)

https://colab.research.google.com/drive/1rlf0YJt7cwdxT_pQlgobJNuX3fANyYmx?usp=sharing

(old) ByteDance USS with Colab by jazzpear94

https://colab.research.google.com/drive/1IRjlsqeBhO9B3dvW4jSWanjFLd6tuEO9?usp=share_link

(old) Probably mirror (fixed March 2024):

<https://colab.research.google.com/drive/1f2qUITs5RR6Fr3MKfQeYaaj9ciTz93B2>

errors out with:

"sed: can't read /usr/local/lib/python3.10/dist-packages/uss/config.py: No such file or directory" (2025)

It works (much) better than zero-shot (not only "user-friendly wise").

Better results, and It divides them into many [categories](#).

Great for isolating SFX', worse for vocals than current vocal models. Even providing acapella didn't give better results than current instrumental models. It just serves well for other purposes.

"Queries [so exemplary samples] for ByteDance USS taken from the DNR dataset. Just download and put these on your drive to use them in the Colab as queries [as similarly sounding sounds from your songs to separate]."

<https://www.dropbox.com/sh/fel3hung4eb83rs/AAA1WoK3d85W4S4N5HObxhQGa?dl=0>

Also, grab some crowd samples from here:

<https://youtu.be/-FLgShtdxQ8>

<https://youtu.be/IKB3Qiglyro>

<https://youtu.be/Hheg88LKVDs>

Q&A by Bas Curtis and jazzpear

Q: What is the difference between running with and without the usage of reference query audio?

A: Query audio lets you input audio for it to reference and extract similar songs based upon (like zeroshot but way better) whereas without a query auto splits many stems of all kinds without needing to feed it a query.

Q: Let's say there is this annoying flute you wanna get rid off...
and keep the vocals only....

You feed a snippet of the flute as reference, so it tries to ditch it from the input?

A: Quite the reverse. It extracts the flute only which ig you could use to invert and get rid of it

Zero Shot (any sample; esp. instruments)

https://github.com/RetroCirce/Zero_Shot_Audio_Source_Separation

(as [USS Bytedance](#) came out now, zero shot can be regarded as obsolete now, although zero-shot might be better for single instruments than for SFX)

You provide e.g. sample of any trumpet or any other instrument, and AI returns it from a song you choose to separate.

[Guide and troubleshooting](#) for local installation (get Discord invitation in footer first if necessary).

Google Colab [troubleshooting](#) and [notebook](#) (though it may not work at times when GDrive link resources are out of download limit, also it returns some torch issues after Colab updates in 2023).

Check out also this Colab alternative:

https://replicate.com/retrocirce/zero_shot_audio_source_separation

It's faster (mono input required).

Also available on <https://mvsep.com/> in a form of 4 stems without custom queries, and it's not better than Demucs in this form.

"Zero shot isn't meant to be used as a general model, that's why it accelerates on a specific class of sounds with some limitations in mind.... It mostly works the best when samples match the original input mixture, of course there are limitations"

"You don't have to train any fancy models to get decent results [...] And it's good at not destroying music". But it usually leaves some vocal bleeding, so process the result using MDX to get rid of these low volume vocals. Zero-shot is also capable of removing crowd from recordings pretty well.

As for drums separation, like for snares, it's not so good as drumsep/FactorSynth/RipX, and it has cutoff.

"I did zero shot tests a week or two ago, and it was killing it, pulling harmonica down to -40dB, synth lines gone, guitars, anything. And the input sources were literally a few seconds of audio.

I've been pulling out whole synths and whistles and all sorts.

Knocks the wind model into the wind, zero shot with the right sample to form the model backbone works really well

The key is to give it about 10 seconds of a sample with a lot of variation, full scales, that kinda thing"

Dango.ai

Custom stem separation feature (paid, 10 seconds for free)

Expensive

Special method of separation by viperx (ACERVO DOS PLAYBACK) edited by CyberWaifu

Process music with Demucs to get drums and bass.

Process music with MDX to get vocals.

Separate left and right channels of vocals.

Process vocal channels through Zero-Shot with a noise sample from that channel.

Phase invert Zero-Shot's output to the channel to remove the noise.

Join the channels back together to get processed vocals.

Invert the processed vocals to music to get the instrumental.

Separate left and right channels of instrumental.

Process instrumental channels through Zero-Shot with a noise sample from that channel.

Phase invert Zero-Shot's output to the channel to remove the noise.

Join the channels back together to get processed instrumental.

Process instrumental with Demucs to get other.

Combine other with drums and bass to get better instrumental.

So it sounds like Zero-Shot is being used for noise removal.

As for how Zero-Shot and the noise sample works...

AudioSep

"I decided to try AudioSep: <https://github.com/Audio-AGI/AudioSep> on MultiSong Dataset.

I used prompt 'vocals'. I was sure it would be bad, but I didn't think it's so bad.

https://mvsep.com/quality_checker/entry/8408

I also tried it on the Guitar dataset - it's even worse - negative SDR. Maybe I'm doing something wrong. But I tried the example with cat from the demo page, and it worked the same as in there. So I think I have no errors."

```
sdr: 0.33
si_sdr: -2.39
l1_freq: 17.62
log_wmse: 6.72
aura_stft: 3.66
aura_mrstft: 5.55
bleedless: 9.29
fullness: 16.58
```

Colab on GH probably gives unpickling issue. You might be able to fix it by executing:

```
!pip install torch==2.5.0
```

After you execute all the installation-related cells.

Since then, probably something more about dependencies is also needed, like it's coded now in the inference Colab.

Medley Vox (different vocalists)

Use already separated [vocals](#) as input (e.g. by Roformers, vox_ft or MDX23C fullband a.k.a. 1648 in UVR or 1666 on MVSEP).

Local installation video tutorial by Bas Curtiz:

<https://youtu.be/VbM4qp0VP8>

(NVIDIA GPU acceleration supported, or perhaps CPU - might be slow)

Cyrus version of MedleyVox Colab with chunking introduced, so you don't need to do chunking manually:

https://colab.research.google.com/drive/10x8mkZmpqiu-oKAd8oBv_GSnZNKfa8r2?usp=sharing (07.02.25 fork with fairseq fix and GDrive integration)

Currently, we have a duet/unison model 238 (default in Colab),
and main/rest 138 to uncomment in Colab.

Recommended model is located in vocals 238 folder (non ISR-net one).

While:

"The ISR_net is basically just a different type of model that attempts to make audio super resolution and then separate it. I only trained it because that's what the paper's author did, but it gives worse results than just the normal fine-tuned."

MedleyVox is also available on MVSEP, but it has more bleeding and "doesn't work as well as the Colab iteration with duets". (Isling/Ryanz)

The "duet/unison model 238" will be used by default.

``and main/rest 138 to uncomment in Colab`` if you need it.

Then go to the first cell again. To "uncomment" means to delete the "#" from the beginning of the line before the "!wget" so the line will be used to download the model files.

Do it for both pth and json lines

(you might be asked whether to replace existing pth and json files by the alternative model you just downloaded in the place of the previous one)

``Recommended model is located in vocals 238 folder (non ISR-net one).``

That's the model used in the Colab by default. You can ignore that information. It's for users using the MV on their own machine.

The output for 238 model is 24kHz sample rate (so 12kHz model in Spek). You might want to upscale the results using e.g. [AudioSR](#) or maybe even Lew's vocal enhancer location further below the linked section.

The output is mono.

You might want to create a "fake stereo" as input by copying the same channel over the two, then do the same with another channel, and then create the stereo result from both channels processed separately in dual mono with MV.

The AI will create a downmix from both input channels instead of processing channels separately.

Be aware that "dual mono processing with AI can often create incoherencies in stereo image (like the voice will be recognized in some part only in left channel and not the other, as they are processed independently)" jarredou

"The demos sound quite good (separating different voices, including harmonies or background [backing] vocals)"

It's for already separated or original acapellas.

The model is trained by Cyrus. The problem is, it was trained with 12kHz cutoff... "audiosr does almost perfect job [with upscaling it] already, but the hugging page doesn't work with full songs, it runs out of memory pretty fast".

It was possible at some point that later stages of the training, looking like over fitting were responsible for higher frequency output.

It's sometimes already better than BVE models, and the model has already similar to demo results on their site.

Sadly, the training code is extremely messy and broken, but a fork by Cyrus with instructions is planned, with releasing datasets including the one behind geo-lock. Datasets are huge and heavy.

Original repo (Vinctekan fixed it - the video at the top contains it)

<https://github.com/jeonchangbin49/medleyvox>

Outdated

<https://colab.research.google.com/drive/1StFd0QVZcv3Kn4V-DXeppMk8Zcbr5u5s?usp=sharing> (pip issues fixed 29.08.24, defunct as of 06.02.25)

(outdated instructions, current Colab explains everything)

"Run the 1st cell, upload song to folder infer_file, run the 2nd cell, get results from folder results = profit"

Further explanations how to use the Colab:

``Run the 1st cell``

So press the first "play" button then you load the Colab

``upload song to folder infer_file``

Looks like the folder for the input file has changed from infer_file to input in newer Colabs.

So, once you started the first cell, and it finished, open Colab file manager (folder icon on the left) and go to /content/MedleyVox/input\

Now paste your song there and wait till it's done.

``run the 2nd cell``

So the next play button below the first one once you scroll down a bit. Now it will start separation

``get results from folder results``

Go to file manager again and find /content/MedleyVox/results

right-click on the result file and download it. Wait till it's done.

``Currently, we have a duet/unison model 238 (default in Colab)``

So you don't have to change anything in the Colab to separate using it.

Old info

https://media.discordapp.net/attachments/900904142669754399/1050444866464784384/Screenshot_81.jpg (dead)

Colab old

<https://colab.research.google.com/drive/17G3BPOPBcwQdXwFiJGo0pKrz-kZ4SdU>

Older Colab

<https://colab.research.google.com/drive/1EHJFBSDd5QJH1FQV7z0pbDRvz8yXQvhk>

(The same one, but here you need to change the .ckpt, .json and .pth files there from Cyrus [more details in the video above].)

About other services:

Check [this](#) chart by Bas Curtiz to check what AIs use various (also online) services, plus their pricing.

At this point everything mentioned above this link for at least instrumentals, vocals, 4-6 stems is better than below, (with exceptions for some single stems described at the top) commonly known services:

Spleeter

and its implementation in:

Izotope RX-8/9/10

which just uses 22kHz models instead of 16kHz in the original Spleeter. There is no point in using these anymore. The same goes to most AIs described below (or only for specific stems):

voiceremover.org, lalal.ai,
phonicmind
melody.ml
RipX, Demix,
ByteDance Ripple/CapCut
[beatstorapon](#)

For reference, you can check a [comparison](#) chart on MVSEP.com, or results of [demixing challenge](#) from Sony (kimberley_jensen there is 9.7 MDX-UVR model for vocals - 2nd best on the time)

and watch [this](#) comparison.

To hear 4 stems models comparison samples you can watch [this](#) video comparison (December 2022).

It all also refers to new:

real-time

AI separation tools like

Serato

and

Stems 2.0

tensorflow model (which can be found in newer Virtual DJ 2023 versions, now free for home users - better than Serato and Spleeter implementations) - they do not perform better than the best offline solutions at the very top of the document. But “Esp. since it's on-the-fly [...] results are more than decent (compared to others).”

Acon Digital Remix
(Vocals, Piano, Bass, Drums, and Other)

“Just listened to the demo, not great [as for realtime] but still”

Others

FL Studio (Demucs)

It's actually not realtime. It takes some time to process tracks first (hence maybe it's the best out of the three).

It's Demucs 4, but maybe not ft model and/or with low parameters applied or/and it's their own model.

"Nothing spectacular, but not bad."

- FL Studio bleeds beats, just like Demucs 4 FT

- FL Studio sounds worse than Demucs 4 FT

- Ripple clearly wins"

djay Pro 5.x

"very good realtime stems with low CPU" Allegedly "faster and better than Demucs, similar" although "They are not realtime, they are buffered and cached." but it's very fast anyway. It uses AudioShake. It can be better for instrumentals than UVR at times.

Neutone VST

Has Demucs model to use in realtime in a DAW

(it uses light "retrained, smaller version" version of Demucs_mmi)

<https://neutone.space/>

<https://neutone.space/models/1a36cd599cd0c44ec7ccb63e77fe8efc/>

It doesn't use GPU, and it's configured to be fast with very low parameters, also the model is not the best on its own. It doesn't give decent results, so it's better to stick to other real-time alternatives. It won't work correctly on low-end CPU, breaking audio in the middle and giving inconsistent audio stream with breaks.

Peel Stems

<https://products.zplane.de/products/peelstems/>

VST for real time source separation (probably same models like in MPC stems)

<https://www.youtube.com/watch?v=0Js5bWQWY7M>

- Service rebranded to

Fadr.com from SongtoStems.com

is just Demucs 4 HT, but paid.

"My assumption, Fadr uses Gain Normalize [for instrumentals] was right [...].

Demucs 4 HT seems to get a cleaner result. The rest = practically identical." And someone even said that vocals in VirtualDJ with Stems 2.0 had less artifacts on vocals.

Apple Music Sing

"I heard a few snippets, and what stood out is, whether intentional or not, the vocals remained in the background just enough to actually hear them. Now that could be great for Karaoke, so u have a kind of lead to go on." but as for just instrumentals, it's bad.

Voxless

VST "uses AI to separate vocals and instrumental in real time. Now it is designed to be used in a DAW, but you can also run it in soundsource [on Mac, or probably SAVIHost (VST2/3) or Equalizer APO (VST3) or JBridge on Windows] so you can use it on your system audio live. It has low latency and doesn't use CPU a lot. The software has a very simple interface, just two knobs to increase/ decrease the instrumental or vocals or a mute/solo button for each. As for the quality it sounds like the first ever days of audio separation with AI like Demucs v1 or Spleeter in 2019 - 2020 but a little worse somehow, since it is very low latency not CPU heavy, but it does the job. Voxless has a trial of 7 days if you wanna check it but the license costs 100\$ which I think is quite a lot for a software that separates vocals with barely first gen quality." midol

Ozone 11 Master Rebalance

"I select vocals and have them dialed down to max using an EQ inside of it (may sound complicated, so you gotta watch a tutorial to see how ozone works). However, the results were far from voxless quality. It leaves so much bleed and whenever vocals are quite loud you can barely hear anything from the way it's fighting it, so it sounds like a complete mess. Both from the master rebalance and the main AI interface" midol

(x) BL-Rebalance

"The most important thing which is the separation quality, is horrible unfortunately, dialing the vocals all the way down to -120db the max, barely picks up vocals to cancel, the song sounds like it's just playing normally with vocals being suppressed in a very horrible way, it's muddy, and it leaves a lot of bleed, also again, when vocals are quite loud, you barely hear anything else because it's fighting hard." midol

algoriddim djay

App for Windows, Android, Mac.

Judging by strings in stemseparation.dll, they seem to use "bytesep" which is a package name of this repository: [bytedance/music_source_separation](https://github.com/bytedance/music_source_separation).

Music to MIDI transcribers/converters

<https://github.com/magenta/mt3>

https://colab.research.google.com/github/magenta/mt3/blob/main/mt3/colab/music_transcription_with_transformers.ipynb

<https://basicpitch.spotify.com/>

“Tried Basic-Pitch and It is way worse than MT3 as It produces midi tracks without an identifier.”

Good results for piano:

https://colab.research.google.com/notebooks/magenta/onsets_frames_transcription/onsets_frames_transcription.ipynb

If you have notes:
musescore

[transkun](#) transcriber

“it's the most accurate piano transcription algorithm ever trained and is unequalled in accuracy and absolute indifference to literally *any audio quality* as long as the piano being transcribed is at A440 it'll spit out a 95 percent accurate transcription from virtually any recording no matter how absolute garbage it is”

[Piano2Notes](#)

(notes and midi output, paid, 30 seconds for free, very good results)

Harmonic mixing (find the song key)

“Since mixed in key change from 10 to 11 the software has several failures especially when overwriting the file name and an error that base 84 error, and you are left without the analysis of the file thing. Which is essential when doing remixes and having a clarity of the tone and bpm. Someone knows of an alternative that does not make a mistake”

https://www.reddit.com/r/DJs/comments/n5byah/key_detection_comparison_mixed_in_key_10_vs_85/

Older separation services

Audioshake

Paid, \$16 per wav stem, 2 or 5 stems (6? (guitar and piano) or 4 stems for preview (Indie creators)

Better piano model than GSEP.

"gsep piano model is very clean but sometimes fails in bigger mix, when there are a lot of instruments"

And also guitar stem

Instead of Audioshake you can use:

- myxt.com (also paid, 3 stem model, prob. 16kHz cutoff which Audioshake normally doesn't have. No other stem. Results, maybe slightly better than Demucs)
 - Algoriddim djay pro
 - Neural Mix Pro (part of Algoriddim, also uses Audioshake), but it's only for MacOS
- LANDR Stems (cheaper, also uses Audioshake; plugin, probably doesn't work locally, free access won't give you access to stems; "LANDR Stems is only included in Studio Standard + Studio Pro" it's not included in trial; SDR: [1](#) | [2](#))
 - <https://twoshot.app/model/289>

Audioshake is suspected that it is just MDX with expanded dataset, but there's no evidence at the moment. Comparing to UVR/MDX-UVR NET 1 model, vocal stem is 9.793 vs 9.702 in free MDX-UVR, so they're close as for vocals.

Their researcher said they were training UMXHQ model at this period of time of 2020 Demixing Challenge.

Free Demucs 3 has a much better SDR for drums and bass than Audioshake, however the SDR for vocals and others is worse.

It accepts only non-copyrighted music for separations, but you can slow it down to circumvent it (some music like K-Pop BTS is not detected) but changing speed to 110% yields better results, even vs reversing the track.

Upload limit is one minute, so theoretically you can cut and merge chunks, but AS will fade out each chunk, so you need to find specific overhead to begin every next chunk with, to merge chunks seamlessly (I don't remember if it solves the problem of AS watermark, though).

Then, you can download preview (chunk) for free using similar method like described in allflac section (Chrome Dev Tools -> Network -> Set filter to amazon) but result file is unfortunately only 128kbps mp3.

They are now limiting how many audio files you can upload to preview, but that can easily be mitigated by just using a temporary email provider or adding "+1" or "+2" or "." to your gmail

address, so you will still receive your email e.g. y.o.u.r.m.a.i.l@gmail.com is the same for Google as yourmail@gmail.com.

You can also ping Smudge, Baul Plart or Bas Curtiz in #request-separation to isolate some song to make this all easily just for you (edit. 09.02.2023 - at least the Bas' tool stopped working, so the rest like AS Tool might be dead too - at least in terms of API access, not sure).

Lalal.ai

7 stem

Acoustic and electric guitar models, piano, bass, drums and vocal with instrumental (for 2 stem UVR/MDX should do the job better)

Online service with 10 minutes/50MB per file limitation per free user.

Now they have some voc/inst models sounding like some ensemble of public Roformers, but still not as good, but close. Some specific models are worth trying out, e.g. lead guitars - the model got better by the time or piano model.

Older notes:

"I love Demucs 3, although for some specific songs (with a lot of percussion and loops) I still find lalal better.

Demucs is great at keeping punchy drums, for example hip-hop, rap, house etc songs"
"lalal is[n't] worth it anymore, most of their models like strings or synths are crap and don't work at all" ~beccruily

How to... abuse it. Doesn't always work for everyone, and sometimes you'll receive only 19 seconds snippets.

Go to the signin/register screen and use a temp email from <https://tempail.com/>

When you are in, make sure you use the settings with a P icon, P meaning Pheonix, which seems to be some hybrid mvsep lalal shit they made

I'd recommend making the processesing level normal, although you can play around with the settings to see what sounds better

They will later process it and since lalal has shorter queues, you get them faster. It took me like 10 seconds to get a preview for a song and 20 seconds for full which is wild.

You will get a sample and if you like it, you can submit it and get your stems!"

You can also use dots in Gmail addresses, instead of +1 (and more) at the end, which is unsupported in lalal. You'll receive your email with dots in its username anyway, and it will be treated as a separate email by their system.

Their app uploads input files to separate on external servers.

DeMIX Pro V3

Paid, 6 stem model, trial

Official site:

<https://www.audiosourcere.com/demix-pro-audio-separation-software/>

https://www.demixer.com/?utm_source=audiosourcere&utm_medium=pop&utm_campaign=exit&utm_term=asre-exit-pop

paid 33\$/month, or x10 for year, or x2,5 permanent license, 7 days trial available

<https://www.audiosourcere.com/demix-pro-audio-separation-software/>

Vocal, Lead Vocal, Drum, Bass & Electric Guitar

<https://www.demixer.com/> has the same models implemented, though they don't currently even describe that guitar model is available, but when you log in, it's [there](#). Guitar might be a bit worse than RipX (not confirmed)

"audioshake [had] the best guitar model [at some point] (its combined [paid only]), second place is deemix pro (electric guitar)"

"Demix launched a new v4 beta, and it can now process songs locally + new piano and strings models

the piano model is not bad at all, it sounds a bit thin/weak, but it detects almost all notes hadn't found good songs to test the strings model yet, but it might be good too"

[Hit'n'Mix RipX DeepAudio](#)

Moises.ai

<https://moises.ai/>

Not really a good models before introducing BS-Roformer ones, no previews for premium features.

You can use apk when it allows previewing for free without downloading, but isling found some workaround googling for "moises premium free apk".

Some information here might be outdated.

"also has a guitar and a b.v. model, and a new strings model, but it's not that good, in my opinion it is not worth buying a premium account.

4-STEM model is something like demucs v2 or demixer.

B.V. model is worse than the old UVR b.v..

GUITAR model is not really good, it's probably MDX, it has a weird noise, and it tries to take the "guitar" where is not at all. It takes acoustic and electric guitar together.

PIANO model is just splitter, maybe better at some songs.

STRINGS model is interesting, It's good for songs with orchestra, but still not that clean

Their service is very interesting, and the appearance of their site is clear and simple, but the models have better competitors." thx, sahlofolina.

Byte Dance
available on <https://mvsep.com/>

"This algorithm took second place in the vocals category on Leaderboard A in the Sony Music Demixing Challenge. It's trained only on the MUSDB18HQ data and has potential in the future if more training data is added.

Quality metrics are available here (SDR evaluated by his authorship non-aircrowd method):

<https://mvsep.com/quality.php>

Demos for Byte Dance: <https://mvsep.com/demo.php?algo=16> "
(8.08 SDR aircrowd for vocal)

MDX-UVR SDR vocal models (kimberley_jensen a.k.a. KimberleyJSON) were evaluated by the same dataset as ByteDance above (aircrowd):

https://www.aircrowd.com/challenges/music-demixing-challenge-ismir-2021/leaderboards?challenge_round_id=886&challenge_leaderboard_extra_id=869&post_challenge=true

<https://discord.com/channels/70857973558358363/887455924845944873/910677893489770536>

and presumably the same goes to GSEP and their very first vocal model (10 SDR) since their chart showed the same ByteDance SDR score like in aircrowd.

____UVR settings for ensemble (section deprecated, see the section above)____

Ensemble can provide different results from one current main model, but not especially better in all cases, so it's also a matter of taste and conscious evaluation.

- Aggressiveness shouldn't be set to more than 0.1
(also check 0.01)
- high_end_process: bypass (official recommendation) or mirroring 2 (in some cases)

- In most cases, you shouldn't use more than 4 models to not decrease the quality (developer recommendation)

Don't use postprocessing in HV Colab for ensemble (doesn't work).

Other recommended models for ensemble:

HP2-4BAND-3090_4band_arch-500m_1.pth,
HP2-4BAND-3090_4band_arch-500m_2.pth
(+new 3 band?)

as they currently the best (15.08.21) but feel free to experiment with more (I also used old MGM beta 1 and 2 with two above, some people used also vocal models as well, and later there was also HP2-MAIN-MSB2-3BAND-3090_arch-500m model released, which gives good results solo).

____Good UVR accapella models_____

In general, it's better to use MDX-UVR models for clean acappellas, but for UVR, these are going to be your best bet:

- Vocal_HP_4BAND_3090 - This model will come out with less instrumental bleed.
- Vocal_HP_4BAND_3090_AGG - This is a more aggressive version of the vocal model above.

"If you wanna removes the vocals but keeping the backing vocals, you can use the latest BV model"

HP-KAROKEE-MSB2-3BAND-3090.pth
(HV)

For clean vocal, you can also use ensemble with following models:

<https://cdn.discordapp.com/attachments/767947630403387393/897512785536241735/unknow.png>
(REUim2005)

____How to remove artefacts from an inverted acapella?_____

This section is old, and "cleaning inverts" in [current models](#) section can provide more up-to date solutions.

0) Currently, GSEP is said to be the best in cleaning inverts. But at least for vocal you can use some MDX model like Kim, or even better MDX23 from MVSEP beta.

1) by charm

(rather outdated) Use Vocal_HP_4BAND_3090_arch-124m.pth at 0.5 aggressiveness, tta enabled

then use any model u like with Vocal_HP_4BAND_3090_arch-124m.pth instrumental results to filter out any vocals that weren't detected as vocals with
Vocal_HP_4BAND_3090_arch-124m.pth model

combine the two results

then use model ensemble with whatever models u like (i used HP2 4BAND 1 and 2)

drag both vocal hp 4band+another model and ensemble results into audacity

use the amplify effect on both tracks and set it to -6.03

render

then use StackedMGM_MM_v4_1band_arch-default.pth

tbh vocal models even at 0 aggressiveness really help inverts
Or 0.3

I mostly use acapellas for mashups and remixes, so the little bit of bleed i get at 0.0 aggressiveness is fine

drums-4BAND-3090_4band.pth
0.5 optionally (less metallic sound)

2)

Utagoe (English version with guide and error messages translated by Anjok) - if the invert isn't good, then try utagoe, but it's not the best.

Settings for Utagoe by HV:

"if your tracks don't invert perfectly" (even when aligned)

<https://imgur.com/a/ZC14xIE>

"if it's perfectly inverting":

<https://imgur.com/a/Qb4pKeX>

Some other settings:

<https://imgur.com/a/fvQwbMO>

"It has a weird issue sometimes tho, even when everything is perfectly aligned and inverts perfectly, utagoe misses some places, and it won't insert for a second or so"

by Mixmasher00

"There is no actually settings depending on songs, but that is what I use, which is the default one.

<https://imgur.com/a/kSDrTAB>

Going higher than 1.3 [of extractable level] imo won't do good at cleaning. Additional tip too, if you want to do just an "invert" and "keep the original vocal volume" just choose "by waveform".

I have been using Utagoe for inversions recently because it keeps the original volume of the vocals, and then I ran it on UVR or MDX. If the chunks are soft, I prefer using UVR but if there are chunks are that heavy like drums, I'd use MDX.

Also, I find it better to clean an invert via UVR or MDX than Utagoe because it's better and cleaner without destroying the vocals"

- "When using Utagoe, or UVR5, for aligning inputs, and inverting them, I get this really strange crackling noise, that not even doing a vocal separation with AI later can get rid of. Anyone know what could possibly be causing this?

So, I actually found a solution to this, for anyone running into a similar issue.

With inversions like this, you're already gonna have to use AI to get rid of the left over noise, since it's not gonna be a perfect inversion. So, the solution isn't to get a perfect one, it's simply to get rid of the noise that the AI cannot recognize, right?

With this particular inversion, I originally was using really compressed MP3s from around 2007 for the instrumentals, because the lossless versions of the instrumentals were lost media, up until a few months ago.

I thought it was odd, because i don't remember this noise being an issue with the MP3s, and that's when it hit me, MP3s cut off the noise, with compression, and added just a small bit more of that noise you get with imperfect inversions.

So I converted the lossless instrumentals to an MP3 with Foobar, and it was better, but still had that damned drum crackling! So i kept trying. I used OPUS, OGG, different bitrates of MP3, even AAC.

I have found that OPUS is the best at removing the drum overlap, I cannot hear any in fact, with OPUS.

So, my final guide is,

If you are getting crackling/crackling/overlap on drum hits in your inversions, then:

1. Convert the instrumental to OPUS (128 Kbps) with Foobar2000.
2. Use a software like Audacity to amplify it to a peak amplitude zero DB (since apparently OPUS auto declips to floating levels?)

3. Export it as a WAV at the original sample rate (since OPUS only supports 48 kHz, I actually tried resampling the instrumental, and original to 48 kHz before converting to OPUS, but found that results in a WORST output.)
4. Do your inversion (hopefully in Utagoe)
5. Use whichever vocal model you like the output for best for cleanup." - sausum

PS. Once, I've runned into similar issue. And I fixed it, actually similarly. I think I was trying to invert with mp3 VBR, and the other file was lossless, so I converted it to the same codec and bitrate/V preset.

Yes, it wasn't perfect, but better.

I wonder if simply applying cutoff at 20kHz wouldn't be a better solution. That's what Opus more or less does, plus upsampling to 48kHz.

- Despite the fact that separation is in 32 bit float, align inputs option in UVR uses something lower internally, hence the clipping may occur.
- As better alternative to Utagoe and UVR's Align feature, you can use paid Auto Align Post 2 (maybe even cheaper MAutoAlign).

Sources of FLACs for the best quality for separation process

Introduction

- Don't use YouTube or mp3 as input files for separation. Compression decreases the quality of the output
- If you're forced to use YT, download audio, preferably as Opus if it's available for your video, and it exceeds 16kHz on spectrogram (AAC might be better up to 16kHz).
- To enhance results from YT by combining Opus and AAC [read](#)
- If you want to verify if your input file is really lossless:
<https://fakinthefunk.net/en> (sometimes streaming services share bad/lossy versions)
- If you want to check the real bit depth of the file, check:
<https://www.stillwellaudio.com/plugins/bitter/>
- For output files, you can untick exporting as mp3 in [Colabs](#) (export your separations as WAV/FLAC)
- To upload your output file losslessly on YT [read](#)

Various versions of the same song

Sometimes the same track you may try to isolate can exist in few versions: e.g.

0) album version

- on streaming services - sometimes both explicit and non-explicit album versions are available, plus sometimes both in 44kHz 16-24 bit or 48-192kHz - they might give a bit of different results (if your old player app struggles with playing these files, download the latest MKVToolNix, drag and drop the file and begin multiplexing. It doesn't reencode/recompress the audio stream)

- on CD - sometimes these are two different masters - if total time and track gain of lossless files scanned by F2K is different by e.g. 1dB or iLufs reading is different, it's a different master.

- ~ Sometimes recent masters of older music are louder on CDs than on streaming services providing fewer dynamics, and in most cases such CD should be worse for AI separation when mastered to -9 ilufs vs -14 ilufs for streaming, although it can be totally opposite for some releases too

- ~ Regional CD version - certain albums in the past used to have different releases for some countries, e.g. Japan, different track order, even slightly different mastering

- on DSD - if available, they are different masters and might be worth to check for separation too

- on SACD - -||-

- on vinyl (so-called "vinyl rip") - might give you a bit of different results for problematic tracks

- on DVD-Audio (sometimes also 2.0 releases) if AC3 was used, it can be lossy - used bit depth or sample rate might depend on a release and it can be a different master in separation - usually vinyls are different masters with bass more/mostly in mono

- 1) single version - in the old days, single versions contained official instrumentals or accapella which not always invert with original mixture to get instrumental if it wasn't available (but if it inverts at least partially it might give you better result - always try lossless files - lossy might not invert well)

- on CD - sometimes contain different track list than on vinyl, extra tracks, remixes, etc. (rarely available on streamings in this form now) - always refer to Discogs to find all releases of your interest

- on vinyl - -||- (won't invert correctly due to constant playback speed fluctuations)

- 2) deluxe edition/reissue/remastered (sometimes separated instrumentals from remastered versions can be crispier than leaked multitracks which are rarely even mastered; also, different remasters might be available on streaming platforms or fan-made ones on YT or on the internet)

- 3) Video released for the song - although lossy, it can be completely different mix or master giving different results for separation

- 4) Official remix - sometimes it might be easier to separate vocals from such version

- 5) Leaks of earlier version of the song (might have different mixing, lyrics, even instrumental)

- YT (often a subject to be taken down), fan-made Discord servers, internet

- 6) Leaks of multitracks or stems of the song - usually it's a different master than the final song, but you might experiment with Matchering and using it mixdown without vocals a target, and well sounding separation as input for Matchering to make it more similar to the final song

7) Leak of instrumental/vocals - it can be lossy or slightly different from the final song, e.g. close to final stage or sometimes might have even dry vocals without any effects

Surround versions

8) 5.1 - e.g. DTS on DVD Audio/Blu-ray or SACD (you can search Discogz to look for multichannel versions released on disks, e.g. whole DTS Entertainment label)

9) 360 Reality or Atmos (7.1 or more) - e.g. on Tidal, Apple Music (how to download is described below).

Sometimes in surround releases, vocals can be there simply in the center channel, but it's not always the case - still, it can be a better source, e.g. when you manipulate with volume of specific channels, or for vocals - when you get only center channel with very little instrumentalization which may turn out easier to separate by AI (for instrumental you might possibly invert result of vocal model and center channel to receive the remain instrumentalization in center).

"For me, I convert left and right together then center alone then LS+Rs+LFE together then I have 3 audio files process them then remix into 5.1 again.

The 2 are in stereo and one mono which is center:..

- killdubo

"I do the same, except that I don't process LFE. Only the other 5." -

- santilli_ /Michael

E.g. "With the Dolby Atmos release of Random Access Memories, some vocals and instrumentals can be separated almost like stems"

Or alternatively, you can simply downmix everything to stereo and then separate (just to check the outcome vs regular 2.0 versions).

Tape

10) Cassette tape - wouldn't recommend it as a source (maybe unless it sounds superb, or you have a great deck at disposal to rip the recording) - even though the cassette tapes might be still released occasionally, contemporary music usually sound worse on them than it used to in the past, and potentially to compensate for it, they might contain different masters (cassette tape won't invert due to speed fluctuations either)

11) Reel to reel tape (better quality, mostly old music, also, usually in the past, the base medium for archival original stems of the recording before final mix and master, but degrading along the time, sometimes can change the sound after some time even when in the period of the song production)

General

As for good quality music on streaming services you can get FLAC 16 bit and 24 bit on Qobuz/Amazon) or on Tidal (now also up to 24 bit FLACs for Master quality - formerly MQA 24 bit (in the past most of Max (formerly Master) quality on Tidal was 16 bit MQA, while High (formerly Hi-Fi) is and was always FLAC 16 bit; MQA was lossy (but less than all other formats), but 24 bit MQA file could have given better results than 16 bit FLAC). MQA was gradually transitioned to FLAC on Tidal, but seems like old uploads in 24 bit MQA are not 24 bit anymore but just 16 bit FLACs, so you might want to use Qobuz to find some of these 24 bit files if they aren't available on Tidal like they used to be.

Most importantly -

Feel free to experiment with different versions and find the best result with a specific version of your song, although 24 bit FLAC should be the best (although not everyone might notice the difference).

If you have seemingly the same FLAC Audio CD rip from before streaming services times (~<2013), it can happen that a lossless file taken from a streaming service may be slightly different in most cases (same length but slight changes in Spek across the whole track which normally don't exist when comparing FLACs from various streaming services which have the same Audio MD5 checksums - also sometimes track finishes in slightly different place). Sometimes it can sound better, sometimes worse

([*outdated - there are no longer MQA files on Tidal*] and we're talking about situation that it's not MQA 16 bit like "Master" quality files on Tidal [but lots are 24 bit as well, though it's better to get them from e.g. Qobuz if 24 bit for some track is available, or at least compare both, because it can give slightly different results]).

Also, it can happen that 24 bit MQA on Tidal will sound better for whatever reason than seemingly better FLAC on Qobuz - it might be possibly due to different files sent to streaming services by the provider/labe.

How to notice difference on spectrogram in e.g. Spek between MQA and FLAC is frequencies from 18kHz (only in certain places) but in all cases - frequencies from 21kHz - press alt-tab between the two windows' - don't hover your mouse between preview of both windows' - use alt-tab - you'll notice the changes easier. That way, you'll notice CD rip vs streaming differences if there are any.

Generally, MQA is the least of lossy codecs - you might consider picking it where its 24 bit variant is available over regular 16 bit FLAC (separate the track using the two, and you should notice any differences easier if you already can't hear them on mixtures/original songs).

Comparisons of various versions of the FLAC files on streaming services

Use Audio MD5 in Foobar >properties of the file (or download [AudioMD5Checker](#)) to not run in circles looking for various versions of the same track with the same length. Some FLAC files don't have MD5 checksums in F2K shown, so you'll need to download AudioMD5Checker.

E.g. on Tidal Recovery by Eminem returns the same MD5 for Deluxe and regular album, but using <https://free-mp3-download.net> (Deezer), checksums are different for both (to differentiate - albums on the net site have various release dates), but Deluxe on Tidal with regular on (Deezer) have the same MD5. And when Audio MD5 checksums were different, there were different results after separation. In this case of one unique vs 3 same MD5, the unique resulted in worse separation (but it can depend on more factors in other cases). Be aware of non-explicit versions which will naturally have different checksum.

Sites and rippers

List of a ways to get lossless files for separation process

Various music is scattered across various streaming services nowadays. If you can't find your track on one service, or its ripper currently doesn't work (it constantly changes) check other streaming service or the net (more below).

List of all lossless streaming services with rippers below:

Tidal, Qobuz, Apple Music, Amazon Music (they support hi-res), Deezer (16/48 FLAC).

0) us.deezer.squid.wtf | (out of order for now, Deezer only; search didn't respond before) eu.deezer.squid.wtf (also offline) - works for queries, not URLs, single songs or albums, if you don't check the "Save songs on download" (supported on Chrome) you need to press download button manually after ripping finishes (so once you're notified), sometimes ripping can be progressing very slow, but consequently when you zoom the progress bar. Also, sometimes downloading in your browser might get interrupted in the middle (press resume in your browser download queue if necessary).

Some users (e.g. French) are unable to reach the site (e.g. 403 error) then use VPN.

0) <https://us.qobuz.squid.wtf> (went offline) | <https://eu.qobuz.squid.wtf> (back offline; Qobuz only) - -||- It can happen that using search on Qobuz won't give you the desired results, while the search in the link will be successful.

0) lucida.to | lucida.su (sometimes works, sometimes redirects to doubledouble.top) (Qobuz, Deezer [may not work], Tidal [icl. 24/96kHz FLAC stereo], Amazon Music, Beatport, lossy: Spotify, Yandex Music, Soundcloud). Various subscription regions, for now there's no Apple Music) - for URLs generated from share option on these services

- Send your request again if you got site unreachable browser error during download
- Sometimes it might fail generating download in the first place
- It frequently shows that it's down, simply retry entering
- "if a track has a status code 404, it means it is unavailable (region locked or completely unavailable) you have to try with another country/account under 'advanced...'"
- Downloading more than EP (7-10 songs) at a time works clunky - it's slow (at least from Tidal), plus during downloading track ~10 it might give an error, so you need to click retry, sometimes a few times, and then the whole album ripping will be completed. It might occur more than once for one album

- Clicking retry while downloading whole albums from e.g. might end up with loops of “An error occurred. Track #1 error: Max retries attempted.” errors, after long wait on “Sending request for item 1” or series of interrupted downloads, while downloading single songs will work fine
 - You cannot download Dolby Atmos versions of songs from e.g. Tidal
 - For “An error occurred. Unexpected token '<', '<html> <h'... is not valid JSON.” just retry the task, or uncheck add metadata option.
 - All linking schemes like:
 - 1) https://www.deezer.com/xx/track/XXXXXXXXXXXX?host=XXXXXXX&utm_campaign=clipboard-generic&utm_source=user_sharing&utm_content=track-XXXXXXXXXXXX&deferredFI=1&universal_link=1
 - 2) Later converted by lucida automatically to: <https://www.deezer.com/xxx/track/XXXXXXX>
 - 3) <https://link.deezer.com/s/xXxXXxxXxxXxxXX>
 can can give “uh-oh!” error - if [stats](#) page shows 0 downloads for Deezer (or any other service), it just doesn’t work
 - Sometimes Tidal links will work after retrying the pasting link request, or if you cut everything which follows <https://tidal.com/track/xxxxxxx> so the “u” (not sure which one).
 - In Amazon linking scheme, referral number appears in the middle, not at the end of the link, so you can debloat the following to restrict the tracker:

https://music.amazon.com/albums/B073JR1FBD?marketplaceId=A3K6Y4MI8GDYMT&musicTerritory=PL&ref=dm_sh_xxxxxxxxxxxxxxxxxxxxxx&trackAsin=B073RRBQR5

 By the following:

<https://music.amazon.com/albums/B073JR1FBD?marketplaceId=A3K6Y4MI8GDYMT&musicTerritory=PL&trackAsin=B073RRBQR5>

 When you use shared Amazon links, sometimes they must be deprived of some information after “&” mark, in order to not return error on the site, but because they contain an “album” string, you might end up with the whole album downloaded instead of a single song if you did it wrong.
- The region string should be rather US instead of whatever your shared links have (it seems the accounts have US region), but other regions in the links might work too.

0) [doubledouble.top](#) (currently Qobuz, Tidal doesn’t seem to work for the EU region for now or for US rarely] and Soundcloud only works) Sometimes works, sometimes redirects to Lucida. For URLs, it supported Apple Music unlike Lucida (but later it stopped working too or worked rarely or with specific music, then lucida started supporting it), and currently from Deezer it returns only mp3 128kbps (check current services status at the bottom).

In specific cases, some streaming services might not have FLAC for your song, then use other streaming service.

*) <https://qqdl.site/> - Qobuz (currently redirects to TIDAL below), some smartphones might run out of memory running this, a new site by Lucida stuff, but seems to currently redirect to doubledouble.top

- 0) <https://tidal.squid.wtf/>
- 0) <https://deezmate.com> - mp3 or FLACs from Deezer, working with links
- 0) <https://tidal.qqdl.site> - TIDAL

Telegram ripping bots

To use Telegram in browser:

<https://web.telegram.org/>

You need the app installed and account registered on your phone, then QR code from there is needed.

1a) Bot

<https://t.me/onlydonuts>

1b) Amazon bot (up to 24/96 FLAC)

<https://t.me/GlomaticoAmazonMusicBot>

Q: "I hit start bot, but nothing happens"

A: "Once you start the bot you must type /codec and send, then it will show a menu where you pick the format you want (mp3, flac, atmos)

After selecting a codec, you simply need to send a link to a track or album.

The bot will download all the tracks in the format you pick, but if the track is not available in Atmos it will be ignored"

1c) Apple Music Bot

<https://t.me/GlomaticoAppleMusicBot>

<https://t.me/bayapplemusicbot>

1d) Deezer Telegram Bot

<https://t.me/deezload2bot> - for ARls [see](#) (but those public get taken down often)

1e) Spotify / Deezer / Tidal / Yandex / VK / FLAC / 25 Daily bot

<https://t.me/BeatSpotBot>

1f) Deezer mp3/FLAC bot

<https://t.me/DeezerMusicBot>

1e) [VK Bot](#), [vkmusbot](#) or [Meph Bot](#) - VK / 320kbps MP3

Rippers

2) [Murglar](#) app - [apk](#) for Android - *player and downloader working with Deezer, SoundCloud, VKontakte and Yandex Music (alternatively you can use it in Android virtual machine)*

3) Apple Music ALAC/Atmos downloader

<https://github.com/adoalin/apple-music-alac-downloader> (*valid subscription required, you can't use an account that's on a family sharing plan, more about installation below in Dolby Atmos section*)

Might be less comfy to install for beginners. It requires Android (rooted at best and in Android Studio w/o Google APIs) and installing specific Frida server version (for not rooted devices it might be more complicated) and specific version of Apple Music app.

Refer to GitHub link above and Frida website for further instructions.

(the section continues later below)

4) <https://github.com/zhaarey/apple-music-downloader>

[Instruction](#)

“and additional note if that does not work

replace Part 2 Line 2

Download and extract the NDK needed to build the wrapper.

```
wget https://dl.google.com/android/repository/android-ndk-r23b-linux.zip && unzip  
android-ndk-r23b-linux.zip -d ~
```

with

Download and extract the NDK needed to build the wrapper.

```
wget https://dl.google.com/android/repository/android-ndk-r27c-linux.zip && unzip  
android-ndk-r27c-linux.zip -d ~
```

the change is the Android NDK version from 23b to 27c”

Bas Curtiz tutorial:

<https://www.youtube.com/watch?v=eJ7a3W8qy5o>

General bots usage instructions

Go to proper dl request channel and write

!dl

and after !dl (on Discord \$dl), paste a link to your album or song from the desired streaming service and send the message, e.g.

!dl <https://www.deezer.com/en/track/XXXXXX>

Follow this link pattern. Sometimes the sharing option on the site changes the link pattern, so you need to open the changed link, and then it will redirect to the one similarly looking like above.

To open the Deezer player to search for files without active subscription, log-in and just go to:

<https://www.deezer.com/search/rjd2%20deadringer>

And replace the search query with yours after opening the link.

If the bot doesn't respond in an instant, it probably means the track/album is regional-blocked, and you should use a link from another service or another channel (UK and NZ alternative servers available). It's capable of printing out unavailability errors as well.

Some bots rip tracks or whole albums from Qobuz, Deezer, Tidal - all losslessly, while: Spotify, Soundcloud Go+, YouTube Music, JioSaavn are lossy.

Providing valid links for bot

For your comfort, you should register and log into every streaming service and share links for specific tracks or albums from these services (e.g. instead of pasting full album links if you want), when you can't simply find a specific single track in Google for this service, or share the link only for it comfortably. So basically go to <https://play.qobuz.com/>, and you can share single tracks to paste for bot to download - available only after logging into free account and only in the link above instead of regular Qobuz file search you can find in Google - there you cannot share single songs to download using bot later. It can happen that you'll see an error that Qobuz is not available in your country. It's fine - you won't have to buy a subscription at this step in order to use their search. It's enough to log-in using specific link and not the main page, use this one:

<https://play.qobuz.com/search/tracks>

And it will allow you to log in.

Because the bot rips from Qobuz, it's the best source of 24 bit files which I recommend if only available (either 44, 48 or 96kHz) as it delivers FLACs for end users, instead of partly lossy MQA on Tidal when some album/song uses Master quality which is compulsory for 24 bit (44/48) there, but MQA 16 bit and Master is also possible for some albums (and you should avoid 16 bit MQA). Of course there might be some exceptions where 24 bit MQA on Tidal will sound better than FLAC 24 bit on Qobuz as I mentioned above - the example is Eminem - Music To Be Murdered By (Deluxe Edition) - Volume 1 (the newer Side B, track - Book of Rhymes).

For using Deezer links with bot, you need to find a song/album, use option to share a link to track or album, then open the shared link so it will be redirected, and then rename the link to this form for a single song (otherwise bot will return “processing” instead of ripping or even possible error):

<https://www.deezer.com/en/track/XXXXXXX>

(ARLs trick doesn't work anymore) Hint: There's also something like ARL, which is a cookie's session identifier which can be shared, so everyone can log into the premium account and download FLACs with ARLs of different regions and regional locks. Might be useful for some specific tracks. ARLs are frequently shared online, though harder to find nowadays (Reddit censorship).

IRC, Deemix might use ARLs beside regular account log in process.

5) [Tidal Downloader Pro](#) (the fastest method for batch and local downloading) in GUI. HiFi Plus subscription is no longer necessary, just valid Hi-Fi subscription (for at least Hi-Fi albums, the two are merged in one for the price of the cheaper now).

You won't be able to download with better quality than 24 bit/48kHz and in Atmos with this downloader (then use [orpheusdl_tidal](#) instead, or [tidal-dl-ng](#) - but for that one I'm not sure if it downloads Atmos files)

Install Tidal app on Windows and log in, then open the downloader and click log, copy and paste the given code in the opened browser tab and voila.

Or if that GUI temporarily doesn't work, go to:

<https://github.com/yaronzz/Tidal-Media-Downloader/releases> and download the newest source code. It contains CMD version for downloading, located in:
Tidal-Media-Downloader-202x.xx.xx.x\TIDALDL-PY\exe
Documentation: https://doc.yaronzz.com/post/tidal_dl_installation/

If you have problems with running the app and people also write in GitHub issues that the current version is not working, keep tracking new versions, or read all the issues about this version, it may happen that someone else will update the app before.

Versions “2022.01.21.1” and “1.2.1.9” need to be updated to newer versions, they stopped working entirely.

(not needed anymore, as current should still work)

You can alternatively grab this [recompiled](#) version by another user.

By these downloaders you can easily download whole albums including hi-res and in GUI (PRO), and also queue for single tracks to download automatically is available (Pro).

There are cases when certain songs are behind regional block, and won't be downloaded by any Divolt or Discord bot resulting in error.

In such a case, you'll need the above downloader used locally, along with a Hi-Fi Plus subscription bought for your localization. Accounts bought from elsewhere, or paid with foreign currency, will most likely have regional block for some other country, so after you log into the service, certain songs won't show in search, so the only way to show them without proper account (at least for your region) is to log out from regional locked account, start new account, and visit: <https://listen.tidal.com/> (you don't need to have a valid subscription to search for songs on Tidal).

Besides trial, you can go for Tidal Hi-Fi cheap subscription to:

<https://www.hotukdeals.com/vouchers/tidal.com> or pepper.pl or mydealz.de which always have some free or almost free giveaways (linked to a ready search). Then install the desktop Tidal app and log in and open the downloader. It might automate the login process in the downloader

(if you need to switch an account, you better delete Tidal-GUI folder from your documents folder in case of any problems). Monthly Argentinian subscription is the most reliable solution now if you don't want to change your account every month or two searching for new offers.

Tidal over some other streaming services has some tracks in master quality which is 24 bit, and it gives better results for separation as the dynamics are usually better. But check if your downloaded file is really a 24 bit and your downloader is configured properly (read the documentation in case of any issues).

But, on Tidal there ~~were~~ were some fake master files in the past, which in reality were 16 bits, and they're MQA to save space on their servers or mislead people, so there is no benefit from using them vs Audio CD 16 bit rip, since MQA alters quality in higher frequencies (only) and it will have an influence on separation process. So to verify if your downloader is set up properly, check whether you can download any track from Music To Be Murdered By, by Eminem in 24 bit. If you can, you have properly installed and authorized the downloader, so it can download 24 bit files or at a higher sample rate than 44kHz if available.

You can paste links from Tidal into the GUI browser to find that track. Just delete "?u" at the end of the shared link.

5b) Colab for downloading from Tidal and Qobuz using your own valid account (based on streamrip, active subscription required):

colab.research.google.com/github/r-piratedgames/rip/blob/master/rip.ipynb

6) For Deezer <https://archive.org/details/deemix> - it allows you to download mp3 320 and FLAC files for premium Deezer accounts, and only mp3 128kbps for free Deezer accounts.

Be aware that deemix.pro site is unofficial, and the PC 2020 version linked there is not functional. The last 2022 is on the archive.org linked above from reddit.

Qobuz or Deezer might give better results since Tidal is recently deleting FLAC support for 16 bit files on some albums, making all the files 16 bit MQA, which is not fully lossless file format, but close (of course Tidal Downloader converts the same MQA to FLAC). It also provides some high resolution files, but most likely less of them than on Tidal.

Be aware that using some streaming services downloaders or even official Deezer/Tidal/Spotify apps, you might not be able to find or even play there some specific tracks or albums due to:

- a) premium lock (it won't be played for free users)
- b) regional lock (search will come up empty [the same applies to Tidal here])

Example: Spyair - Imagination instrumental - it shows up in search probably in Japan, though it cannot be downloaded using 2) <https://free-mp3-download.net>, but deemix with premium Deezer subscription did the job in downloading the song (not sure if it was Japan account).

PS. You can cancel your trial subscription of Deezer or Tidal immediately to avoid being charged in the future, but also keeping the access to premium till the previous charge date at the same time.

7) <https://github.com/yarrm80s/orpheusdl>

Supports Qobuz, Tidal (with [this](#) module, and unlike tidal-dl, also downloads files greater than 24/48 and Atmos) and probably more

7a) Bas Curtiz [GUI](#) for Orpheus (still needs working subscription)

7b) [QobuzDownloaderX-MOD](#)

(*May not work anymore)

7*) If you have a Qobuz subscription, you can just use [qobuz-dl](#) (last updated a year ago, probably no longer works, but not sure, there might be some alternative already).

Alternatively check:

Qobuz Downloader X

or Allavsoft (both requires subscription)

<https://www.qobuz-dl.com/> (takendown frontend browser client for downloading music for Qobuz. The code for hosting on [GH](#))

7b) <https://github.com/nathom/streamrip>

A scriptable stream downloader for Qobuz, Tidal, Deezer (active subscription required) and SoundCloud.

8*) For Deezer you can use [Deezloader](#) or Deezloader Remix - it doesn't require any subscription for mp3 128kbps, just register a Deezer account for free before, and use the

account in the app. For free users it gives only mp3 128kbps with 16kHz, so it's worse than YT and Opus, so don't bother.

9a) For Spotify, you can use Soggfy, or

9b) SpotiDown (premium subscription for 320kbps downloading and app compiling required)

9c** Seemingly you can use <https://spotiflyer.app/>

but it "doesn't download from Spotify, but from Saavn, in 128kbps/low-quality.

Also, since it doesn't d/l from Spotify, you can't d/l exclusives released from there."

It doesn't require a valid subscription irc and also allows playing and sharing music inside the app.

9d** The same sadly goes to this telegram bot downloader:

https://t.me/Spotify_downloa_bot

9e) <https://spotify-downloader.com/>

Other lists of rippers and sites:

<https://retnry.org/firehawk52>

<https://ripped.guide/Audio/Music/>

<https://fmhy.net/audiopiracyguide#audio-ripping-sites>

Sites

10) Go to allflac.com - it's paid, but they don't pay royalties to the artist and its labels, as I spoke with at least one. They don't keep up with the content with the streaming services, but they share stuff also not available on streaming services, even including vinyl rips as hi-res ones. Most if not all the files on the site are CD rips taken from around the net.

I'll explain to you how to download files for free from allflac and flacit:

0. Log in
1. Find desired album (don't press play yet!)
2. Open the Chrome Menu in the upper-right-hand corner of the browser window and select More Tools > Developer Tools>navigate to "Network"
3. Press CTRL+R as prompted
4. Play audio file

5. If it's 16/44 FLAC, go to media, sort by size, right-click on the entry and open in new tab to download (sometimes it appear after some time of playing and only in "all" instead of "media")

6*. On some 24 bit files, go to all, play the file and sort by size. You will find an entry with increasing size with xhr type and flac name if it's not shown in the media tab.

7. Recently it happened once, that the point five stopped working and the FLAC link is red. Now you need to go to the console and open a link with ERR_CERT_DATE_INVALID in the new tab and open the site, clicking on advanced.

In case of 32 KB/s download, get Free Download Manager, and paste the download link there, and with 2 active connections in the downloader, it will speed up to 96KB/s occasionally (properly set JDownloader also allows increasing number of connections). Haven't tried switching accounts to check if it will make the speeds back to normal (it wasn't like that before).

Some albums on allflac.com don't have tracks separated, but all the albums are in track 1. If you want to physically divide the audio file -

In such case, you can search for cue sheet here: <https://www.regeert.nl/cuesheet/> Place it near the file, and eventually rename, and it's ready, but it's only for playing and playlist purposes. It doesn't separate the audio file physically. To cut the file losslessly you need lossless-cut <https://github.com/mifi/lossless-cut/releases/> - it allows importing cue sheets to cut the album. Now if you have all the files divided you can probably use MusicBrainz (probably Foobar2000 plugin is available) to tag the files (but not the filenames - for that, you need mp3tag and tagged files to copy tags to filenames with specific masks). I know that lossless-cut might be not precise, and it may create a problem with automatic content detection in MusicBrainz, but I know that tool or similar allowed to just search for the album you specifically searched for, and not by just mark files>album detection in Foobar which may fail. So technically cutting and tagging the files should be possible, but time-consuming.

Looks like, unlike 24/48 files, all 24/96/192kHz ones are just vinyl rips taken from various torrents. If again there's only one or two files with the whole album, originally attached with cue, you should be able to find specific cue files simply searching in Google for its specific file name with quotes (file list is below track list there). Of course, you can also cut your album manually, or even make your own cue sheet to cut the album.

Also be aware that sometimes you won't be able to download the file, and it won't appear as FLAC, if you do not press CTRL+R on Network before starting playing the file, otherwise you need to close and reopen the tab and press CTRL+R in Network again.

And also, such files can reset during downloading near the end (maximum size of downloaded file cannot exceed 1GB, otherwise it gets reset for some reason). To prevent it, copy the download link from your browser, and paste it to some download accelerator. Even free BitComet will do the trick since it supports HTTP multiple connection downloading. If you're lazy, to prevent losing at least these 1GB, simply open the still downloaded file using

MPC-HC and Chrome won't reset the file size after it starts to reset the whole download (because the file cannot be deleted now), wait for the reset of the download, now just make a copy of the file and rename file extension to FLAC from temporary extension added by e.g. Chrome during downloading. Now you can stop downloading in Chrome. The downside is - the moment the file gets reset is not when it ends, meaning it's not fully complete. But mostly. Of course, you can be lucky enough to find the original torrent with the files and simply finish downloading by verifying checksums of existing ones in P2P client (filenames must match to torrent files, simply replace them and find option to verify checksums).

10b. All of the above applies to <https://www.flacit.com/>

Looks like it has the same library taken from
adamsfile.com

which is also a warez site allowing playing files and downloading them using the method above.

You also need to register before playing any file there (registration is free).

11. <http://flacmusicfinder.com/>

But it has a small library.

*. FLAC sites listed [here](#)

12. Soulseek - but it's simply a P2P based client, so carefully with that, and better use VPN (good one at best). GUI - [Nicotine+](#) and Seeker working on Android

13. Rutracker (the same advice as above)

14. Chomikuj.pl (use their search engine, eventually Yandex, Duckduckgo, Bing) - free 50MB per week for unlimited amounts of accounts, free transfer for points from files uploaded or shared from other people's profiles. People upload there separate tracks or loose as well, but they frequently get taken down, so search for rather full album titles in archives rather than single files. Mp3 files and those files which allow preview, can be downloaded for free with JDownloader, but occasionally some of such files might not work in JDownloader, and they'll have to be downloaded manually.

15. <https://music.binimum.org>

16. <https://monochrome.tf>

17. <https://dab.yeet.su> (requires free account creation)

18. Simply search for the track on Google, or even better - Yandex, Duckduckgo, eventually Bing, because Google frequently blacklists some sites or search entries. Also, your specific

provider may cut connection to some sites, so you'll be forced to use VPN in those cases when a search engine shows up a result from a site you cannot open.

19. Redtopia archives on torrent sites

20. YouTube Music - higher bitrate (256kbps) than max 128/160kbps on regular YT for Opus/webm (20kHz) and AAC/M4A 128kbps (16kHz). Similarly, like in Spotify - it can possess some exclusive files which are unavailable in lossless form on any other platform, but most on YTM are available losslessly on other streaming services, so use them instead. If you have YouTube Premium you apparently can download files from it if you provide your token properly to yt-dl.

Maybe logging into Google account with enabled premium in JDownloader 2 will do the trick as well.

Anyway, Divolt bot (or any other currently available) will work too.

Outdated/closed/defunct

(it's been closed)

0) Go to <https://free-mp3-download.net> (Deezer, FLAC, separate songs downloading)
Here you can find (all?) mp3/flac files from Deezer. If the site doesn't work for you, use VPN. If the site doesn't search, mark "use our VPN". Single files download and captcha. No tags for track numbers and file names, FLAC/MP3 16 bit only.

- If you see an error "file not found on this server" don't refresh, but go back and click download again.

- From time to time it happened that it didn't show up the FLAC option, and that it's "unavailable", and sometimes it can show up after some period of time. The site started to have some problems, but it was fixed already.

- Open every found track in a new tab, as back button won't allow you to see search results you looked for

1 b) (doesn't work anymore for 07.02.24)

Discord server with sharing bot (albums and songs)

<https://discord.gg/MmE4JnUVA>

-||-

<https://discord.gg/2HjATw6JF>

(another invite link valid till 12.11.13; needs to be renewed every month, probably current invitations will be on Divolt server here when the above will expire)

Later, they required writing to the bot via DM to access the welcome channel with requests. Once I couldn't access the channel, and I needed to update Discord or wait 10-15 minutes, so the input form appeared.

To download, in welcome channel, paste:

\$dl [link to the song or album on streaming service without brackets]

More detailed instruction of usage below.

(Defunct)

2) <https://slavart.gamesdrive.net/tracks>

(sometimes used to work, but not too often)

As of June 2023-March 2024 it is defunct, and throws: "There was an error processing your request!" on track download attempt, or in the past it was loading forever and nothing happens on multiple tries, before it worked after download button will stop being gray, and it's green again, so you should click it and download may start shortly, but it stopped, lately it was working, you only needed to wait a bit.

Similar search engine for FLACs. Files are sourced from Qobuz (including hi-res when available). Songs listed double are sometimes in higher bit depth/resolution (different versions of the same track).

If you want to know what is the version you download, go to <https://play.qobuz.com/> share track from there, and use download bots.

1 b) Join their Divolt server directly by this link (if the above stopped working):

<https://divolt.xyz/invite/Qxxstb7Q> (currently the bot don't allow posting, containing only

Discord invite, check it again later for valid link if necessary)

Free registration required.

If this Divolt server is also down, go here:

<https://slavart.gamesdrive.net/> (defunct)

to get a valid Divolt invite link (it might have changed). But it had the old link for the long time later.

0) yams.tf (offline) (Qobuz, Tidal, Deezer, Spotify, Apple Music [currently 320 kbps]) - for URLs, currently doesn't seem to work with even VPNs

Dolby Atmos ripping

"Streamed Dolby Atmos is eac (5.1 Surround) and JOC (Joint Object coding) it's a hybrid file of channels and objects that decodes the 5.1 + joc to whatever your speakers are from 2.0 up to 9.1.6.

It's not a multitrack, although clearly what some mixers do is put all the vocal in the center channel, so effectively you have an a cappella in center and then the instrumental in

everything else, but many labels forbid engineers doing it and have policies that they must mix other sounds into center, so people don't rip the a cappella.

[“apparently Logic Pro does it automatically as well” isling, src: ScretTure]

YouTube only supports FOA Ambisonics as spatial audio, but you can encode Dolby Atmos to Ambisonics. [by e.g. <https://www.mach1.tech/>]

Apple Music has a larger amount [of Atmos songs] because Apple Pay the labels for exclusive Atmos deals.” ~Sam Hocking

Tidal only supports 5.1 or maybe 5.1.4, and Apple Music at least up to 7.1.4 (9.1.6 support could have been dropped since macOS Sonoma, not sure “On latest MacOS you do now have the ability to decode directly to 7.1.4 pcm realtime from Apple Music.”).

“I tried using channels from an Atmos mix to get better instrumentals and very surprisingly it sounds a lot worse

I rendered it into 7.1 and upmixed the channels into 3 separate stereo tracks, and processed each using unwa's v1e+

It ended up sounding more muddy than using a lossless stereo mix” - santilli_

“sometimes rendering into 9.1.6 is good for some instruments yet everyone says it's really unnecessary

which is kinda true

like the 1-2 channel for 9.1.6 dolby is insanely clean on the songs i've tried but some other stems sound a bit ehh” - Isling

- from Tidal (via ~~Tidal Media Downloader PRO [Tidal-DL-GUI]~~)

(doesn't work anymore for Atmos; see further below)

Just get Tidal-dl with HiFi Plus [subscription](#) - now merged into one subscription (CLI version; for one user on our server it works for 13.10.22, but for some people strangely not).

For 30.04.24 with Tidal app installed on Windows and tidal-gui authorized by browser prompt/or automatically, Atmos files are not downloaded (checked all qualities in settings incl. 720/1080), at least on subscription automatically converted into higher plan due to recent changes (MQA files started to play since then, so it might be not subscription issue).

If having some problems, use tidal-dl (non-GUI) and tidal account with valid subscription and proper plan, set up to fire tv device api (option 5 iirc).

But I cannot guarantee it will work for Atmos.

> from Tidal (with [orpheus_dl_tidal](#) installed over [orpheusDL](#); max 5.1[.4?])

Downloads [Atmos](#) and [high resolution](#) files bigger than 24/48.

It's only CLI app (valid [subscription](#) is still required).

A bit convoluted installation.

If you have problem with using git in the Windows command line, use [this](#) ready OrpheusDL package (works for 30.04.24, later it can get outdated; it already has Tidal settings and Atmos enabled) after you install python-3.9.13 or newer (works currently also on python-3.12.3-amd64).

Or else, to install manually following GH instructions, to fix git issue, execute:

pip install gitpython

and/or install git from [here](#)

(one or both of these should fix using git in CML when pip install git cannot find supported distribution and git command is not recognized).

Once Python and the OrpheusDL package is correctly installed, CML usage is:

```
orpheus https://tidal.com/browse/track/280733977
```

You can place it as parameter for shortcut to orpheus.py on your desktop in Target (PPM on shortcut).

E.g. "C:\Program Files\OrpheusDL\orpheus.py" https://tidal.com/browse/track/280733977

Or else, press Win+R>cmd.exe, and if you're currently not at the same partition as Orpheus (e.g. C:\) press e.g.

d:\

and seek to the folder you have Orpheus installed, e.g.

```
cd D:\Program Files\OrpheusDL\
```

then execute

```
orpheus https://tidal.com/browse/track/280733977
```

Always delete "?u" at the end of the link copied from Tidal, or it won't work.

Once you execute the command, it will ask you for login method (I tested the first one - TV) - now it will redirect to your browser to authorize.

MQA is disabled by default (not used by Atmos), but you can enable it in config\settings\ by editing "proprietary_codecs": to false in line 21.

Downloaded files are located in OrpheusDL\downloads folder

spatial_codecs flag is enabled by default and supports Dolby Atmos and 360 Reality Audio.

"Some of the 360 stuff is impossible to split right now. Not sure what is going on. Maybe some type of new encryption. I have the MP4 to David Bowie Heroes 22 channels, and it's a brick, useless..."

The output of downloaded Atmos files is m4a encoded in E-AC-3 JOC (Enhanced AC-3 with Joint Object) - Dolby Digital Plus with Dolby Atmos and possibly AC-4, and FLAC for hi-res. Downloaded hi-res and Atmos files can be played in e.g. MPC-HC or VLC Media Player, but will fail on some old players like Foobar2000 1.3 and 1.6.

> from Tidal (with <https://github.com/exislow/tidal-dl-ng>)

- from Apple Music (Android, max 7.1[.4?])

<https://github.com/adoalin/apple-music-alac-downloader>

Installation tutorial:

<https://www.youtube.com/watch?v=blazHnCh6jQ>

“Pre-Requisites:

x86_64 bit device (Intel/AMD Only)

Install Python: <https://www.python.org/>

Install Go: <https://go.dev/doc/install>

Install Android Platform Tools: <https://developer.android.com/tools/releases/platform-tools>
and set it to environment variables / path

Download and extract Frida Server -

https://github.com/frida/frida/releases/download/16.2.1/frida-server-16.2.1-android-x86_64.xz

Download Apple Music ALAC Downloader -

<https://github.com/adoalin/apple-music-alac-downloader>

Extract content to any folder.

1)

Install Android Studio

Create a virtual device on Android Studio with an image that doesn't have Google APIs.

2)

Install SAI - <https://github.com/Aefyr/SAI>

Install Apple Music 3.6.0 beta 4 -

<https://www.apkmirror.com/apk/apple/apple-music/apple-music-3-6-0-beta-release/apple-music-3-6-0-beta-4-android-apk-download/>

Launch Apple Music and sign in to your account. Subscription required.

3)

Open Terminal

adb forward tcp:10020 tcp:10020

if u get a msg that there are more than 1 emulator/devices running, seek up

NTKDaemonService in task manager/services and stop it

adb root

cd frida-server-16.2.1-android-x86_64

adb push frida-server-16.2.1-android-x86_64 /data/local/tmp/

adb shell "chmod 755 /data/local/tmp/frida-server-16.2.1-android-x86_64"

adb shell "/data/local/tmp/frida-server-16.2.1-android-x86_64 &"

The steps above place Frida-server on your Android device and starts the Frida-server.

4)

Open a new Terminal window

Change directory to Apple Music ALAC Downloader folder location

pip install frida-tools

frida -U -I agent.js -f com.apple.android.music

5)

Open a new Terminal window

Change directory to Apple Music ALAC Downloader folder location

Start downloading some albums:

go run main.go https://music.apple.com/us/album/beautiful-things-single/1724488123

go run main_atmos.go "https://music.apple.com/hk/album/周杰倫地表最強世界巡迴演唱會/1721464851"

- from Apple Music (alternative tool)

<https://rentry.co/AppleMusicDecrypt>

(after March 5, 2025 get WSA from here:

<https://github.com/MustardChef/WSABuilds>)

from Apple Music (MacOS, virtual soundcard recording)

(Guide by Mikeyyyyy/K-Kop Filters, [source](#))

You will need a Mac to do this, this will only work for MacOS, you will need an Apple Music subscription, "Blackhole 16ch" and any DAW of your choice I prefer FL Studio (can be also Audacity),

Step 1. Install Blackhole Audio driver (search for it in Google)

Step 2. Download the song you want in Dolby Atmos (if you don't know how to do it, go to settings in Apple Music then to general then toggle download Dolby Atmos)

Step 3. Go to your desired DAW and in your mixed select input, and it will show your 16 outputs select 1, (Mono) for the first mixer, then number 2 mixer do the same but 2 and so on until you reached 6

Step 4. Hit record and play the track in Dolby, and you're done!

[Similar](#) tutorial based on Blackhole and Audacity on Mac (open the link in incognito in case of infinite captcha)

"On latest MacOS you do now have the ability to decode directly to 7.1.4 pcm realtime from Apple Music. If you use a loopback virtual audio driver you can record the 12 channels. Depending on how song was mixed might mean the C channel has even even clearer vocal. Probably worth mentioning Dolby Atmos is delivered as dd+ (Dolby Digital 5.1 Surround downmix) but JOC allows it to be decoded up to 9.1.6 16 channels. To do that you need either an AVR or Dolby Reference Player or Cavern/Cavernize." Sam

You won't be able to do the same on Windows with [LoopBeAudio](#) instead (paid, but trial works for every 60 minutes after boot) because Apple Music on Windows (including the one in MS Store) doesn't provide Dolby Atmos (7.3.1) files at all (only stereo hi-res lossless) no matter what virtual soundcard you use, so you'll need Hackintosh or VMware.

"Vmware kinda lag
and find own seri to fix login apple services"

- [ittiam-systems/libmpegh: MPEG-H 3D Audio Low Complexity Profile Decoder](#)

Using this program, you can extract the 12 channels of the Dolby Atmos tracks.

"MPEG-H is essentially Sony360, just Sony360 licenced decoders needed. Fraunhofer allow it to be used for free, though.

ia_mpeghd_testbench.exe -ifile:"FILENAME.mhm" -ofile:track1.wav

or:

ia_mpeghd_testbench.exe -ifile:"input file name.m4a" -ofile:"output file name.wav" -cicp:13
"renders to 22.2 as well"

<https://mpegh.lze-innovation.de/#LZE>

But seems like you need to write them some message.

Above it tells it's for professionals, but try your luck:

https://www.iis.fraunhofer.de/en/ff/amm/broadcast-streaming/mpegh.html?source=post_page

You should also have a success with extracting stems with MMH Atmos Helper "includes a MPEG-H decoder built-in apparently"

All Dolby Atmos is encoded, so to play it, basically it has to be decoded to audio playback through a Dolby licensed decoder. There are ways to decode, though. Easiest is to use Cavern.

<https://cavern.sbence.hu/cavern/>

Atmos is a lossy format. 768kbps across 6 channels so not the highest resolution, but to decode to multichannel .wav just download cavern and put your dd+joc file through Cavernize. Streamed Atmos [is lossy]. TrueHD Atmos isn't. Atmos Music is only distributed lossy, though.

You can encode Dolby Atmos to Ambisonics by e.g. <https://www.mach1.tech/>.

Atmos files downloaded from Tidal with OrpheusDL are simply FLACs in m4a container, and can be read by MPC-HC, VLC, and Foobar 2.x.

On the side, “The process of making Atmos [*from an engineer standpoint*] is:
DAW > 128 channel ADM_BWF > 7.1.4 >5.1(joc). So basically those 128 channels are encoded to 6, but the object audio is still known where it should exist in the space and pulls that audio out of the 5.1 channels to make up to 9.1.6 (max supported for music)”

And authoring for Atmos is not available on Windows but:

“Traditionally it's not been unless you ordered a DELL from Dolby configured to use as a Rendering machine, but today both Dolby Atmos Renderer, DAWs like Cubase and Nuendo and 3rd party VST exist to do it on Windows now. I use Fiedler Atmos Composer on a stereo DAW called Bitwig to build demix projects for Atmos engineers to then master to Atmos from Stereo (sometimes all they have left to work with as multitrack tapes lost/destroyed/politics/easier)” ~Sam Hocking

360 Reality Audio FAQ

“A lot more stems in 360, and it has no bleeding or filtered sounding artifacts.
A big problem with Dolby stems is artifacts, essentially none of that in 360.
And if there is bleeding in 360 then the volume is completely balanced and doesn't change all the time” - I.

Q (isling): Is there a way to just listen to 360 files locally while still getting the immersive 3d effect?

Decoding them and listening in DAWs don't sound like how they do on Amazon Music for example.

Is there even a way to listen to them without decoding? I've downloaded the MPEG-H software stuff.

The MPEG-H format player doesn't read the decoded WAVs nor supports m4a which is what the original non-decoded files are.

The 360 WalkMix player didn't work either.

<https://github.com/ittiam-systems/libmpegh/releases>

This one is the one I use, works great to use, but doesn't have the same 3D effect.
I tried playing the decoded 360 Reality Audio stems in VLC as it has the best Dolby effect, but it didn't have all the channels playing.

A (Sam Hocking): I've worked it out, there's a free decoder. These are the main MPEG-H tools for both creating and playing MPEG-H 3D: <https://mpegh.lze-innovation.de/>
All free. The plugin is really cool. Request the plugin from their site.

Dolby and Sony 360 are object based [not channel base like Ambisonics]. Sony 360 is just protected MPEG-H 3D.

When you rip from Tidal, you can choose how you decode the file to channel-based audio. This is the point of object-based audio, you decode it to what number of speakers you have

If you use tidal-gui, enter an Android token from your Tidal app on the phone and download the file, then you can decode the Sony 360 / MPEG-H by the following:

Description in format Front/Surr.LFE

- 1: 1/0.0 - C
- 2: 2/0.0 - L, R
- 3: 3/0.0 - C, L, R
- 4: 3/1.0 - C, L, R, Cs
- 5: 3/2.0 - C, L, R, Ls, Rs
- 6: 3/2.1 - C, L, R, Ls, Rs, LFE
- 7: 5/2.1 - C, Lc, Rc, L, R, Ls, Rs, LFE
- 8: NA
- 9: 2/1.0 - L, R, Cs
- 10: 2/2.0 - L, R, Ls, Rs
- 11: 3/3.1 - C, L, R, Ls, Rs, Cs, LFE
- 12: 3/4.1 - C, L, R, Ls, Rs, Lsr, Rsr, LFE
- 13: 11/11.2 - C, Lc, Rc, L, R, Lss, Rss, Lsr, Rsr, Cs, LFE, LFE2, Cv, Lv, Rv, Lvss, Rvss, Ts, Lvr, Rvr, Cvr, Cb, Lb, Rb
- 14: 5/2.1 - C, L, R, Ls, Rs, LFE, Lv, Rv
- 15: 5/5.2 - C, L, R, Lss, Rss, Ls, Rs, Lv, Rv, Cvr, LFE, LFE2
- 16: 5/4.1 - C, L, R, Ls, Rs, LFE, Lv, Rv, Lvs, Rvs
- 17: 6/5.1 - C, L, R, Ls, Rs, LFE, Lv, Rv, Cv, Lvs, Rvs, Ts
- 18: 6/7.1 - C, L, R, Ls, Rs, Lbs, Rbs, LFE, Lv, Rv, Cv, Lvs, Rvs, Ts
- 19: 5/6.1 - C, L, R, Lss, Rss, Lsr, Rsr, LFE, Lv, Rv, Lvr, Rvr
- 20: 7/6.1 - C, Leos, Reos, L, R, Lss, Rss, Lsr, Rsr, LFE, Lv, Rv, Lvs, Rvs

Note: CICP 13 is applicable for baseline profile streams with only object audio.
But the delivery is all contained within 12 channels (7.1.4)

There are different levels of MPEG-H. For streaming, it's 7.1.4 which is level 3 IIRC.

Q: 3rd order Ambisonics you mean? And 7.1.4 is literally just Dolby, right?

A: No 7.1.4. You can consider it the same as Dolby Atmos.

It's object-based audio, Ambisonics is channel based audio.

Q: Isn't 360 RA Ambisonics though? Dolby is object based, right?

A: Yep, Dolby and Sony 360 are object based. Sony 360 is just protected MPEG-H 3D

Q: So Sony 360 is also object based? So it's not Ambisonics

A: [Sony 360 7.1.4 "decoded to normal channel-based audio" looks like just 12 stems which can be imported into Audacity]

Q: The one I got was from Amazon Music, not Tidal. Shouldn't make a difference?

A: It's actually far more powerful than Dolby Atmos in this sense.

Music Media Helper can decode Sony 360. Last time i checked they are possible to rip from Tidal although iirc Tidal are dropping 360 support?

<https://www.quadrphonicquad.com/forums/threads/music-media-helper-tools-for-multichannel-audio-music-videos.22693/> (Sam)

Alternatively, this paid plugin can handle 360RA downmix to binaural audio.

<https://www.perfectsurround.com/> but you need an iLok ID even for the free trial (jarredou)

AI mastering services

Might be useful even for enhancing quality of instrumentals after separation (or your own mixed music)

Be aware that at least some advanced mixing beforehand may cheat the content ID detection system, so your song won't be detected. If some label prevents from uploading their stuff on YT by blocking it straight after uploading regular file, you may get a copyright strike after some time of uploading mastered instrumental as they also use search engine on YT too to find their tracks.

If you don't find satisfying results with the services below, read [that](#).

Paid

<https://emastered.com/> (unlimited free preview, 150\$ per year)

Preview is just mp3 320kbps @20kHz cutoff, which is claimed to have a watermark, but it cannot be heard or seen in Spek. The preview file can be downloaded by opening Developer Tools in browser, and playing preview, then in "media", the proper file should appear on the list (don't confuse it with original file), now open the proper link in the new tab and open options of the media player and simply click download.

It's the most advanced and better sounding service vs all free ones I tested (even if you have only access to mp3, but I also listened to max 24 bit WAVs on their site with a paid account). Also, it's one of those, which are potentially destructive if you apply wrong settings, but leaving everything in default state is a good starting point, and works decent for e.g. mixtures and even previously mastered music to some extent, at least which does not hit 0dB (but e.g. even -1dB, but it is claimed to work the best between -3dB and -6dB). Generally I recommend it. Worth trying out.

Note for paid users - be aware that preview files can be mp3 files as well. So what you hear during changing various parameters, is not exactly the same as final WAV output.

<https://distrokid.com/mixeia> (99\$ per year/first master for free)

"[vs LANDR, BandLab and eMastered] I experienced that Mixea mastered with a much stronger sound and brighter (in a good way, the trebles are very clear) than the others."

<https://www.masteringbox.com/> >

<https://www.landr.com/> (now also plugin available)

<https://masterchannel.ai> (15/20\$ per month, only free previews, also can convert stereo to multichannel audio)

<https://ariamastering.com/en/Pricing> (from 50\$ per month or 9.90\$ per master, mastering based on fully analog gear and robotic arm to make adjustments in real time)

VST plugins

[iZotope Ozone Advanced](#) 9 and up (paid)

Version Advanced has a new AI mastering feature which automatically detects parameters which can be manually adjusted after the process. It works pretty well, and repairs lots of problems with muddy mixes (especially with manual adjustments - don't be afraid to experiment - AI is never perfect).

Mastering Assistant built-in the recent versions of [Logic Pro](#) DAW (MacOS only)

It can give more natural results than Izotope above

[AI Master by Exonic UK](#) (paid)

[master_me](#) (free)

It contains a decent mastering chain which adjusts settings for you automatically for the song which can be changed later, and also you can change target ilufs value manually. By default, it's -14 ilufs and can be too quiet for songs already mastered louder, and it can become destructive while set that way for some songs

Free online services (all below remarks apply when mastering AI separated instrumentals)

<https://aimastering.com/> (redirects to <https://bakuage.com/app/>)

wav, mp3, mp4 accepted, output: wav 16-32, mp3 320kbps, 44 or 48kHz

You can optionally specify a reference audio file.

Tons of options but not comfortable preview during tweaking them. You can optionally specify the reference audio, uploading a file. Also, there's one completely automatic option. Generally it can be destructive to the sound, even using the most automatic setting - attenuation of bass, exaggerating of higher tones.

Preferred options while working with a bit muffled snare in the mix of 500m1 model for instrumental rap separation result

(automatic (easy master) is (only) good for mixtures [vocal+instr]):

- True Peak, Oversampling 2x, AM Level 0.3, WAV 32. SAO, 0/22000 (the rest untouched)

For still too muffled sound (e.g. when lost in lots of hi-hats):

- YouTube Loudness, OVS to 1x and AM Level 0.2 and 24 bit (+ true peak, SAO, 0/22000)

Alternative (good for mixtures and previously mastered music with a bit muddy snare):

- YouTube Loudness, Target Loudness -8, Ceiling -0.2, OVS to 2x, True Peak and AM Level 0.3 and 32 bit, SAO, 0/22000

The most complicated tool, but the most capable amongst all free ones mentioned here so far. After two first files, it gets you into a short queue. Processing takes 2-3 minutes. Cannot upload more tracks than one at the same time. Great metrics, e.g. one measuring overall "professionality" of the result master. At this point, it can also start exaggerating vocal leftovers from the separation process. Equalize Loudness doesn't do anything when checked just before download (probably only after when you click remaster).

They also have offline app: <https://github.com/ai-mastering/phaselimiter-gui/releases/> with some features used on Bakuage/aimastering.com

"but most of the settings you want are on their site, their offline version is set and forget. (...) doesn't give you some specific settings to adjust."

<https://moises.ai/>

16-32 bit WAV output (now WAV is only in premium), any input formats. They have bad separation tools, but great, neutral mastering AI. It works very good for vinyl rips. You can get more than 5 tracks per month for free (don't know how many - the 5 tracks limit is for separation, not for mastering feature, at least 30 worked in 2022).

The mastering feature is only available in the web version, so if you're on the phone, run the site in PC mode.

24 bit -9 iLUFS / or without limiter does the best job in most cases for e.g. GSEP (the latter is when you don't want to smooth out the sound). -8 tends to harm the dynamics of songs, but in some cases it might be useful to get your snare louder.

The interface has a bug when you need to pick your file to upload twice, otherwise you won't be able to change parameters and confirm the upload process (also on mobile parameters not always appear immediately after you pick your file/pasted link enlisting the options

manually doesn't let you confirm the step to proceed to upload, and you need to retry picking the file, and now you can proceed).

Sometimes uploading is stuck for very, very long on 99% and if you leave your phone in sleep mode and return after 15 minutes, it will start some upload again on this 99%, but eventually it will return the error. You simply need to retry uploading the file (it will also stack at 99%, but it will still upload at that time).

Also, importing the same file via GDrive may not work.

Additionally, if you pick 32 bit output quality, when mastering is done, when you will want to download the file, in WAV it will show 24 bit, but the file will be 32 bit as you selected before.

It's the most neutral in sound in comparison to the two below.

If you plan to master your own music, read "Preparing your tracks" here:

<https://moises.ai/blog/how-to-master-a-song-home/> I think these tips are pretty universal for all of these services.

<https://www.mastering.studio/>

Four presets with live preview, only 16 bit WAV for free, only WAV as input accepted (for the best quality convert any mp3's to WAV 32-bit float (you can use Foobar2000), 64 bit WAV input unsupported).

If you see "upload failed", register and activate a new account in incognito mode and everything using VPN (probably a block for ISP which I had).

Judging by only 16 bit output quality (which is unfair comparison to 24 bit on moises.ai) and for GSep 320kbps files, I found it worse, and even the London smooth preset is not so neutral like moises in overall, and it can be destructive to the sound quality. But, if you need to get something extra from the mix if it's blurry, that's a good choice (while some people can find emastered too pricy).

BandLab Assistant mastering

First, you need to download their assistant here:

<https://www.bandlab.com/products/desktop/assistant>

Then insert the file, pick preset, listen, and then it is uploaded for further processing, and you're redirected to the download page.

They write more about it below:

<https://www.bandlab.com/mastering>

Four presets - CD, enhance, bass boost, max 16 bit WAV output only. In comparison to paid emastered, it's average. But in some cases it's better than free mastering.studio when you have a muffled snare in the instrumental. On GSEP only CD preset was usable. The sound is more crusty than even LA Punch - more saturated (less neutral) a bit too bassy and compressed, but it may work in some songs where you don't have a better choice and all above failed.

If your file doesn't start uploading (hangs on "Preparing Master"), make sure you don't have "Set as a metered connection" option enabled in W10/11. If yes, disable it, and restart the assistant.

Straight after your file is done uploading, it is being processed, so don't bother going to BandLab site too fast - sometimes it's being processed even after download button appeared, where you start waiting in a queue even few minutes after you press the WAV button later, and you will not make this any faster.

On the side. The audio you hear during preview is not exactly the same as in result downloaded from the site. Preview is a bit louder, and stresses vocal residues more, and snare is less present in the mix, although the file is more clear, sadly it's also 16 bit, in overall it doesn't seem to be better. Also, the file doesn't seem to be stored locally anywhere. But if you're desperate enough to get this preview, fasten your seatbelt. If you processed more files before, close the assistant, and open again, now process the file, so preview can be played, pause it.

On Windows go to task manager, go to details, sort by CPU, RBM on BandLab Assistant.exe (the one with the most memory occupied)>Create dump file. Open it in HXD (located in temp), write in bytes per row instead of "16", "4000", find string "RIFF,". If you cannot find it, it's wrong process - make a dump of another assistant one (one of three most intensive). If you found the "RIFF," delete everything above it (mark everything dragging the mouse to the top, with page up pressed and then keep shift pressed and left arrow to mark also the first row, then press delete), then save it as wav. The file can be played, but it's too big. To find the end, go to "find" (CTRL+F), hex and write FF 00 00 02 00 01 00, find (it shouldn't be at the beginning of the file - press F3 even more than once if necessary), mark everything dragging the mouse to the top with page up pressed and press copy (CTRL+C) and paste it into new file and save as wav.

You can also use **Matchering**. It works in a way that you provide a reference file, and it tries to match the sound of your audio to the reference you provided.

Reference file(s) to use

"Mastering The Mix" (all-in-one collection of reference songs in one file):

https://drive.google.com/file/d/1kqPmcVC3qvh_Mqd9vIssGUKpz3jTddPc/view?usp=sharing

You need 7zip with WavPack plugin to extract it.

Brown/Pink noise:

https://drive.google.com/file/d/1wJHKRb2SIgJZlc-J8kEDD1k4OQj_OXzp/view?usp=sharing

"Try to use this as reference track in Matchering to get nice wide stereo and corrected high frequencies." zcooger

But you can use a whole song of your choice, or its short fragment (e.g. instrumental part to get better result of separation)

- New Colab:

<https://colab.research.google.com/github/kubinka0505/matchercli/blob/master/Documents/MatcherCLI.ipynb>

- Old Colab:

<https://discord.com/channels/708579735583588363/814405660325969942/842132388217618442>

- UVR5 (in Audio Tools) - incorporates Matching 2

- [Songmastr](#) can be used online instead of Colab, uses Matching 2 (7 free masters per week).

Be aware that there's a length limit in at least UVR5, and it's 14:44 (or possibly just 15 minutes). Instead of hit or miss by lots of reference files in one, you can also use simply one song you think will fit the most for your track. You can even further split it to a smaller fragment with e.g. lossless-cut in order to avoid reencoding. It can work even more efficiently that way.

Sometimes I use Matching for different master versions of the same song when I have a few masters I like certain things in them, but none good enough on their own.

Usually, in the target file should be placed the file with the richest spectrum (but feel free to experiment).

Can be a target file e.g. after a lot of spectral restoration, which e.g. lost some warmth and fidelity, and you need something from the previous master version.

You can also try to reprocess your result up to even 6 times, inputting a new file in target or reference each time, till you'll find the best result. But usually 2-3 should do the trick, sometimes while using target and reference interchangeably for different result files.

For using Matching in UVR5, necessarily check the option "Settings Test Mode" in additional settings. It will add a 10 digits number to each result, preventing you from overwriting your old files during multiple experiments conducted on your files. UVR doesn't ask before overwriting!

Feel free to experiment with WAV output quality. Probably the further you'll go from 24 bit, the more different your result will be after converting back to 16 bit by some lossy codec like Opus on YT. But if you care mostly about the result file, then simply be aware that you can use output quality to your advantage, knowing in what way specific bit depth affects output results. E.g. the muddier results start with PCM_32 (non-float), 64 bit has it too, but additionally with some grittiness. 16 bit is usually good to glue well sounding audio together with loudly sounding snares already, but can be muddy frequently or harsher than e.g. 32 bit non-float. Usually your result will be not so good in most cases, hence I'd encourage using higher bit depths than 16 bit here, but 24 bit can make your audio too bright at times, hence in such cases you can check 32 bit float and non-float. There's no simple setting working for

every song, but the most universal setting I found so far is using non-float 32 bit and convert it to 16-bit manually. It's the most balanced setting across the whole song (might be slightly too muddy at times).

Sometimes it can be good to have the richest file on a spectrogram as a target file, as it won't be lost after processing.

Matchering can be generally useful when you have different versions of your masters, and you're running in circles finding the best one. Then you can use such different versions as target and reference (or in reverse), check what sounds the best, get the result, use in one of the fields, retry, and the same up to 4 times till it sounds the best. Then you could potentially master it further and/or separate into stems and bring the session back from this place.

If you need more customizable settings for Matchering, e.g. controlling limiter intensity, or disabling it completely, consider using [ComfyUI-Matchering](#) (standalone/portable ComfyUI package for CPU or Nvidia: [new_ComfyUI_windows_portable_nvidia_cu124_or_cpu](#))

.masterknecht

<https://masterknecht.klangknecht.com/>

Web-based competitor of Matchering (it's not associated with Matchering). All the processing is done locally on your machine without uploading files to a server.

The results using default settings usually sound a bit softer/warmer to those from Matchering, output is 48kHz, plus there's much more customizable settings.

[EQ curve/master transfer](#)

Others

Windows app

<https://www.curioza.com/>

Some newer AI:

<https://huggingface.co/spaces/nateraw/deepafx-st>

Also try this one:

https://github.com/jhtonykoo/music_mixing_style_transfer

Or:

<https://github.com/joaomauricio5/AssistedSpectralRebalancePlugin>

ChatGPT

It can now master songs based on prompts and whatever you ask to make it sounds like. In the video below, the author wanted to make three instrumentals sound like Juice World. One result was decent, in the other one there was an issue with overcompressing/overlimiting, so

the guitar was fading in/out once some other instrument was kicking in. Some prompts might fail to give you the result file, and he provided the examples.

<https://www.youtube.com/watch?v=0kGJVgiyhAk>

"GPT does a lot of things right now, but the biggest problem is that it can't get larger files (wav) into the buffer and thus can't process them. Compared to unstable work results, not working is more serious." - tat_evop1us

<https://beatstorapon.com/ai-mastering> (only mp3 192kbps for free)

For enhancing 4 stems separations from Demucs/GSEP:

<https://github.com/interactiveaudiolab/MSG> (16kHz cutoff)

Platinum Notes

(Windows/Mac paid software)

"corrects pitch, improves volume and makes every file ready to play anywhere (...) add warm" and dynamics, remove clipping.

Mastering services I'm yet to test:

Landr, Aria, SoundCloud, Master Channel, Instant Mastering (iirc April fools joke), Bakuage, Mixea.

AI mixing services

<https://automix.roexaudio.com>

AI online auto-mixing service. Various instruments, genre settings, stem priority, pan priority. 1 free mix per month.

Might be useful for enhancing 4 stem separations.

"I tried 2 songs with it. Wasn't really pleased with results"

"The biggest problem I had [...] while I am trying to balance my vocals in instrumental like Hollywood style"

Other tool by Sony (open-source)

<https://github.com/sony/fxnorm-automix>

You can also train your own models using wet music data.

A new free tool by Sony:

<https://github.com/SonyResearch/MEGAMI>

<https://www.arxiv.org/abs/2511.08040>

HyMPS list of AI/CML tools

AI mixing plugins

iZotope Nectar

iZotope Neutron (Mix Assistant)

Sonible Pure Bundle

Creating mashups and also DJ sets (two options)

<https://rave.dj/mix>

It can give better results than manual mixes performed by some less experienced users (but I doubt it will work with more than 2 stems).

Ripple

iOS only app (currently for US region only)

"Ripple seems to be SpongeBand just translated into English, it was released last year:

<https://pandaily.com/bytedance-launches-music-creation-tool-sponge-band/>

(more info about its capabilities)

Back then, it only didn't have separation to 4 stems (but now the separation feature is defunct, anyway).

For enhancing vocal track you can use WSRGLOW, and better yet, process it through Izotope RX (7-9) spectral recovery tool (in RX 10 it's only in more expensive version irc), and then master it, or send it somewhere else above.

<https://replicate.com/lucataco/wsrglow>

There are a lot of requests for music upscaling on our Discord. You can use online mastering services as well. Technically it's not upscaling in most cases, but the result can be satisfactory at times.

If you try out all solutions, and learn how they work and sound, you can easily get any track in better quality in few minutes.

For very low resolution music (if you manage to run it):

AudioSR - used more often than the below, lately (voc/inst)

Audio Super Resolution

https://github.com/olvrhnn/audio_super_resolution

hifi-gan-bwe

<https://github.com/brentspell/hifi-gan-bwe/>

More details and links, Colabs for these in the upscaler's full [list](#)

If you want to start making your own remasters (even if your file is in terrible quality, especially 22kHz):

https://docs.google.com/document/d/1GLWvwNG5Ity2OpTe_HARHQxgwYuosseYcxpzViLwY/edit?usp=drivesdk

Might be useful also for low quality, crusty vocals, but it is also a guide for mixing music in overall but focused on audio restoration as well.

___Best quality on YouTube for your audio uploads___

- 1) If you already have a ready video which is not just a one frame (e.g. a cover all over the video), download MKVToolnix and replace audio track with lossless one instead of rendered lossy track. You will avoid recompression or reencoding, unlike it is during rendering normal video.
- 2) If you can, upscale the video to at least 1440p or greater. It will avoid deferred transitioning of your AAC (16kHz) audio stream to Opus (20kHz) when your video gets popular, or it's old enough (for current YT audio format, check statistics for nerds). QHD/+ makes your video play in better Opus codec from the beginning, and it will sound better than after deferred transition from AAC to Opus on FHD clip (Opus audio streams checksums differs in FHD and QHD videos despite the same video source file and most likely something is broken on YT side during the process, though both Opus files are 20kHz, so the file in FHD is not recompressed from AAC, perhaps from other audio file created during YT rendering, but not from the source video).
- 3) Alternative - if you have just one image to make a video of it (e.g. cover), make sure it's at least 1440p or greater. If not, simply upscale it (e.g. XnView has some basic upscaling filters). Then place the image nearby this batch [FFmpeg script](#) with your lossless audio files. It will render videos with the same audio streams like original files, but muxed into your output MKV files (you can check in Foobar2000 for Audio MD5 comparison or by using AudioMD5Checker, if MD5 checksum is not embedded when looking in F2K file properties) so it won't be recompressed on your end while making a video for upload on YT (yes, YT supports MKV!). It's faster than MKVToolnix and you can convert multiple files with the same image at the same time (it's very fast, incomparable to normal video rendering, and output is only 1 FPS, so it will buffer in YT also very fast).

- 4) You don't have to wait till YT stop processing your HD version for Opus to appear. It happens at a point when FHD resolution appears before QHD when processing is still in progress. So check it out from time to time before you hit the publish button.
- 5) Because Opus is 16 bit, and your input audio file in Matroska container might have higher bit depth, it's good to compress your input file to Opus VBR 128kbps for testing purposes to check how it will sound on YT (of course don't use it later for MKV file). Downsampling performed by the encoder can occasionally introduce some unwanted changes to the sound. It's the most noticeable when audio input is 64 bit, but smaller can be still good enough.
- 6) YouTube videos from early 2010 on archive.org have 192kbps AAC for 1080p ([example](#)) (thx theamogusguy)
- 7) If you deal with some harshness on your YT audio uploads with original 44kHz sample rate, consider resampling them manually to 48kHz before upload using e.g. Izotope RX (smooth) or e.g. dBpoweramp/SSRC (F2K plugin) and save the output as lossless format.
- 8) Since now MVSEP supports batch API conversions, you can use Case Changing in Ant Renamer to reestablish uppercase letters to song titles for your YouTube uploads.
So if you want to batch upload on YouTube the name will not appear as "song artist song title" but "Song Artist Song Title" (since YT removes dashes and commas etc.)
- 9) It seems like 720p is now enough to get Opus after upload (thx dca100fb8 - 9. & 8.)

Best quality from YouTube and Soundcloud - how to squeeze out the most from the music taken from YT for separation

Sometimes a better source just doesn't exist, and only YouTube audio can be used for separation in some cases.

Introduction

Audio on YT in most cases is available in two formats:

- 1) AAC (m4a) and Opus. As I mentioned, the latter appears for older or popular uploads, or videos uploaded in QHD or 4K. Most videos will have both formats available already. Currently only browsers without Opus support play that audio stream (iirc Safari)
AAC on YT is @128kbps with 16kHz cutoff and 44kHz (that's not artificial cutoff - that's how the codec normally behaves when such bitrate is set).
- 2) Opus on YT is 96/128/152kbps with 20kHz cutoff (spectrum up to 24kHz for videos uploaded before ~2020+, but only with some aliasing above 20kHz, probably as a result of applied resampler) always 48kHz (44kHz audio is always upsampled with built-in resampler in Opus - that's how the Opus works - it has always 48Khz output).

1) and 2) can be downloaded, e.g. via JDownloader 2 (once you downloaded one file, you must delete the previously shown entry in link grabber and add the link once more, and now pick the Opus (m4a is default) for download).

You can also use online too <https://cobalt.tools/> which is probably just GUI for yt-dlp.

Opus files downloaded from JDownloader are different than Opus in webm files seeing by spectrum, but I can't compare it with Cobalt as Spek doesn't cooperate with its webm files in at least progressive mode which is "direct vimeo stream". yt-dlp with -x argument might be free of the issue, but I haven't checked yet.

Don't download as Opus from JDownloader 2. The quality will be affected.

Download always as webm in any quality - all qualities will contain the same Opus audio stream in the same bitrate.

Be aware that sometimes JDownloader wrongly reports bitrate as 96kbps, while when you demux the webm file with MKVToolnix-GUI and then with MKVExtractGUI2 (compatible with MKVToolnix v 20), the result Opus file (add extension manually afterwards) will have average bitrate of not much below 128kbps (that's how VBR works).

Don't download in OGG from Cobalt. It's recompression from webm/Opus. OGG file is not on variants list in JDownloader (and probably the same would be in CML tools like yt-dlp, so it's simply not on YT).

However, it will have some additional information below 16kHz compared to Opus downloaded from JDownloader, probably because it was sourced from webm, and not JDownloader's Opus, but that's it. Recompression here will add some ringing issues and compression artefacts. Details and spectrograms [here](#).

Sometimes it happens that m4a (AAC) sounds better than Opus. It all depends on a track. It is more likely to happen if both have the same cutoff in spectrogram due to how it was uploaded on YT.

What to do to improve the audio gathered from YT?

#1 Joining frequencies with EQ method

- 1) Download both M4A and Opus audio from YT (if Opus is available for your video)
- 2) Upsample M4A to 48kHz (or else you won't align the two files perfectly) with e.g. Resampler (PPHS) in Ultra mode in Foobar 1.3.20>Convert>...
- 3) To have frequencies above 16kHz from Opus and better sounding frequencies up to 16kHz from AAC, we will combine the best of the both worlds by:
 - a) applying [resonant highpass](#) on Opus file at 15750Hz in e.g. Ozone 8/9 EQ
 - b) aligning the track to M4A audio file (converted to 48kHz WAV 32), so added as separate track in free DAWs like Audacity, Cakewalk, Ableton Lite, or Pro Tools Intro (or eventually Reaper with its infinite trial).

Export the mixdown as WAV24. It should be more than enough.
Using brickwall highpass instead will result in a hole in frequency in the result spectrogram (check it in Spek afterwards, and also whether there are no overlap frequencies in the crossover - consider checking also linear phase in e.g. free QRange EQ).

#2 Manual ensemble in UVR

Files ensemble with Max Spec in UVR

Instead of EQ, you can use ensemble after manual upsampling of M4A file. You can have your files aligned in UVR.

Be aware that this method is not fully transparent, and produce files a little bit brighter, and still with cutoff, but not brickwall like in M4A.

Without upsampling step, you can use Max Spec method with great results also for *Soundcloud* which provides 64kbit/s opus and 128kbps mp3 and 256kbps aac. You only need to amplify mp3 file by 3dB. Align step is also necessary here, but it can be performed in UVR.

(fixed in UVR 5.6) Be aware that a bug in manual ensemble exist which forces 16 bit output despite choosing e.g. 32-bit float. To fix it, you need to execute regular separation of a song with any AI model with 32 bit set, and then you need to return to manual ensemble without changing any settings now, so from now on it will retain 32-bit float in manual ensemble.

You can fix this by changing the 510th line of lib_v5/spec_utils.py to:

```
sf.write(save_path, normalize(output.T, is_normalization), samplerate,  
subtype='FLOAT')
```

then restart the program (you may not find that file if your UVR is not taken from source).

TBH, I didn't compare directly the first EQ vs the latter Max Spec method, but the latter sounds brighter for sure than opus, and m4a.

"while it helps to make trebles more defined, it's a bit flawed, due ensembling 3 different compression methods, so 3 different compression flaws/errors and noises".

PS. For YT I also tried downsampling Opus to 44 and to leave M4A intact, but it gave worse results (probably because of more frequencies affected by resampler in this case).

Explanation

Audio file sizes and bitrate are the same for both formats. Knowing that the cutoff in AAC is not artificial, but codec without a doubt efficiently compresses only audio up to 16kHz, leaving everything higher blank and untouched, we can come to the conclusion that frequencies up to 16kHz in AAC may sound better than in Opus,

since the size and bitrate of both files is the same, and most likely bitrate in AAC is not used to frequencies above 16kHz, so full 128kbps bitrate is used only for frequencies up to 16kHz in AAC codec while in Opus for the whole spectrum up to 20 or even 24kHz in some old videos till around 2020, while keeping the same size, so that might be more harmful for frequencies up to 16kHz than in AAC.

PS. After some time, I receive explanation/reassurance on the purpose of this process [here](#), saying it's generally justified and Opus is actually better than AAC even above 9600Hz, so one more additional cutoff in AAC will be needed. Also, might be worthy to use phase linear EQ to get rid of some coloration of the result file. Experimenting with it, make sure that you don't run into overlapping frequencies in area of bypassing (e.g. you can see it [here](#) as slightly brighter area above 9.6kHz up to 12kHz) to avoid it in e.g. in RX editor, one filtered signal needs to be 10Hz away from another one. I.e. if lowpass is 12000 Hz, then highpass is 12010 Hz. "But there is a catch with iZotope RX. The 10Hz away I described is only applied to the Copy operation (when you basically select the frequency range, and just CTRL+C by copying it). But there is also Silence operation (when you select freq. range and press Delete, it eliminates the freq. in this range), and it is another way around: you need to get the other signal 10Hz inward, so they overlap. I.e.: 12000 Hz lowpass, 11990 Hz highpass. Here is the video demo: <https://youtu.be/h5yE5cpqqMU>"

#3 Bash script to automate the AAC/Opus quality combining from YT audio

introC eventually wrote his bash script which makes an alignment (so trimming 1600 samples from m4a), performs cutoffs and joins frequencies of both files for you - without an overlap issue (tested with white noise). The script works for multiple m4a and webm files with the same name. Probably, MSYS2 (or cygwin) is required to run this script on Windows or for W10/11 use WSL ([read](#)).

He also took a more conservative approach here and changed cutoff frequency from 9600Hz to 1400Hz since AAC didn't perform better in one song, but below 1400Hz it will be rather in every case. What cutoff is actually the best might be sometimes depending on a song. The [script](#) is a subject to change.

#4 Method for better quality of instrumental leaks on YT by theamogusguy

"I did something really odd. (...) since you can only rip max 128kbps I did something really odd to get a higher quality instrumental:
I inverted the 128kbps AAC YouTube rip into the original to get the acapella
I took the subtracted acapella and ran it through AI (mel-roformer 2024.10) to reduce the compression artifacts
I then inverted the isolated acapella and mixed it with the lossless to get an... unusual lossless instrumental file?
also the OPUS stream goes up to 20khz but I feel like the sample rate difference is

gonna cause issues, so I ended up ripping AAC (OPUS is 48khz while most music is 44.1khz)"

_____Custom UVR models_____

Mostly outdated models, see [here](#) for more submissions from 2024

- 0) BubbleG — 15.06.2021
[Final drum model](#) (for UVR 5 and 4band_44100.json 4band_44100.json)
- 1) Dry Paint Dealer Undr — 08.07.2021
haring [wip piano model](#) trained on almost 300 songs might continue to train might not, has an issue where it also removes bass guitar too
- 2) BubbleG — 16.06.2021
[Temp. bass model](#). Must use with 4band_44100.json
- 3) viperx — 04.08.2021
My [simple karaoke model](#) that I trained in month 5 until epoch 25/28 doesn't complete the training because I've been busy with other projects, and I left this one aside, but this simple model removes the second voice, it can be useful in only some cases, it's bad but it's acceptable
- 4) [centre isolation model](#) epoch 0 inner epoch 1 - 150 pairs for UVR [4.0.1](#)
- 5) K-POP FILTERS — 02.07.2021
[model_0_0_1024_2048.pth](#)
feedback will be appreciated

Check [#model-sharing](#) for current WiP models

____Repository of old Colab notebooks____

UVR 5 (Colab by HV):

https://colab.research.google.com/github/NaJeongMo/Colaboratory-Notebook-for-Ultimate-Vocal-Remover/blob/main/Vocal%20Remover%205_arch.ipynb
(On Mobile Chrome use PC mode)

Alternative UVR 5 notebook up to date (not HV's):

https://colab.research.google.com/github/lucassantilli/UVR-Colab-GUI/blob/main/UVR_v5.ipynb#scrollTo=-KYA8iOZ8BKq

MDX (Colab by CyberWaifu, 4 stem, cannot be used in Mobile Chrome even using PC mode - there's no GDrive mounting and track downloading is always 0%. Model A cleaner but with more bleeding; Audioshake is based on it, but with different model based on larger dataset iirc, UVR team consider training it on their own bigger dataset to get better results - it's based on phase unlike UVR, but tsumeruso works on adding phase, so then it might get rewritten to UVR)

https://colab.research.google.com/drive/1R32s9M50tn_TRUGIkfnjNPYdbUvQOcfh?usp=sharing

(wait patiently, it doesn't show the progress)

UVR 5 (old version by HV with any 2 files ensemble feature, put tracks in separated folder. As for x/z - similar results, but not the same. Put as first the one you want the result more similar to)

<https://colab.research.google.com/drive/1eK4h-13SmbjwYPecW2-PdMoEbJcpqzDt?usp=sharing>

https://colab.research.google.com/drive/1C6i_6pBRjdbueVw27FuRpXmEe442n4k?usp=sharing#scrollTo=CT8TuXWLBrXF (+12 ens, no batch ens, deleted)

2021-ISMIR-MSS-Challenge-CWS-PResUNet (byteMSS) (if you run out of memory, split up the input file)

https://colab.research.google.com/drive/17m08bvihZAov_F_6Rg3luNj030t6mtyk?usp=sharing

Woosung Choi's ISMIR 2020 (Colab by CyberWaifu)

<https://colab.research.google.com/drive/1jlwVgC9sRCGnZAKZTpqKgeSnzP3slj8U>

Vocal Remover 4:

<https://colab.research.google.com/drive/1z0YBPfSexb4E7mhNz9LJP4Kfz3AvHf32>

To fix librosa error, try adding the

!pip install librosa==0.8.0

or 0.9.? works as well

line about librosa, and if still the same, about pysound as well:

<https://discord.com/channels/708579735583588363/767947630403387393/108951896325317652>

https://colab.research.google.com/github/burntscarr/vocal-remover/blob/main/vocal_remover_burnt.ipynb

(UVR4 + models description:

<https://github.com/Anjok07/ultimatevocalremovergui/tree/v4.0.1>

Search for:

"Models included" at the bottom".)

UVR 2.20 (it achieved some good results for old 70's pop music for me where cymbals got muffled on current models, but prepare for more bleeding in some places vs VR4 and newer)

<https://colab.research.google.com/drive/1gGtjAo3jK3nmHcMYTz0p8Qs8rZu8Lhb6?usp=sharing>

Spleeter (11/16kHz, 2, 4, 5 stems, currently doesn't work):

<https://colab.research.google.com/drive/1d-NKFQVRGCV5tvbd0GOy9spMMel6mrth?usp=sharing>

According to my experience, if you don't need piano stem, 4 stem model makes better job than 5 stem (and even vs 2 stem, and it is also reflected in SDR results). Use 11kHz models only if your input files are sampled at 22kHz (it will provide better result in this and only in this case).

If you can, use Izotope RX-8 for 22kHz 4 stem, as it provides better separation quality with aggressiveness option. It's Spleeter, but with better model (full band).

Demucs 3.0

https://colab.research.google.com/drive/1yyEe0m8t5b3i9FQkCI_iy6c9maF2brGx?usp=sharing

To install it locally (by britneyjbitch):

I cracked the Da Vinci code on how to install Demucs V3 sweat_smile For anybody who struggled (on Windows) - I got you!

1. DL a zip folder of Demucs 3 from Github (link:

<https://github.com/facebookresearch/demucs>) and extract it in a desired folder

2. Inside the extracted folder run cmd

3. If you want to simply separate tracks, run the following command:

python.exe -m pip install --requirement requirements_minimal.txt

4. If you want to be able to train models too, run the following command:

python.exe -m pip install --requirement requirements.txt

5. If a read error for incompatible versions of any of the modules appears (e.g. torch) run the following command:

pip install desired_module==version_of_desired_module

e.g. pip install torch==1.9.0

6. Repeat step 5 for any incompatibilities that might occur

7. Separating tracks:

python.exe -m demucs -n "desired_model_to_run_separation" "path_to_track"

8. If you want help finding all additional options (for example overlap or shifts), run:

```
python.exe -m demucs --help
```

At least that worked for me, feel free to let me know if this worked for others as well
exclamation Oh, and I forgot - between step 6 or 7, don't pay attention to a potential red error
"torchvision 0.9.1+cu111 has requirement torch==1.8.1, but you'll have torch 1.9.0 which is
incompatible."

Do NOT change back to torch 1.8.0 cuz you won't be able to run demucs
warning! If "torchvision 0.9.1+cu111 has requirement torch==1.8.1, but you'll have torch 1.9.0
which is incompatible." is the only red error you're getting after executing the commands
from step 3,4 and/or 5, you're good to go with separation!

Demucs (22khz, 4 stem):

<https://colab.research.google.com/drive/1gRGRDhx9yA1KtafKhOaXZUpUoh2MuF8?usp=sharing>

<https://colab.research.google.com/github/facebookresearch/demucs/blob/master/Demucs.ipynb>

https://colab.research.google.com/drive/1gRGRDhx9yA1KtafKhOaXZUpUoh2MuF_8?usp=sharing

Other one(s):

LaSAFT:

https://colab.research.google.com/drive/1XInqzXDi2mF_y6WwDrLLx4XZtl8_1FAz?usp=sharing

(original, cannot define model ATM)

[https://github.com/ws-choi/Conditioned-Source-Separation-LaSAFT/blob/main/colab_demo/LaSAFT_with_GPoCM_\(large\)_Stella_Jang_Example.ipynb](https://github.com/ws-choi/Conditioned-Source-Separation-LaSAFT/blob/main/colab_demo/LaSAFT_with_GPoCM_(large)_Stella_Jang_Example.ipynb)

If you cannot load the file, upload it manually to your Colab, or just wait patiently. Refresh Github page with CTRL+R if you can't see the code preview.

Check out also this LaSAFT [download](#) with [message](#) which says about superiority of 2020 model (said in march 2021).

Clone voice:

<https://colab.research.google.com/github/tugstugi/dl-colab-notebooks/blob/master/notebooks/RealTimeVoiceCloning.ipynb>

Matchering:

https://cdn.discordapp.com/attachments/814405660325969942/842133128851750952/Matc_heringColabSimplified.ipynb

For more Colabs search for colab.research.google.com on our Discord server

Google Colab troubleshooting (old)

- *Error of authorisation during mounting:*

TL:DR - you need to log into the same account in Colab you want to mount drive later, or just change your Colab account.

It was introduced to Colab at some point. Once I tried to log into another account during mounting, it displayed a new window with only one account, where the wanted account didn't appear, and when I manually signed in to it, Colab showed an error on Colab, something about unsuccessful authorisation. When I changed account in the right corner this time for the same account I wanted to choose when mounting, everything went fine as it always used to be. Full list of accounts appeared. HV Colabs already have the new mount method implemented, so the old one doesn't cause error, but in UVR notebook you can choose between the new (default) and the old one (just in case Google changed something again).

- Try to log into another Google account(s) if you cannot connect with GPU anymore and/or you *exceeded your GPU limit*
- (cannot really say if it's really helpful at this point)
Paste this code to console (Chrome: CTRL+Shift+I or ...>more tools>tools for developers>console) to avoid disconnections from runtime environment or if you encounter problems while being AFK and if you run into issues of being unable to connect to GPU after reconnection after idle time or possibly after the code was executed, and you're AFK for too long. It won't prevent you from showing one captcha in the session.

```
interval = setInterval(function() {  
    console.log("working")  
    var selector = "#top-toolbar > colab-connect-button"  
    document.querySelector(selector).shadowRoot.querySelector("#connect").click()  
    setTimeout(function() {  
        document.querySelector(selector).shadowRoot.querySelector("#connect").click()  
    }, 1000)  
}, 60*1000)
```

It will constantly reclick one window to appear in Colab to prevent idle check.

Repository of stems/multitracks from music to create your own dataset

Datasets search engine

<https://datasetsearch.research.google.com/>

Up-to-date list of datasets

<https://github.com/Yuan-ManX/ai-audio-datasets-list#music>

33 datasets compilation list:

<https://sites.google.com/site/shinnosuketakamichi/publication/corpus>

ZFTurbo's list (contains duplicates from below):

<https://github.com/ZFTurbo/Music-Source-Separation-Training/issues/40>

Check out also:

[#resources](#) | [#datasets \(invite\)](#)

musdb18-hq (for described errors in the repo [read](#))

[\(14GB 7z\)](https://drive.google.com/file/d/1ieGcVPPfgWg__BTDIIGi1TpntdOWwwdn/view?usp=sharing)

[\(mirror, 22GB zip, it can be slow at times\)](https://zenodo.org/record/3338373#.Yr2x0aQ9eyU)

Slakh2100 (2100 tracks), mono, guitar + piano, and a LOT of other stems, no vocals

If we were to ever train a multiple-source Demucs model, it would be greatly helpful

https://drive.google.com/file/d/1baMOSgbqogexZ5VDFsq3X6hgnIpt_bPw/view

<https://github.com/ethman/slakh-utils>

<https://drive.google.com/file/d/1sxdNk0kekvv8FwDvzNypYe6Nf7d40Iek/view?usp=drivesdk>

Jammit ([torrent](#))

"the audio files can't be mixed directly. You need to apply a gain reduction of 0.77499997615814209 (in dB : -2.2139662170837942) on each track to get a perfect mixdown. This factor is about to set a 0dB on the original jammit mixtable."

MoisesDB

<https://music.ai/blog/news/introducing-moisesdb-the-ultimate-multitrack-dataset-for-source-separation-beyond-4-stems/>

"Total tracks: 240

How often folders exists for track: ('vocals', 239), ('drums', 238), ('bass', 236), ('guitar', 222), ('other_keys', 110), ('piano', 110), ('percussion', 99), ('bowed_strings', 45), ('other', 39), ('wind', 26), ('other_plucked', 7)"

Script to convert MoisesDB in MusDB18 format:

For help and discussion, visit our Audio Separation Discord: <https://discord.gg/ZPtAU5R6rP> | Download [UVR](#) or [MSST-GUI](#)

For inst/voc separation in cloud, try out free Colabs: [BS/Mel-Reformer](#) | [MDX23](#) (2-4 stems) | [MDX-Net](#) | [VR](#) | [Demucs 4](#) (2-6)

<https://gist.github.com/kiselecheck/df62174c5d986afcc5875300fd38bf9a>

Cambridge Multitrack Library

<https://multitracksearch.cambridge-mt.com/ms-mtk-search.htm>

A nice collection of legally available multitracks.

"I believe about 2/3rds of musdb18's tracks are taken from this."

Great for dataset for creating stem specific models like acoustic guitars, electric guitars, piano, etc. You will just get the stem file you want and combine the rest

DAMP-VSEP

<https://zenodo.org/record/3553059>

Smule Digital Archive of Mobile Performances - Vocal Separation

seems to be a really big dataset of instrumental-amateur vocal-mix with compression and such triplets.

Metapop

<https://metapop.com/competitions?p=1&status=ended&type=all>

"Most of them have a click through to download stems. You might need to automate downloads using Simple Mass Downloader browser extension or something. Some are remix competitions, some a production, but all have stems."

Guitar Hero / Rockband stems

remixpacks.ru / remixpacks.club (taken down, now it's under <https://remixpacks.net/> address [not sure if the site content is the same])

[Python script](#) by MissAllure for downloading stems from:

<https://docs.google.com/spreadsheets/d/1BtUSqPffbcaW4bMuGCIYi8FGvaYmYyc1p4SkfpNtyU/edit?gid=0#gid=0> (only 10, but you can change it; saves you from having to open links; written by AI)

Remix packs master post (removed dead links) - still has like 2000+ stems

https://drive.google.com/file/d/11NrEIQSjrXT_DbTL00r9OeMrrEBral3V/view?usp=sharing

Torrent:

or here:

magnet:?xt=urn:btih:45a805dbd78b8dec796a0a127c4b4d2466ddbb9a

(list with names:

<https://docs.google.com/spreadsheets/d/1uCWmuAUfvVLonbXp9sQUb9dEODYTHmPAOyvGxuIMOCA/edit?usp=sharing>

Renamer - python script

<https://mega.nz/file/gEgwwaaB#BCDDMpI-VcIZDnNYQziykIOV9Vpf43wuc76hsS3JTIw>

Showcase

<https://www.youtube.com/watch?v=95Q31HjU04E>

Archive.org copy

<https://web.archive.org/web/20230105142738/https://telegra.ph/Remixpacks-Collection-Vol-0-1-04-12-25>

Or here (but you can't access all the sections at the bottom and after some time you get "Unable to load" error; probably using the old Manifest uBlock with blocking specific site element would work, not sure):

https://web.archive.org/web/20230521064118/https://docs.google.com/spreadsheets/d/1_dIFNK3LC8A40YK-qCEHhxOCFlbny7Jv4qPEoOKBrIA/edit

(separate downloads)

OG subreddit source along with the file was deleted, and back when it was online, probably it was locked from downloading and scrapping it was difficult.

Q:

<https://web.archive.org/web/20230105142738/https://telegra.ph/Remixpacks-Collection-Vol-0-1-04-12-25> contents list? I don't want to download all of them just to find one thing (genie in a bottle stems)

A:

<https://docs.google.com/spreadsheets/d/1eN2-I0OBD3R8AHRGjKuHpxTHbevYi0kg1O7zJZHyvIY/edit?usp=drivesdk>

If there are no seeds, so the torrent is dead, "a major part of these stems are on the songstems telegram chat, including new stems that aren't in these packs"

<https://t.me/+mrluHEcfixwwNzRk>

"For those that aren't able to d/l the torrents anymore, or just want to d/l some of the remixpacks content,

I uploaded all 26 collections (~3TB) here: <https://remixpacks.multimedia.workers.dev/>
DM me to request username/password." Bas Curtiz#5667

<https://clubremixer.com/> - outrageously big database, probably reuploads from remixpacks too (but on slow Nitroflare or simply paid irc)

<https://songstems.net/> - lots of remixpacks stuff reuploaded from masterposts of clubremixer.com to Yandex (free Nitroflare is 20KB/s)

~~Mega collection of stems/multitracks (remixpacks - Guitar Hero, Rock Band, OG)~~

https://docs.google.com/spreadsheets/d/1_dIFNK3LC8A40YK-qCEHhxOCFlbny7Jv4qPEoOKBrIA/edit

Rock Band 4 stems (free Nitroflare mirror)

<https://clubremixer.com/rb4-stems/>

Different mixing of the RB tracks was a factor in models trained by the community. "Also, RB tracks never fade out. They are also never brickwalled."

"brickwall audio has negative influence on waveform based archs, but on spectrogram based one like all recent ones, it doesn't seem to have big impact on results quality" - jarredou

GH stems from X360 instead of Wii for better quality

<https://www.fretsonfire.org/forums/viewtopic.php?f=5&t=57010&sid=3917a8e390f65097f07d69595dd5ba55>

(free registration required, basically content of all zippyshare links of the PDF below:)

PDF with separate RB3-4 stems description and DL (lots of links are offline as zippyshare is down), page 6 shows some table of content with evaluation progress.

toaz.info-stemspdf-pr_7a1e446f01c9b1666a9bebe9fd51f419.pdf (reupload)

Huge database (probably contains some of the above)

<https://songstems.net/>

Others:

Multitracks' section of rutracker (requires free account):

<https://rutracker.org/forum/tracker.php?f=2492>

Multitracks/multitrack queries on The Pirate Bay

<https://thepiratebay.org/search.php?q=multitrack&all=on&search=Pirate+Search&page=0&orderby=>

<https://thepiratebay.org/search.php?q=multitracks&all=on&search=Pirate+Search&page=0&orderby=>

"You can just go here <https://rutracker.org/forum/tracker.php?f=1674> (sample libraries category) and type the instrument you want, it will pop all the sample packs.

Maybe add "loop" to the search too, will filter out some weird packs"

Maybe you find something useful on sharemania.us too (160 lossy/261 lossless)

Seems 'acapella tools' or 'instrumental tools' are good key-words to search for.

Some are covers of original tracks, but that shouldn't matter, since they represent the same. This is on Deezer, but you might find others on Tidal.

There's also some stuff available on Soulseek (P2P service)

frp.live instrumentals/acapellas

<https://docs.google.com/spreadsheets/d/1NuQV8cfFPehvlwPBUGOMbiC4FSei2p923qC6af5tCV8/>

22 instrumental albums and some single tracks ([DL](#)) - hard to align for inversion, even for lossless, sometimes time shifts every verse, possible artefacts/bleeding after inversion to be cleaned further with [models](#).

127 hip-hop instrumentals with vocal chops (duplicates from the above), and 80 with scratches or harmonies ([DL](#))

Giliaan stems for 4 songs (EDM/Dance/House) and Mainstream Dataset with 20 songs:

https://drive.google.com/drive/folders/1JbQRMYH9DT_vUHpf4jHwD80eC6VvpDZX

Mirror (with messages below)

<https://discord.com/channels/708579735583588363/1286052299931652106/1304884347492372580>

50 Produce Like a Pro multitracks

<https://producelikeapro.com/blog/happy-new-year-2022-3/>

<https://producelikeapro.ipages.co/keep-truckin-multitracks-form/>

Potentially more:

<https://www.youtube.com/playlist?list=PLnLOmVwRMCqS1ia3o9Vv0nG5sMgcFR9Tc>

Sites:

<https://promodj.com/tools>

There is a lot of filtered trash, but you can also find official acapellas.

<https://www.acapellas4u.co.uk/>

Collection of 40K instrumentals and accapellas (lossy, rather avoid using such files for training, and search for lossless if possible)

<https://isolated-tracks.com/>

Multitracks. Looks like paid, but it has also few pages with some free ones (e.g. Fleetwood Mac, not sure if free)

<https://www.multitracks.com/>

This is also paid, but it has less known music

“those are covers from famous songs, but all in multitracks.

And from what I've listened to so far, is that they are pretty conservative.

The vocals all seem to be dry and none seem to contain bleed so far.

Also, the instrument stems are proper / not mixed up with other instrumentals.

The stems are the exact same duration.

All in all, a solid dataset right off-the-bat imo.

I should've calculated it prior, what the better subscription was, the 10GB or 20GB a day one vs. price vs. content approx. in total.

52mb (wav) * 12 (multitracks) = 624mb per song

4.766 songs * 624 = 2973984 mb = 2.97tb

weekly limit = 70gb * 4 (weeks) = 280gb = 280000mb

2973984 / 280000 = 10,6 weeks in total.

10,6 / 4 = 2,65 so 3 months x \$30 = 90 bucks"

<https://www.epidemicsound.com/music/search/>

Can be ripped. Some tracks there will be a subject to rule out due to bleeding. Plenty of genres. Might be good for diverse dataset.

<https://bleep.com/stream/stems>

Looks like official stems for sale. ~45 songs in total.

FullISOL (only for premium users; min. 200EU for year)

<https://forum.ircam.fr/projects/detail/fullsol/>

19,91 GB of audio samples. No percussion.

Instruments: Bass Tuba, Horn, Trombone, Trumpet in C, Accordion, Harp, Guitar, Violin, Viola, Violoncello, Contrabass, Bassoon, Bb Clarinet, Flute, Oboe, Alto Saxophone
(jarredou have it)

Vocals/speech

MedleyVox dataset (for separating different singers) of which they refrain from releasing the model for (and Cyrus eventually did it single-handedly):

<https://github.com/CBeast25/MedleyVox> (13 different singing datasets of 400 hours and 460 hours of LibriSpeech data for training)

<https://zenodo.org/record/7984549>

k_multisinger:

<https://drive.google.com/file/d/18evyY82ec4IdNT2z8q76zm30EWhfc-9j/view?usp=sharing>

k_multitimbre / K_multitembre:

<https://drive.google.com/file/d/1lc4P8gCGwbLshR118N8V3tbAU-D9Us-i/view?usp=sharing>

Potentially more here:

<https://sites.google.com/site/shinnosuketakamichi/home>

Be aware that the only one MedleyVox dataset which remains unobtainable to this day is TONAS, but it's small, esp. compare to the Korean datasets. Besides this one, queer and Cyrus have them all on our Discord, but they're huge. Ksinger and Ktimbre takes ~300GB unzipped for both.

- Jarredou's (@rigo2) dataset with screaming, cheering, applause, whistling, mumble, etc... collected from all the sources I've found, to help model creation:

+5000 stereo wav files, 44100hz

~37 hours of audio data

- "Ultimate **laugh** tracks for sitcoms, game shows, talk shows, and comedy projects (available on Amazon Music and Apple Music ([ripped](#), YT upload has similarly looking spectrograms)

- Laughter-Crowd Dataset #2.zip https://terabox.com/s/1xLuZWvpGX0LTQypO1p7u_g

<https://multitracks.pages.dev/> (only a list, no DL links)

English and Spanish multitracks

- Around 30GB of T.Swft stems ([1](#) (not necessarily mirror) | [2](#))

- There is 768.44 GB of K-pop stems somewhere in the wild (maybe ask .mikeyyyyy)

- Gabox karaoke dataset (2GB)

<https://gofile.io/d/TyzaH8>

"(may need a check, iirc there were songs without bv, also it doesn't have the vocals part)"

- RawStems

<https://huggingface.co/datasets/yongyizang/RawStems> (DL)

<https://github.com/yongyizang/music-source-restoration/blob/main/preprint.pdf> (paper)

"A dataset annotation of 578 songs with unprocessed source signals organized into 8 primary and 17 secondary instrument groups, totaling 354.13 hours. To the best of our knowledge, RawStems is the first dataset that contains unprocessed music stems with hierarchical categories".

- *Expressive Anechoic Recordings of Speech (EARS) dataset.*

- **100 h** of speech data from **107 speakers**
- high-quality recordings at **48 kHz** in an anechoic chamber
- **high speaker diversity** with speakers from different ethnicities and age range from 18 to 75 years
- **full dynamic range** of human speech, ranging from whispering to yelling
- 18 minutes of **freeform monologues** per speaker
- sentence reading in **7 different reading styles** (regular, loud, whisper, high pitch, low pitch, fast, slow)
- emotional reading and freeform tasks covering **22 different emotions** for each speaker

https://github.com/facebookresearch/ears_dataset

DL:

[1](#) | [2](#) | [3](#) | [4](#) | [5](#) | [6](#) | [7](#) “The dataset is made of 107 zip files that you can download one by one manually”

“What is great with this dataset is that it was recorded in anechoic chamber, so no reverb, no echo, with high-end hardware. You can use it as baseline for reverb removal, speech enhancing, etc...”
jarredou

- Metal dataset

<https://zenodo.org/records/8406322>

760 audio excerpts from 1 to 30 seconds in mono.

“iirc, the audio samples can be very short, it may need pre-processing (merging multiple samples in 1 file) to be used for training”- jarredou

I think mesk did the job already for his dataset (it's uploaded later below).

Around a hundred of Eminem's acapellas leaked:

<https://drive.google.com/drive/folders/141t33Qa2h3rEi2T0lYokvBPiiDBbC6dQ>

Official and unofficial Eminem instrumentals (single links):

https://docs.google.com/spreadsheets/d/1x9tTOOqH5WpKOoptdQzABSN_x8oZbMqzIGIGH9w1IKA/edit?gid=965054462#gid=965054462

Piano

“**MAESTRO**” is a dataset composed of about 200 hours of virtuosic piano performances captured with fine alignment (~3 ms) between note labels and audio waveforms.

<https://magenta.tensorflow.org/datasets/maestro>

GiantMIDI-Piano is a classical piano MIDI dataset contains 10,855 MIDI files of 2,786 composers. The curated subset by constraining composer surnames contains 7,236 MIDI files of 1,787 composers. GiantMIDI-Piano are transcribed from live recordings with a high-resolution piano transcription system

<https://github.com/bytedance/GiantMIDI-Piano>

SFX

Datasets for potential SFX separation

- <https://cocktail-fork.github.io/> (SPEECH-VOICE-SFX (3 stems), 174GB)
- <https://www.sounds-resource.com/>
- <https://mixkit.co/free-sound-effects/game/>
- <https://opengameart.org/content/library-of-game-sounds>

- <https://pixabay.com/sound-effects/search/game/>
- https://www.boomlibrary.com/shop/?swoof=1&pa_producttype=free-sound-effects
- Spongebob stems (500MB)
<https://drive.usercontent.google.com/download?id=1P19Diyw7CRteqeLs0beDpCaexFyZJiDs&export=download&authuser=0>
- Nickelodeon leak (2024 Nick Giga Leak7.zip/nick.7z) (10.7GB)
<https://myrient.erista.me/files/Miscellaneous/Nickelodeon%20Leaks/>
(not a full leak, as it has 500GB and only some people have it)
- Sound effects HQ by soniss 2024 (27.5GB+)
<https://gdc.soniss.com/>
- “Free sound FX samples packs from Adobe”:
<https://www.adobe.com/products/audition/offers/adobeauditionlcsfx.html>

Drums

[StemGMD: A Large-Scale Audio Dataset of Isolated Drum Stems for Deep Drums Demixing](#)
(although drumsep used bigger dataset consisting of MIDI sounds to avoid bleeding, with XLN only)

Virtual drumkits

The “advantage is that you can have zero bleed between elements, which is not possible with real live drums.

You can create “more than 300 drumkits as virtual instruments (toontrack, kontakt, xln, slate, bfd, XLN ones are nice too (from their trigger and drums VST) + a Reaper framework to multiply that by 10 (using heavily different mixing processes for each drum elements), so potentially 3000 different sounding drumkits “
“one could use producer sample packs/kits for more modern samples” there are tons of packs around the net.

jarredou (rigo2):

“For those interested, I'm sharing on demand my drums separation dataset.
It's not a final version. I've realised after generating 130h of audio data that I've made a mistake in routing, leading to some occasional cowbell in snare stems. So it's [Kick/Snare-cowbell/Toms/HiHat/Ride/Crash] stems.
I've stopped it's rendering and will not make the final "mastering" stage that was planned.

I will make a clean no-cowbell version, but as I'm lacking free time, I don't know when, and as this one is here and already great sounding why not using it in the meantime.

Just don't mind the cowbell!"

Looks like it's the thing:

<http://rigaudio.fr/datasets/DrumsDataset.zip>

Newer version in a better formatted version, with train/valid separated parts, generated mixtures and a fixed filename that was containing an extra space:

<https://rigaudio.fr/datasets/DrumsDatasetv2.zip> (25GB)

(still the same issues with some occasional cowbell in snare stems. "There are also few other percussions here and there on some little parts for some tracks (like tambourine in ride stem).")

"I realise now that I totally forgot to lowercase all filenames before reuploading the dataset.

To avoid issues where some awaited filenames are hardcoded in ZFTurbo's script, the best way is so to lowercase all filenames in train/valid parts, convert the valid part to .wav files (no need for the train part that can handle flac correctly).

And lowercase the stem names in training part of the config file accordingly."

"Can probably be useful to create electro drums separation dataset, free 50,000 drums MIDI files:"

<https://abasynthphony.gumroad.com/l/50000MIDIFilesforDanceMusicDrum?layout=profile>

mesk's metal drums dataset (drums in one stem):

(dead) <https://drive.proton.me/urls/5PGCB22KKC#0koU5OEx71f4>

"resharing my metal dataset for people to claw their hands on

https://drive.google.com/drive/folders/1ajlzmyAuX-fsiKiaypN8y2GT8EYBAws5?usp=drive_link

this consists of:

official instrumentals, straight from my stems, what's not labelled with my name are official as well [remixpacks stuff were curated]

vocal folder has official vocals (from the stems again), remixpacks (curated), inverted vocals and some weird whispery sh!t from yours truly"

<https://www.monotostereo.info/>

"Helped me find not only tools but also other resources like research papers, etc on audio source separation in general. A fantastic resource for anyone into audio source separation"

DnR

Divide and Remaster v3: Multilingual Validation & Test Set

<https://zenodo.org/records/12658755>

"but there are a bunch of other versions of v3 for specific languages, I'm not sure what is the difference.

There are separate versions for English, Spanish, French, etc.

<https://zenodo.org/communities/opencass/records?q=&l=list&p=1&s=10&sort=newest=>

In fact, you can train on validation too, but it's not necessary anymore as the dataset was published already:

<https://github.com/kwatcharasupat/divide-and-remaster-v3/wiki/Getting-the-Dataset>

Rhythm and lead guitar

<https://www.mrtabs.com/>

"He has isolated tracks for his videos that he makes, and it's free (or he does not know how to properly Patreon lock certain content on his website).

You can navigate to any tab page and look for the header: "Isolated TRACKS (mp3)" and find the textbox below where it says:

"Please sign up on Patreon, or if you are already a member, please login."

The last word links to a Patreon signup page, and if you sign in, it does not check if you are subscribed to his Patreon or not, it will give you access regardless.

Boom! Now you have access to 250+ Lead and rhythm guitar pairs. There is a goldmine worth of metal stuff in there too.

This is probably the closes we could ever get to having a contemporary rhythm/lead guitar dataset that is both relatively large, the rhythm and lead has their own tracks, its diverse, and actually includes songs that we like/listen to.

Only problem is all of them are exported in mp3 with a cutoff of 16khz, so it is equivalent 128 kbps, and the denoising that was done in post is pretty lazy.

However, I think if these parts are upscaled with FlashSR it would be great.

Or maybe Re-Amp a low pass filter version of the stems with the Ampltibe 5 presets that he also attaches to all tabs, and ensemble the remaining frequencies that way.

I personally would not recommend using the drum and bass stems, only the guitar parts, since the drum and bass are both programmed and are uniform.

The tone for every video is unique and tone matched to their respective albums and tracks. Even if it's not dead on, it's better than trying to use some yayhoos guitar doodles that uses the same amp/cabinet/simulator for every track." Vinctekan

Ernhu

[China traditional music instrument dataset]

<https://zenodo.org/records/8012071>

Instrumentals/vocals/stems

- Metal genre dataset

Contact @33meskvlla33 (iirc 2K unique songs)

Here is a smaller version of the metal dataset + the validation dataset (there is also not metal in there, but lots of the data is metal oriented)

302 vocals + 802 instrumentals

https://drive.google.com/drive/folders/1TIY1FXP54sVA9T0Kfq0oOXJXq03czxrv?usp=drive_link

If anybody wants to train an instrumental/vocal model on metal, this can get you started (I'm severely limited by my hardware).

A lot of the instrumentals are official instrumental versions of albums

the stuff with my username is from my stempacks except for Omega Virus, Behold the Void and Rings of Saturn (IDK why I named these with that xd)

- Index of ~7,5K songs in multitracks in the wild - 13.03.2023 (updated link above)

<https://krakenfiles.com/view/XiDE82aLOR/file.html>

No download links. Probably some will be available around the net if you search well.

- Here's a magnet link with some stems:

<https://web.archive.org/web/20200606113408/https://pastebin.com/6bZtpvur>

- "From the 90s hit maker Moby himself, 500 multitracks (unreleased songs, copyright free):

<https://mobygratis.com/>

- Official accapellas, instrumentals and stems

<https://infinity101.wolf.usbx.me/filebrowser/share/Q9HHIUB6>

- "beatmania the rhythm game makes charts very interestingly because they are all keysounded, but what's interesting is that someone made a chart to reaper project converter and essentially it just gives you stems. I think people could probably export a shit ton of electronic stems and improve models because there are a LOT of bms charts" [src](#)

- [Songstems.net](#) Telegram group where you might find some music stems

<https://t.me/+mrluHEcfiwwNzRk>

- Lots of instrumentals (sometimes with backing vocals) - [click](#)

- The Spheres Dataset
(orchestral)
<https://zenodo.org/records/17347681>

For more links, check [#resources](#) and [#datasets](#) and [Post dataset](#) (you may encounter duplicates)

List of cloud services with a lot of space or for temporary storage

Unlimited

<https://filegarden.com/>

No info on any limits, URL shortener, browser bar player, open [source](#), registration required.

Unlimited

<https://imgur.gq/>

500MB/file and 5GB/file for registered users, no expiration, no registration required.
Audio files previewing.

Unlimited

<https://krakenfiles.com/>

1GB/file and 5GB/file for premium users, no registration required.
Audio files previewing.

You can accidentally download a virus without having any adblocker, esp. on iOS - be aware ([example](#)).

Unlimited

<https://pillows.su/>

200MB/file and 500MB/file for registered

Perhaps not the best solution out of all. "It had issues during December and January 2025" it might have a problem with uploading not working randomly. Even in August 2025 someone had issues with accessing pillowcase uploads.

Only for audio files (also zip for registered users) - allows playing audio files, shows spectrograms. I've met with a case when some files uploaded one by one couldn't be played or downloaded (HTTP 500 error) at least after some period. For at least 503 error, it was enough to reload on a page the error appeared on to start the download.

(formerly pillowcase.su, thx Nick088)

?Unlimited

<https://vocaroo.com/upload>

Not sure on file size limit or expiration

For audio files only, no registration option

Unlimited

<https://buzzheavier.com/>

(mirrors: <https://trashbytes.net/> / <https://flashbang.sh/>)

Files kept “forever”. Almost 600 Mbit/s of upload speed.

It optionally creates not only download link, but also torrent file and seeds it. Optional expiration (with also “never”). “The owner doesn't give a damn about DMCA.” On unstable connections, it might break the download in the middle, forcing to start from scratch. Then you must refrain from using the connection during the download, or potentially using Free Download Manager might fix the issue too. Be aware that first download click on the site redirects you to advertisement without uBlock Origin, and it might lead to starting downloading malicious files instead. Also, uploading stops on 0% from certain hosts.

No audio previewing.

Unlimited

<https://qiwi.gq/>

(at least any info about limits in the account panel cannot be found)

Registration required

Unlimited

<https://pd.heracle.net/drive>

No file size limit (999TB Storage). No download speed limit.

Slow upload (around 5 Mbit/s), registration required. Unlimited file size upload and storage space.

Epiring/temporary or problematic

Unlimited

<https://catbox.moe/>

200MB max file size,
files kept forever, donations.

Don't use it. Some providers block the site and its subdomains: “litter.catbox.moe” (domain for downloading files uploaded with litterbox.catbox.moe), and litterbox.catbox.moe (expiring, with up to 1GB file limit) and e.g. not everyone on our server are able to download files from there. Also, not all VPNs work with it. Possible issues: SSL errors, DNS block (then using

108.181.20.36 might work) or IP block (then use VPN, but same issues may occur at times), timeouts, also be aware that deleted files by users or which already went offline have weird old school "404 Not Found nginx/1.18.0 (Ubuntu)" which might be misleading that there's something wrong with their provider, but it's just a file being offline. Countries blocking the site: Australia, UK, Ireland, Afghanistan, Iran. Providers: Verizon, Spectrum, Rogers, Quad9 DNS, Comcast/Comcast Business. [More](#)

Unlimited

<https://transfer.it/>

No file size limit, up to 90 days expiration

"New file upload service hosted by MEGA", no account required, MEGA opt. integration

8 minutes/25MB for free, 5 hours/250MB for pro accounts

For audio files only, it compresses all non-mp3 files to mp3 320kbps (mp3s remain untouched). Files uploaded without an account expire after 24 hours.

<https://whyp.it/>

2GB max file size, free, expiring up to 7 days (or paid), now requires email

<https://wetransfer.com/>

2,5GB max file size, free, no account, expiring

<https://send.vis.ee/>

5GB max file size/storage

<https://www.sendgb.com/>

Expiring after 1-90 days, paid 1TB storage and 500GB max file size

15 GB, expiring

<https://fileditch.com/>

10 GB, expiring

<https://tmp.ninja/>

Unlimited, but usually 14GB (10 day expiry date till last DL):

<https://gofile.io>

Don't use it,

as certain files, e.g. with 1GB size (at least in some cases), can be only downloaded with premium account if servers are overloaded - you can visit the link for 10 days till it expire in hope of the server being offloaded, but still not be able to download the file at all. GFY, WS.

Unlimited (till their server space is full, which sadly is often the case, files get deleted after 6 days):

<https://filebin.net/>

2TB (once you used your mobile app)

Baidu Pan

Users outside of China are restricted from registering, but there are ways to circumvent the issue with [Baidu Cloud](#) or [duspeaker](#). Guides: [1](#) | [2](#)

“I don't recommend it, unless you pay, you're stuck with 150KB/s downloads maximum” plus “a requirement to use their app to actually download (at least if you decide to share your files)”

100GB (files expire after 21 days/50 downloads, max 6GB per file)

<https://filetransfer.io>

It could mess up with filenames after downloading from direct link which was also possible [at least in the past] and could be used e.g. in Google Colab, there was some sneaky method of extracting direct links clicking on download buttons instead of sharing classic links

50GB without registration (up to 30 days of expiry date)

<https://www.swisstransfer.com/>

50GB without registration (up to 14 days expiry date):

<https://dropmefiles.com/>

250GB (expires 15/45 days after last download [un/reg], or never for 1TB 6\$ per month)

<https://filelu.com/>

1000GB

<https://www.terabox.com/>

(but I have some reports that after uploading X amount of data (it depends) they block the account and tell you to pay)

100GB

<https://degooc.com/>

(but Degoo has bots which look for DCMA content, and they close even paid accounts in such cases or even some files without any reason)

Unlimited

Depositfiles (now dfiles.eu)

10GB/file, FTP

They exist since forever (2006) and probably didn't collapse so far due to nightmarishly slow download speeds (20 or even 50KB/s, can't remember) for at least non-Gold accounts. But links get offline there occasionally too (maybe less frequently than some other services). Registration required.

Unlimited

[Chomikuj.pl](https://chomikuj.pl)

Only 50MB of downloading for free (even for your own files)

Be aware that it happened in the past, that along years, among very big collection of files, someone had I think some even encrypted private files deleted, but usually only DCMAed files are taken out.

Since around the end of 2023, they started sending PMs to users warning about deleting some of the files on their account, moving them to special folders with deletion date. You can reupload your file after deletion date again. The same action needs to be repeated at least once a year.

Also, the site exists since “forever” (2006), and didn’t collapse despite many court cases, probably due to creative changes of owners from specific countries. It can be also used as public sharing disk with points of transfer for downloaded content from your disk.

Unlimited

[Pixeldrain.com](#)

20GB/file, expiring for free accounts (4 months) or till the pro account is valid (min. 8 months?)

6 GB per day downloading cap for free, then it downloads with 1 MiB/s.

~~Some files uploaded on Pixeldrain are only available for Pro users.~~

~~On some files it will just tell you that servers are overloaded, and the error will last for days, weeks even, and not let you download. So I’d rather refrain from using it.~~ I think I confused it with an issue with gofile I once had.

How to download faster from sites like PixelDrain:

<https://pixeldrain-bypass.cybar.xyz/>

50GB

No registration required, 7 days expiry time for free

<https://fex.net/en/>

32GB

(down) <https://www.transferfile.io/>

Decentralized file hosting. If it goes down, perhaps the link can be replaced by ipfs.io

Code/datasets/models repository sites

[GitHub](#)

2GB/file

On the release page of (at least) public repositories, you can upload any files, and also bigger than directly in repositories (e.g. encrypted to avoid any problems - copyrighted content, even one music file in the repository can get taken down after some time).

You can split your archive into parts if necessary.

It's perfect to use in Colab notebooks. Very reliable and fast.

Huggingface

Lot of big models are stored there nowadays too. Probably there weren't any problems with their hosting in the past. Prob. 50GB file limit.

Zenodo

We had some issues with slow downloading from there in Colabs in the past, but tons of big datasets are stored there.

Bigger popular cloud services

(size provided for total storage space per account)

20 GB

mega.nz

No expiry, usual DCMA

One user from my other community had his whole account deleted years ago. It happened after a few file takedowns on his account before. That time he uploaded all music segments, basically assets extracted from a computer game publicly on a forum, and shortly later he was banned. It wasn't even arranged and ready for a normal release. Probably the publisher was snitching on him for quite some time already.

15GB

Google Drive

Big, suddenly popular files can become suddenly either offline or with some other error disappearing later during the day.

Also, it happened in the past that very popular, bigger files started being limited as visible only for the owner due to reaching quota. You could circumvent it by adding the file into your own GDrive (if you had enough space) - not sure if the trick still works.

Google reserves the right to delete the account after 2 years of inactivity. From what I've found, they don't delete accounts which have YouTube account with at least one video uploaded (not sure if it must be online and public). I had a case few times with revoked privileges to some documents on GDoc, which were changed to private without my knowledge. Can't guarantee if the same cannot happen with GDrive files.

15GB

Mediafire

I've seen very old uploads from there.

15GB

4shared.com

Max 3GB of daily traffic, 30GB per month

(Once I got my account deleted over years, maybe due to inactivity, but even if I was warned, messages were coming into Gmail spam)

15 GB

fileditch.com

It allows sharing direct links like from FTP or GitHub when you upload file on release page where also bigger files can be uploaded vs 50MB in source files. I see 9 months old links still active from fileditch. Download can however be slow, 5 mbit, on old files that have not been accessed in 30 days.

10.2 GB

safe.fiery.me

I think this has no expiration, not sure

10GB

<https://box.com>

250MB file limit for free

11GB

yandex.com/client/disk

At least after the war, some users started to report very slow downloads.

Rare DCMAs vs Mega.

Since 2022 difficult registration without a phone, and/or when you use public SMS receiving gate and/or VPN e.g. Russian - they can prevent you from access to a disk right after registration if they detect something suspicious during registration process, after the war they decreased max file size limit to 1GB iirc.

Also, there are very little means to recover your account in case your secret name miraculously get changed, and using new ISP or after long inactivity, you're asked for it (I had such situation in the past, and IDK how long the files are stored on inactive accounts).

Attempts of using automated forms to fill for account recovery are vain (esp. if you don't know precise account creation dates, didn't use their email, etc.). Even if you're logged into their Disk app on the phone, you cannot change any account related information like secret question or password without being redirected to their page and having to log into the account which you forgot the secret question (or it got suspiciously changed; quite honestly - I got it changed either by them after long inactivity of around 2 years, or by some attacker [but it would require secret question]).

5GB

OneDrive

MS have taken some space expansions given away for free once (some older accounts might be still bigger than that)

5GB

Proton Drive

End-to end encryption working also for sharing (although mega has probably the same)

2GB

Dropbox

With some options to expand it for free and also with referrals

20GB/month

files.fm

5GB upload per file limit/5GB zip file download limit,
information on expiring not provided (iirc possible to set manually at least);
unregistered up to 60 days.

Temporary file uploads that expire anytime (by HV):

<https://litterbox.catbox.moe/> (don't confuse with catbox.moe)

(1 GB)

<https://pomf.lain.la>

(512 MiB)

<https://cockfile.com/>

(128 MiB, IK funny name, but)

<https://uguu.se/>

(128 MiB)

<https://sicp.me/>

(114 MB)

<https://www.moepantsu.com/>

(128 MiB)

Hint. In the case of some free, not very well-known services which can even disappear after some longer period of time (do you remember RapidShare, copy.com, hotfile or megaupload, catshare, freakshare, uploaded.to, fileserv, share-online.biz, odsiebie, hostuje.net?) it's better to keep your files in more than one service (I recommend 3 copies for important data kept long), or stick to some popular big tech companies which are unlikely to disappear soon (if they don't take your upload down) or if another war will break out and increasing energy costs will make smaller services unprofitable like not long ago.

Paid:

- You can get 1TB OneDrive with an .edu email

- If you sign up for the Google Workspace (it was called G Suite until recently) version of Google Drive, you can get 1TB for ~\$10 USD a month, but here's the thing... I have been way over 1tb for a couple of years now, and they have never charged me anymore. I am over 4tb now and have been for ~3 months, and it is still only ~\$10. If you do it, just create a

1 user account and just keep filling it up until they say you need to add more users or pay more.

Well it looks like it is \$12 now, but it's for 2tb and maybe that is what they change my plan to and are charging me now too... I thought there was some kind of surcharge and tax (never really paid attention to the exact amount) but guess it is just \$12 + tax now...

<https://workspace.google.com/pricing.html>

It looks like they might have gotten rid of it, but it used to be \$50/month for unlimited storage, but I think as long as you do what I do, I think it is probably close to unlimited for \$12/month

It's pretending you're in a college and college drives have infinite storage

I used to have one for 1-2 years, but it suddenly got removed, so it's not safe. All the files are gone too, without notice

BTW. For workspace you still need to have your own domain (with the possibility of changing DNS entries, so free ones are out). Yearly cost is negligible, but you have to remember about it.

- Also, if you have ProtonVPN on the Proton Unlimited plan, you get 500GB of storage on Proton Drive for free.

- Also, Google Pixel 1 phones used to have like unlimited or at least bigger GDrive plans iirc (it was withdrawn from later Pixel phones). Some people bought these phones just for the space.

- You can get very cheap 2TB (around 16\$) for a year on Google Drive in Lyres (I think they only changed it in methods of payment, not necessarily whole region), but some people say it's better to get it in Brazil due to fewer problems.

I heard it's better to not buy it in Lyres on your main account, because your apps can get regional lock (e.g. Tidal). Some people even had problems with currency in their other accounts, and you can change it only once for a year on an account, and in case of some emergency, you might have to be forced using Revolut cards. There is a lot of misinformation about that promo trick so verify it all, but there should be some reasonable amount of info scattered around the net already (e.g. hotukdeals, pepper.pl, or the site's German counterpart).

- 50GB on Dropbox from some Galaxy phones (e.g. S3) can no longer be redeemed (since ~2016 I believe)

<https://www.multcloud.com/>

Service allowing moving files across various cloud accounts and services for free

Pitch detection and to midi convert

<https://discord.com/channels/708579735583588363/708579735583588366/1019280811461181510>

(outdated)

(for old MDX SDR 9.4 UVR model):

(input audio file on Colab can be max 44kHz and FLAC only).

Original MDX model B was updated and to get the best instrumental - you need to download invert instrumental from Colab.

Model A is 4 stem, so for instrumental, mix it, e.g. in Audacity without vocals stem (import all 3 tracks underneath and render). Isolation might take up to ~1 hour in Colab, but recently it takes below 20 minutes on 3.00 min+ track.

If you want to use it locally (no auto inversion):

<https://discord.com/channels/708579735583588363/887455924845944873/887464098844016650>

B 9.4 model:

<https://github.com/Anjok07/ultimatevocalremovergui/releases/tag/MDX-Net>

Or remotely (by CyberWaifu):

https://colab.research.google.com/drive/1R32s9M50tn_TRUGIkfnjNPYdbUvQOcfh?usp=sharing

Site version (currently includes 9.6 SDR model):

<https://mvsep.com/>

You can choose between MDX A or B, Spleeter 2/4/5 Stems), UnMix 2/4 stems, but output is mp3 only)

New MDX model released by UVR team on mvsep is currently also available. If you have any problems with separating in mobile browser (file type not supported) add for a file additional extension: trackname.flac.flac.

MDX is really worth checking. Even if you have some bleeding, and UVR model cuts some instruments in the background.

CyberWaifu Colab troubleshooting

If you have a problem with noise after a few seconds of the result files, try to use FLAC. After an unsuccessful attempt of isolation, you can try restoring default runtime to default state in options. The bug happened a few days after releasing the Colab suddenly one day and the

is prevailing to this day (so WAV no longer works). If you run the first cell to upload, and afterwards after opening file view, one of the 001-00X wav files is distorted (000 after few second) it means it failed, and you need to start over till you get all the files played correctly. But after longer isolation, it may cause reaching GPU limit, and you will not be able to connect with GPU. To fix it, switch to another Google account. If you have a problem, that your stems are too long, and mixed with a different song, restore default runtime settings as well, or delete manually

(outdated, deleted feature from HV Colab) Be aware that normalization turned on in Colab for instrumentals achieved with inversion may lead to occurrence of some artifacts during inversion, but general mix quality and snare in the mix might be more loud and sound more proper with normalization on, though it's not necessarily universal solution in every case when the track might sound a bit off than more flat sound of normalization turned off (at least in some parts of it).

AI-killing tracks - difficult ones to get instrumentals (or vocals) - a lot of e.g. vocal (or instrumental) leftovers in current models

"instrument-wise, the problematic ones I can remember are:

alto sax, soprano sax, any type of flutes/whistles (including synths), trombone slides, duduk, some organ sounds (close to sine wave sound)" plus harmonica, erhu, theremin.

"even if some models do a bit better job than others, these instruments are still problematic because their timbres are close to [human] voice"

And in general - songs heavily sidechained, with robotic, heavily processed vocals, sometimes with lots of weird sounding overdubs where some are missed (e.g. in trap), also laughs and moans.

Anjok stated that the hardest genre for separation is metal and vocal-centered mixes. If the instrumental has a lot of noise, e.g. distorted guitars, the instrumental will come out muddier. Tracks from the 70-80s can separate well. The 50-60s will be harder, e.g. recorded in mono. Early stereo era gets a little better.

Open [GSheet](#) with more songs for everyone with a Google account to contribute (we kinda tried not to duplicate any songs in both places too much).

Instrumentals

- Childish Gambino - Algorithm (robotic vocal effects, autotune, echoes, specific processing plugins on vocals, constant audible vocal residues for all current models)

- tatu - Not gonna get us / Nas ne dogonyat ("This song is impossible to quality separate by any model. Our dataset contains several songs by this artist, but this did not improve the result in any way. Just forget about it for a few years") - IRC, the result was enhanced by slowing down, a.k.a. soprano option on x-minus.
 - Eric Prydz - Call On Me (aggressive sidechain compression "It's literally ditching the vocal part [and instruments] out to make room for the kick. So yeah, good luck in getting that vocal back.")
 - Jamaroquoi - Virtual Insanity ("One of the most difficult challenges of all my experience has been that is not very well handled even when maxing out quality in v5.")
Others:
 - Beyonce - I'm that girl
 - Half Alive - Still feel
 - Queen - White Queen
 - Queen - Bohemian Rhapsody (very complicated song; mix of various vocals and guitars)
 - Queen - These Are The Days Of Our Lives - to evaluate BVE model and how it reacts with harmonies. If it works on this track, probably all the others will work.
 - J Dilla - Don't Cry (lots of so-called lo-fi "cuts" or chops of vocals from old vinyls, characteristic for hip-hop productions which are sometimes harder to separate)
 - Lots of Juice WRLD (his tracks have leftovers here and there, e.g. in "Off the rip (Gamble)")
 - Eminem - No regrets (constant low-volume vocal leftovers)
 - Louis The Child - Better not (problem with vocals with currently the best MDX23 MVSEP beta model, and also Demucs ft and Kim model)
 - A\$AP Rocky - Fashion Killa (same as for "Night Lovell - Dark Light" - "almost every AI can't separate the main vocals from the melody, the melody has a part that sounds like vocals, so just about every AI picks some of it up in the vocals section instead of the instrumental section")
 - Porcupine - Trees Don't Hate Me ("Quiet bits, loud bits, flutes and strings, things I can't even name plus all the usual suspects, drums etc, and Steven Wilson has a crisp clean voice a lot of the time")
 - Thomas Anders - You Will Be Mine (vocal residues in instrumentals using all current models for April 2023)
 - Modern Talking - One in a million (also minimum vocal residues)
 - Modern Talking - Mrs. Robota (too many synthesizer effects bleed in vocals of MVSEP MDX23 10.04.23 Ensemble [so consisting of only kim vocal 2, kim inst and Demucs 4 yet])
 - Crush 40 - Live & Learn'
 - JPEGMAFIA - HAZARD DUTY PAY! (hard to get vocals from rapping section; Kim vocals 1)
 - Bjork - All Is Full Of Love (2:06, 2:12, 2:50 and throughout from that point, the vocals partially still bleed.)
- The song was tested on MDX Inst 3, Inst HQ_1 and 3, Inst Main, Kim Inst,

HTDemucs, HTDemucs_FT and 6S, and ensembles including (Kim vocal 2, Kim Inst, Inst Main, 406, 427, HTDemucs FT) and (Kim Inst, Voc FT, Inst HQ 3)

- Stray Kids - GO生 (GO LIVE)
- Royce da 5'9" - I'm The King (Vinyl rip)
- Mike Mareen - Love Spy
- Bomfunk MC's - Freestyler
- Lasgo - Something
- South Park - Chocolate Salty Balls (bad results with most models)
- Tally Hall - Never meant to know (the almost impossible goal for now is to remove "with" in 2:39).
- WWE - Demon in Your Dreams - (here's a track that sounds bad - the parts where the vocals usually are sound muffled and dull, guitars are barely audible - HQ_3, Demucs 6s tested)
- Taylor Swift - Better Than Revenge (Taylor's Version) (background vocals in all models including HQ3 and voc_ft - using Dolby Atmos version, and (I think just) muting (?vocal) channel(s) helped)
- Bon Jovi - I believe
- Twenty One Pilots - The Hype ("voc_ft leaves too many perc/drums/synths [in vocals] that sounds like t's and s' or just sound like vocals, and it's really annoying, also because of this nearly no other model can separate it either because they think it's part of the vocals, but it's mostly just synths, Ripple put a lot of echo into the other stem")
- The Weeknd - Until I Bleed Out (vocal stem includes a bunch of drum and synth bleeding. Tested on htdemucs_ft, VocFT, Inst HQ 3, InstVoc HQ 2, Kim 1, Kim 2, Kim inst, and ensembles (htdemucs_ft, VocFT, Inst HQ 3, InstVoc HQ 2), (Kim 2, Kim inst, Inst Main, 406, 427, htdemucs_ft) and (Kim inst, VocFT, Inst HQ 3))
- Travis Scott - Nightcrawler (vocal residues in 1:27, 2:24, 3:56, and 4:50 using BS-Roformer 1296 in UVR beta and overlap 2/8 - less than in other ech models though it's more muddy, though 04.24 model on MVSEP less, [discussion](#))
- Shaft - Mambo Italiano
- Yello - Oh Yeah (both Aufr33 suggestions)
- song_45_vocals in the multisong dataset have some weird effects which some models struggle with
- Black GryphOn - Jester (Pomni's Song) (feat. Lizzie Freeman) "the vocoded vocals in the drop of this song [are] impossible to isolate with current models either"
- Isabela Souza - Tu color para pintar - violin put in vocal stem in a lot of models (e.g. Bas Curtiz FT 25.10.24)
- George Michael - Amazing (Vocal bleed starting at 3:52 with zfturbo mel roformer 2024.10, residues with Dango too)
- Eminem - The Warning (quite laugh residue at 0:04; instr unwa v1)
- Gregory Brothers - Dudes a Beast (trumpets in vocal stem at 0:51; unwa's beta4 and inst v1e, fixed in Gabox voc_fv5)

- DJ Krust / Saul Williams - Coded Language ("I can hear a light bleed of the lowcutted bassline at times" becruily Mel)
- Dayseeker - Sleeptalk ("serious volume clipping issues throughout when removing vocals (...) I've tried every model and several ensembles, edited UVR settings, and also tried Dango." - comfortable couch)
- Darko - Starfire (almost everything can't remove that particular scream; shift conversion pitch method kinda did - mesk)
- Meshuggah - Ayahuasca Experience
- Fredrik Thordendal's Special Defects - Vitamin K Experience (A Homage to The Scientist / John Lilly)
- Thievery Corporation - Culture of Fear ("only supported by MDX Kar v2 to have an instrumental with backing vocals, all the other karaoke models/bve fail" dca)
- Inside Out · Bad Suit ("all UVR models I've tried (all that jarredou had on his Colab) struggle with the slap bass. Slap sounds (which contains mid to high frequencies) goes into guitar stem, so that's still hard" - 03.2025)
- "Songs that were produced at Cheiron Studios from the 90s/00s still don't isolate well
- the mixes are so d**n complex" - JadDeluxe
- Yello - Oh Yeah
- Duck Sauce - Barbra Streisand (Radio Edit) (sidechain is the problem I assume)
- Al Bano & Romina Power - Sharazan
- Billie Eilish - L'AMOUR DE MA VIE [OVER NOW EXTENDED EDIT] ("it seems like none of the existing models can pick up those high pitched vocals" - black_as_night)

Complex vocals and vocoder - severe bleeding on every model tested as of 06.12.24 including Dango (dca100fb8 contributions)

- Tame Impala - One More Year
- Daft Punk - Around the World
- Daft Punk - Television Rules the Nation
- Daft Punk - Doin' it Right
- Daft Punk - Human After All
- Daft Punk - Get Lucky
- Daft Punk - Lose Yourself to Dance
- Daft Punk - Robot Rock
- Deep Forest - Sweet Lullaby (Version 1992) (yodelling difficult to remove; BV bleed starting at ~1:21)
- Radiohead - The National Anthem (vocal effects difficult to separate; LV effects bleed starting at ~1:36)
- Moby - One Last Time (vocal bleed in instrumental; LV effects bleed starting at 1:39)
- Air - Run
- Kodex - Do jutra (esp. at 1:30 vocals bleeding in a lot of models with low bleedless metric)

Bleeding on every model tested as for 06.12.24 incl. Dango - not so severe, but containing vocal pop-ins (dca100fb8):

- "Supersonic" by Jamiroquai (BV bleed starting @~0:07)
- "Amazing" by George Michael (BV bleed starting @~3:45)
- "El lilady" by Samo Zaen (BV bleed starting @~3:30)
- "Here Comes the Rain Again" by Eurythmics (BV bleed starting @~1:17)
- "Sun is Shining" by Bob Marley & The Wailers (BV bleed starting @~1:52)
- "Sun is Shining (Kaya 40 Mix)" by Bob Marley & The Wailers (BV bleed starting @1:52)
- "Road to Zion" by Damian Marley (BV bleed starting @~0:00)
- "Les Rubans" by Daniel Masson (LV bleed starting @~2:13)
- "Remember" by Air (BV bleed starting @~0:46 + LV bleed starting @~0:58)
- "Strangers" by Portishead (LV bleed starting @~0:30)
- "Samsam (Chanson du générique)" (BV bleed starting @~0:00)
- "Aicha" by Khaled (BV bleed starting @~3:45)
- "Forest Hymn" by Deep Forest (BV bleed starting @~0:33)
- "33 Degree" by Thievery Corporation (LV effects bleed starting @~1:44)
- "Run" by Air (BV bleed starting @~1:08)
- "An Indian Summer" by Al-Pha-X (BV effects bleed starting @~3:13)
- "Forest Hymn" by Deep Forest (BV effects bleed starting @~0:33)
- "In The Air Tonight" by Phil Collins (LV effects bleed starting @~3:03)
- Goldfrapp - Utopia (BV bleed starting @~0:00)
- Depeche Mode - Sacred (BV bleed starting @~0:00)
- Seal - Love's Divine (BV bleed starting @~3:26)
- Da Lata - Alice (No País Da Malandragem) (LV or BV bleeding at a different time with each model)
- ABBA - Gimme! Gimme! Gimme! (A Man After Midnight) (BV bleed starting @1:22)
- 311 - Beyond the Gray Sky (LV effects bleed @~1:24)

Vocal bleed in instrumentals using MVSEP MelRoformer 2024.10 model which Dango fixes (dca100fb8)

- "For You" by Coldplay (BV bleed starting @1:32 --> v1e, Kim Mel or even Dango doesn't have this issue)
- "L'été indien" by Joe Dassin (BV bleed starting @0:24 --> v1e or Dango fixes the problem)
- "Porcelain" by Moby (LV bleed starting @2:10 --> Dango and SCNet XL high fullness models on MVSEP fix the issue)
- "Night Bird" (from "Essence of the Forest" album) by Deep Forest (BV bleed starting @0:28 --> Dango fixes the issue)
- "Desert Walk" (from "Essence of the Forest" album) by Deep Forest (BV bleed starting @0:03 --> Dango fixes the issue)

- "Desert Walk (Version 1992)" by Deep Forest (BV bleed starting @0:07 --> Dango fixes the issue)
- "Love Is Gone (Fred Riester & Joachim Garraud Radio Edit Remix)" by David Guetta (BV bleed starting @0:49 --> Dango fixes the issue)
- "Attention Mesdames et Messieurs" by Michel Fugain & Le Big Bazar (BV bleed starting @0:50 --> Dango fixes the issue)
- "Everything In Its Right Place" by Radiohead (BV bleed starting @0:52 --> Dango fixes the issue)
- "Sweet Dreams (Are Made Of This)" by Eurythmics (BV bleed starting @0:48 --> Dango fixes the issue)
- "Lebanese Blonde" by Thievery Corporation (BV bleed starting @0:52 --> Dango fixes the issue)
- "Lift Me Up" by Moby (LV chops starting @2:45 --> Dango fixes the problem)
- U.S.A. for Africa - We Are The World (BV bleed starting @5:34, fixed using v1e)
- Led Zeppelin - Kashmir (LV bleed starting @2:16, fixed using v1e)
- Jamiroquai - White Knuckle Ride (LV bleed starting @0:15, fixed using v1e)

Duplicates from the [GSheet](#)

- Moby - Porcelain (in Gsheet; in instrumental, vocal reverb bleeding at 1:00, and bleed at 2:10/20, all good MDX models, GSEP, MDX23 by ZFTurbo tested, still getting more or less the same results, Dango and SCNet XL fixes the issue)
- Queen - March Of The Black Queen (always causes issues, the best result on Full Band 8K FFT, as for 06.08.23, but still lot of BV is missed)
- Night Lovell - Dark Light ("almost every AI can't separate the main vocals from the melody, the melody has a part that sounds like vocals, so just about every AI picks some of it up in the vocals section instead of the instrumental section")
- Bob Marley - Sun is Shining (all current models bleed in the same timestamps: 1:02, 1:42, 1:54, 1:57, 2:50)
- Daft Punk - Give back life to music (problem with vocoder in the vocals rendering bad instrumental results)
- Daft Punk - Within (robotic voices)
- Frank Ocean - White Ferrari

List of songs where all the current Mel-Roformer instrumental models fail in recognizing some instruments correctly in the instrumental track (they mainly struggle with sax and harmonica), whereas not the case with SCNet XL except for talkbox, theremin and erhu (SCNet still fail at it for these) by dca100fb8

- Cowboy Junkies - I Don't Get It (harmonica picked up in vocal, starting @00:12)
- Pink Floyd - Shine On You Crazy Diamond (Parts I-V) (saxophone picked up in vocal, starting @11:09)
- Travis - Sing (FX picked up in vocal, starting @00:00)
- Thievery Corporation - Revolution Solution (FX picked up in vocal, starting @00:00)

- Thievery Corporation - Safar (The Journey) (instrument picked up in vocal, starting @00:02)
- Samsam Song (International Version) (flute picked up in vocal, starting @00:00)
- Portishead - Humming (FX and theremin picked up in vocal, starting @00:00)
- Asian Dub Foundation - Tu Meri (instrument picked up in vocal, starting @02:33)
- Goldfrapp - Horse Tears (organ picked up in vocal, starting @00:00 + elec guitar @01:28 + talkbox @01:22)
- Goldfrapp - Lovely Head (talkbox picked up in vocal, starting @01:15)
- Goldfrapp - Pilots (talkbox picked up in vocal, starting @00:00)
- Moby - Lift Me Up (synth picked up in vocal, starting @00:00)
- Phil Collins - In The Air Tonight (elec guitar picked up in vocal, starting @02:10)
- Pink Floyd - Money (saxophone picked up in vocal, starting @02:00)
- Thievery Corporation - Radio Retaliation (FX picked up in vocal, starting @00:00)
- Thierry David - Huong Vietnam (erhu picked up in vocal, starting @00:35)
- Supertramp - The Logical Song (saxophone picked up in vocal, starting @01:52)
- Supertramp - School (harmonica picked up in vocal, starting @00:00)
- Archive - Again (harmonica picked up in vocal, starting @00:00)
- Deep Forest - Desert Walk (Version 1992) (flute picked up in vocal, starting @00:00)
- Deep Forest - Dignity (elec guitar picked up in vocal, starting @00:36)

Vocal bleeding not existing on MVSEP's SCNet XL high fullness vs Roformers

- Tame Impala - On Track

You can visit our [#request-separation](#) channel to look for some interesting cases of people seeking help with some specific songs they struggle with and a new [#your-bad-results](#) channel.

Songs to compare weaker vs more effective models in instrumentals (e.g. inst 464/Kim inst or HQ_2/3 or 4 vs all others)

- O.S.T.R. - Incognito (non Snap Jazz version) (lo-fi Polish hip-hop with constant vocal leftovers in all models and AIs except MDX-UVR inst 1-3, main where inst 3/464 performs the best, it's also good to test an influence of various chunks settings at 1:53. Publicly available songs for datasets usually don't include hip-hop at all, especially not from some low, weird sounding languages with loud, bassy, over processed voices. In Snap Jazz version also in 464 there are e.g. less vocal residues than on GSEP - still slightly hearable).
- Kaz Bałagane - Stara Bida (constant vocal leftovers in all models and AIs except MDX-UVR inst 1-3 and inst main where inst 3/464 performs the best [good to test weaker models or specific epochs], flute from 1:11 gets deleted on MDX-UVR HQ models).
- The Weeknd - Hardest To Love (htdemucs_ft did well here).

- NNFOF - Jeśli masz nierówno pod sufitem (all MDX-UVR instrumental models will filter out inconsistently flute from the track, while GSEP handles that song well - it happens for all kinds of songs containing flute and oriental instruments in these models)
- Ace of Base songs ("any of them have those flute-ish synthetic instruments which have always been a nightmare in terms of getting a flawless a cappella").
- Różewicz Interpretacje (Sokół) - Wicher (very deep and low rap voices cause problems with weaker models, e.g. original MDX23 on mvsep1.ru (now MVSEP.com)/ZFTurbo MDX23 Colab; you can also try out also Sokół - Nic and Sokół - Wojtek Sokół albums)
- Chaos (O.S.T.R., Hades) - Powstrzymać Cię (lots of bleeding in e.g. MDX23 model on MVSEP in 2:00. Not that much in Kim inst)
- DJ Skee & The Game (from 2012 mixtape) or Tyler the Creator (album version from 2011) - Yonkers (same beat prod. by Tyler the Creator)
(the first is from a mixtape with more cuts/vocal chops difficult to get rid of. HQ models usually confuse vocal chops with vocals, but here it might be useful)
- Avantasia - The Scarecrow (HQ3 generally has problems with (here bowed) strings. mdx_extra from Demucs 3 had better result, sometimes 6s model can be good compensation for these lost instruments)
- Static Major - CEO (if someone wants to test out isolation of many vocal layers using e.g. Melodyne)
- oikakeru yume no saki de - sumikeke (here vocal layers extraction Karaoke models and Melodyne fail)
- Dizzy Wright - No Writer Block (hard track to keep hi-hats consistent throughout the whole output with even some snares - it can all get easily washed out, also more vocal leftovers in MDX23C ensemble on MVSEP1.ru vs MDX23 2.1 Colab [despite better SDR], not bad GSEP result, but it makes hi-hats like a bit out of rhythm probably due to some built-in processing in GSEP)
- Dariacore - will work for food (generally that whole Dariacore album can be tasking due to its loudness and "craziness")
- Centrala Katowice - Reprezentowice (first version of GSEP in 192kbps was consistently failing in picking up vocals and also leaving strong vocal residues)
- Cher - The Music's No Good Without You ("there is progress on removing Cher's vocals. Previously, this song was an AI killer, but the mel-rofomer model removes the vocals almost completely (...) Removing vocals from "Believe" is no problem either." Aufr33)

A list of songs which have vocal bleed in the instrumental using unwa's v1e model
 "These songs also present issues after using Mel 2024.10 and BS 2024.08 from MVSEP, but the timestamps where the bleed occurs might be different" plus Mel 2024.10 model might have less of these residues by dca100fb8

- "Supersonic" by Jamiroquai (BV bleed starting @0:07)
- "Amazing" by George Michael (BV bleed starting @3:45)

- "Here Comes the Rain Again" by Eurythmics (BV bleed starting @1:17)
- "Porcelain" by Moby (LV bleed starting @1:00 --> Dango and SCNet XL fix the issue)
- "Sun is Shining" by Bob Marley & The Wailers (BV bleed starting @1:52)
- "Sun is Shining (Kaya 40 Mix)" by Bob Marley & The Wailers (BV bleed starting @1:52)
- "Give Life Back to Music" by Daft Punk (LV bleed starting @0:49 --> Dango fixes the issue)
- "Road to Zion" by Damian Marley (BV bleed starting @0:00)
- "El lilady" by Samo Zaen (BV bleed starting @3:30)
- "Night Bird" (from "Essence of the Forest" album) by Deep Forest (BV bleed starting @1:12 --> Dango fixes the issue)
- "Desert Walk" (from "Essence of the Forest" album) by Deep Forest (BV bleed starting @0:44 --> Dango fixes the issue)
- "Desert Walk (Version 1992)" by Deep Forest (BV bleed starting @0:18 --> Dango fixes the issue)
- "Love Is Gone (Fred Riester & Joachim Garraud Radio Edit Remix)" by David Guetta (BV bleed starting @1:15 --> Dango fixes the issue)
- "Attention Mesdames et Messieurs" by Michel Fugain & Le Big Bazar (BV bleed starting @0:50 --> Dango fixes the issue)
- "Strangers" by Portishead (LV bleed starting @0:30)
- "Everything In Its Right Place" by Radiohead (BV bleed starting @3:28 --> Dango fixes the issue)
- "The National Anthem" by Radiohead (LV effects bleed starting @1:36)
- "Samsam (Chanson du générique)" (BV bleed starting @0:00)
- "33 Degree" by Thievery Corporation (LV effects bleed starting @1:42)
- "Sweet Dreams (Are Made Of This)" by Eurythmics (BV bleed starting @0:48 --> Dango fixes the issue)
- "Sweet Lullaby (Version 1992)" by Deep Forest (BV bleed starting @1:21)
- "Lebanese Blonde" by Thievery Corporation (BV bleed starting @0:52 --> Dango fixes the issue)
- "Forest Hymn" by Deep Forest (BV bleed starting @0:33)
- "Aicha" by Khaled (BV bleed starting @3:45)
- "Run" by Air (BV bleed starting @1:08)
- "Remember" by Air (LV bleed starting @0:31)
- "Doin' it Right" by Daft Punk (LV bleed starting @1:21)
- "Human After All" by Daft Punk (LV bleed starting @0:49)
- "Get Lucky" by Daft Punk (BV bleed starting @4:06)
- "Lose Yourself to Dance" by Daft Punk (BV bleed starting @1:55)
- "Robot Rock" by Daft Punk (LV bleed starting @1:02)
- "J'ai demandé à la lune" by Indochine (BV bleed starting @1:45)
- "Hey Jude" by The Beatles (LV bleed starting @0:00)
- "Within" by Daft Punk (LV bleed starting @1:42)
- "Lift Me Up" by Moby (LV chops starting @2:45)

- "Nothing Else" by Archive (LV effect bleed starting @1:13)
- "One More Year" by Tame Impala (BV bleed starting @1:03)
- "Around the World" by Daft Punk (LV bleed starting @3:57)
- "Television Rules the Nation" by Daft Punk (LV bleed starting @1:49)

List of songs which have important crossbleeding of vocals in instrumental using "basic" SCNet XL model from mvsep (don't confuse with undertrained SCNet XL on ZFTurbo GitHub) by dca100fb8

- Thievery Corporation - Where It All Starts (crossbleeding starting @0:15)
- Thievery Corporation - Le Monde (crossbleeding starting @0:02)
- Thievery Corporation - Lebanese Blonde (crossbleeding starting @0:52)
- Jamiroquai - Canned Heat (crossbleeding starting @0:56)
- Jamiroquai - Black Crow (crossbleeding starting @0:55)
- Jamiroquai - Supersonic (crossbleeding starting @0:07)
- Andru Donalds - Mishale (crossbleeding starting @0:50)
- Moby - Porcelain (crossbleeding starting @2:10)
- George Michael - Amazing (crossbleeding starting @0:28)
- Samo Zaen - El lilady (crossbleeding staerting @3:30)
- Zero 7 - In The Waiting Line (crossbleeding starting @0:48)
- Coldplay - For You (crossbleeding starting @1:33)
- Kool & The Gang - Fresh (Single Version) (crossbleeding starting @0:51)
- Kool & The Gang - Too Hot (Single Version) (crossbleeding starting @1:07)
- Khaled - Aicha (crossbleeding starting @3:22)
- Keane - Put The Radio On (crossbleeding starting @1:30)
- Nelly Furtado - Say It Right (crossbleeding starting @0:23)
- Black Sabbath - Planet Caravan (crossbleeding starting @0:10)
- Groundation - Smile (crossbleeding starting @0:10)
- Eurythmics - Here Comes the Rain Again (crossbleeding starting @1:23)
- Damian Marley - Road to Zion (crossbleeding starting @0:10)
- David Guetta - Love Is Gone (Fred Riester & Joachim Garraud Radio Edit Remix) (crossbleeding starting @1:15)
- Indochine - J'ai demandé à la lune (crossbleeding starting @1:47)
- Bob Marley & The Wailers - Sun is Shining (crossbleeding starting @1:53)
- Morcheeba - Blindfold (crossbleeding starting @0:53)
- Daniel Masson - Les Rubans (crossbleeding starting @2:13)
- Depeche Mode - Sacred (crossbleeding starting @0:00)
- Goldfrapp - Utopia (crossbleeding starting @0:11)
- Da Lata - Alice (No País Da Malandragem) (crossbleeding starting @0:38)
- Joe Dassin - L'été indien (crossbleeding starting @0:24)
- Led Zeppelin - Kashmir (crossbleeding starting @2:16)

List of songs which have bleed in the vocal track using the new BS-Roformer Revive v1 experimental vocal model by unwa (dca's contribution as well):

- Travis - Sing (FX picked up in vocal, starting @00:09)
- Thievery Corporation - Safar (The Journey) (instrument picked up in vocal, starting @00:03)
- Asian Dub Foundation - Tu Meri (instrument picked up in vocal, starting @02:33)
- Thievery Corporation - Radio Retaliation (FX picked up in vocal, starting @00:00)
- Thierry David - Huong Vietnam (erhu picked up in vocal, starting @00:35)
- Deep Forest - Dignity (elec guitar picked up in vocal, starting @00:36)
- Zaz - Je veux (whole kazoo solo picked up in vocal, starting @02:08)
- Archive - Fool (harmonica picked up in vocal, starting @00:00)
- Portishead - Humming (FX and theremin picked up in vocal, starting @00:00)
- Moby - Lift Me Up (synth picked up in vocal, starting @00:00)
- The Cardigans - My Favourite Game (guitar picked up in vocal, starting @00:10)
- Talk Talk - It's My Life (FX picked up in vocal, starting @00:05)
- Radiohead - The National Anthem (various FX and instruments picked up in vocal throughout the song)
- Zero 7 - Distractions (synth/FX bleed @00:07/@04:29)
- Porcupine Tree - What Happens Now? (synth/FX/guitar bleed @~07:31)
- Porcupine Tree - Start of Something Beautiful (synth bleed @04:55)
- Porcupine Tree - The Start of Something Beautiful (Live) (synth bleed @04:38)
- Porcupine Tree - Don't Hate Me (guitar bleed @00:29/@03:54)
- Porcupine Tree - Dark Matter (synth bleed @03:18/@05:42)
- Porcupine Tree - Arriving Somewhere but Not Here (synth bleed @04:12, elec guitar bleed @04:46)
- Porcupine Tree - Way out of Here (Live) (synth bleed @00:13, @06:36)
- Radiohead - Pyramid Song (FX bleed @00:05/@04:08)
- Talk Talk - Happiness is Easy (wind instrument bleed @04:20)
- The Cinematic Orchestra - Evolution (scratching bleed @04:38)
- Pink Floyd - Dogs (elec guitar/fx/synth bleed at times)
- Archive - Again (harmonica/fx/synth bleed at times)
- Archive - Lights (fx/synth bleed at times)

List of song where Roformer SW and BS 2025.06 solves the problem of vocals/BV crossbleeding in some songs by dca100fb8

- Jamiroquai - Canned Heat (no crossbleeding)
- George Michael - Amazing (no crossbleeding)
- Samo Zaen - El lilady (no crossbleeding)
- Coldplay - For You (no crossbleeding)
- Kool & The Gang - Fresh (Single Version) (no crossbleeding)
- Kool & The Gang - Too Hot (Single Version) (no crossbleeding)
- Khaled - Aicha (no crossbleeding)
- Nelly Furtado - Say It Right (no crossbleeding)
- Eurythmics - Here Comes the Rain Again (no crossbleeding)

- Eurythmics - Sweet Dreams (Are Made of This) (no crossbleeding)
- David Guetta - Love Is Gone (Fred Riester & Joachim Garraud Radio Edit Remix) (no crossbleeding)
- David Guetta - Love Don't Let Me Go (no crossbleeding)
- Bob Marley & The Wailers - Sun is Shining (no crossbleeding)
- Bob Marley & The Wailers - Running Away (no crossbleeding)
- Daniel Masson - Les Rubans (no crossbleeding)
- Michael Jackson - Workin' Day And Night (no crossbleeding)
- Radiohead - Morning Bell (from Kid A) (no crossbleeding)
- Radiohead - Scatterbrain (no crossbleeding)
- Portishead - Strangers (no crossbleeding)
- Moby - Lift Me Up (no crossbleeding)
- Seal - Love's Divine (no crossbleeding)

Unwa's BS Roformer Resurrection Inst model fixing crossbleeding of vocals in the instrumental by the first time. So it fixes the crossbleeding problems like BS Roformer SW/2025.06/07 did on these songs, for some reason (dca):

- Bob Marley - Running Away (Kaya 40 Mix)
- Samo Zaen - Tonight
- Eurythmics - Here Comes The Rain Again
- Eurythmics - Sweet Dreams (Are Made of This)
- George Michael - Amazing
- Kool & The Gang - Fresh (Single Version)
- Kool & The Gang - Too Hot (Single Version)

Songs which can't be separated to an instrumental with BVs (generally because lead vocals can't be differentiated from backing vocals) by dca100fb8 (from before becruily karaoke model release) by dca100fb8

- Nelly Furtado - Maneater (@0:16, LV are counted as BV because of the panning, but using stereo 50% with uvr bve v2 doesn't solve the issue)
- Lenny Kravitz - Low (@1:03, BV are counted as LV by BVEs or Kar models, including Dango and LALAL.AI)
- Timbaland - The Way I Are (@0:25, LV are counted as BV by BVEs or Kar models, including Dango and LALAL.AI, uvr bve v2 50% stereo LV panning failed)
- Timbaland - Give It To Me (@0:25, LV are counted as BV by BVEs or Kar models, including Dango and LALAL.AI, uvr bve v2 50% stereo LV panning failed)
- Timbaland - Morning After Dark (@2:02, BVEs and Kar models couldn't differentiate LV from BV, including Dango and LALAL.AI, uvr bve v2 50% stereo LV panning failed)

Fixed:

- Simply Red - Sunrise (@0:45, LV are counted as BV because of the panning, but using stereo 50% with uvr bve v2 doesn't solve the issue; fixed by Dango Backing Vocal Keeper by processing left then right channel)

List of difficult songs to extract the BVs with becruily's karaoke model - divided in three categories by dca100fb8

Very important:

- Phil Collins - In The Air Tonight (LVs crossbleeding during the whole song starting @00:52; BVE v2 fixes it)
- UB40 - Red Red Wine (LVs crossbleeding during the whole song starting @00:00)
- Supertramp - It's Raining Again (it still has the lead vocals during the whole song after conversion to Inst w/ BV; MDX Kar v2 fixes the problem)
- Phil Collins - I'm Not Moving (LV crossbleeding, It seems Becruily Kar has issues with the way Phil Collins's voice is often mixed)

Important:

- Charlie Winston - Kick the Bucket (BVs are missing starting @01:42)
- Coldplay - Daylight (LVs are still present starting @00:28)
- Massive Attack - Spying Glass (LVs are still present in the whole song starting @00:19)
- Indochine - J'ai demandé à la lune (BVs are missing starting @02:28)
- Pierpoljak - Pierpoljak (Radio Edit) (BVs/LVs crossbleeding starting @00:05)
- Jamiroquai - King for a Day (BVs are missing starting @00:55)
- Jamiroquai - Light Years (BVs are missing starting @00:22)
- Jamiroquai - Rock Dust Light Star (BVs are missing starting @01:42)
- ABBA - Dancing Queen (LVs are still present starting @00:19)
- Archive - You Make Me Feel (BVs/LVs crossbleeding starting @00:24)
- Justin Timberlake - Rock Your Body (BVs/LVs crossbleeding starting @00:28)
- Simply Red - Turn It Up (BVs/LVs crossbleeding starting @00:04)
- Justin Timberlake - SexyBack (BVs/LVs crossbleeding starting @00:15)
- Demis Roussos - From Souvenirs to Souvenirs (BVs/LVs crossbleeding starting @01:34)

Less important:

- Jamiroquai - Hot Tequila Brown (LVs are still present starting @01:19)
- Bee Gees - How Deep Is Your Love (LVs are still present starting @00:47)
- Simply Red - How Could I Fall (BVs are missing starting @01:40)
- Simply Red - Something Got Me Started (BVs are missing starting @00:53)
- Phil Collins - Don't Let Him Steal Your Heart Away (BVs/LVs crossbleeding starting @01:51)

- Phil Collins - Another Day in Paradise (BVs/LVs crossbleeding starting @01:13)
- Phil Collins - That's Just the Way It Is (BVs/LVs crossbleeding starting @00:59)
- Groundation - Confusing Situation (LVs bleed starting @00:24 + crosbleeding starting @01:06)
- Wham! - Everything She Wants (BVs missing starting @00:41)
- Thievery Corporation - Thief Rockers (BVs missing starting @00:00)
- Thievery Corporation - Radio Retaliation (BVs missing starting @00:01)

Due to uncommon panning:

- Nelly Furtado - Maneater (@0:16, LV are counted as BV because of the panning, but using stereo LV Panning doesn't solve the issue)
- Lenny Kravitz - Low (@1:03, BV are counted as LV by BVEs or Kar models)
- Timbaland - The Way I Are (@0:25, LV are counted as BV by BVEs or Kar models, LV panning failed)
- Timbaland - Give It To Me (@0:25, LV are counted as BV by BVEs or Kar models, LV panning failed)
- Timbaland - Morning After Dark (@2:02, BVEs and Kar models couldn't differentiate LV from BV, LV panning failed)
- Simply Red - Sunrise (@0:45, LV are counted as BV because of the panning, but using Lead Vocal Panning doesn't solve the issue)
- Jamiroquai - Automaton (Lead vocal panning didn't help)
- Thievery Corporation - Culture Of Fear (MDX v2 works)
- Jamiroquai - Supersonic (Lead vocal panning didn't help)
- Jamiroquai - Travelling Without Moving (Lead vocal panning didn't help)
- Philip Bailey & Phil Collins - Easy Lover (LVs very difficult to differentiate from BVs)
- Thievery Corporation - Sol Tapado (Lead vocal panning didn't help)
- Moby - Landing (LV panning didn't help)
- Moby - We Are All Made of Stars (LV panning didn't help)
- The Weeknd - Sacrifice (@01:25, LV panning didn't help)
- Nitin Sawhney feat. The London Symphony Orchestra - Songbird (LV panning didn't help)

Uncategorized

- Night Lovell - Dark Light (thx 97chris)
- Supertramp - Dreamer (LVs difficult to distinguish from BVs - dca)

Warning. If you upload lots of music in on our server (or any other server), recently our long-term users receive warnings from Discord about possible deletion of their accounts and whole good results channel got deleted - we advise sharing only links to e.g. GDrive or any other cloud instead of uploading music directly to Discord. So far our user received two warnings from Discord without deleting account yet. The whole good results channel got deleted after linking to uploads instead of uploading

after the last clean-up we got. Recently we added bot automatically deleting audio files uploaded directly to Discord instead of links added and the channel has been reopened.

Training models guide

[Read mesk's guide](#) (new link #2), then proceed below for arch explanations and more details.

As for a training code for Roformers and MDX23C, SCNet or adding new archs, most people use [MSST](#) by ZFTurbo. It's also provided with bunch of documentation.

"You can start with Sacial MSST [WebUI](#). I use that to train all my models" - Gabox

Introduction

"There are three components to the model scaling laws.

They are the size of the data set, the number of parameters in the model, and the computational resources." unwa

For training, depending on model type (explained above), it can be e.g. three files for training e.g. vocal model - vocals, instrumental, and mixture. When you'll try to train without mixture, the results will be "terrible" (iirc it was said somewhere in times of Mel Kim's model).

"You can train any sound you want with any architecture (MDX-Net, Demucs, Spleeter)"

But don't use Spleeter, it's deprecated since so many archs were released ([Kim](#)).

Just be aware that not every arch is a good choice for some specific tasks or instruments.

(Among others, the following based also on Anjok's [interview](#), around 0:40:00)

For training a new model, use at least 200 samples for such a model to achieve any good results. Anything below that might give you the results you might not be happy with, and of course, above that will give better results.

For BS/Mel Roformers, 525 songs were not enough to train a good model from scratch at some point.

Q: Anyone know how many songs are generally needed to finetune a Mel-Roformer model

A: Few thousand - Unwa.

Mesk for metal dataset at some point had 2135 instrumentals and 1779 vocals (total 3914 tracks)

For fine-tuning of existing models of these archs, RTX 3070 Ti and 4060 (both 8GB) were used by unwa and Gabox respectively (RTX 2000 series don't support flash attention implementation in ZFTurbo's training [repo](#)).

“You COULD train using 8GB of VRAM, it is doable, but not recommended, you at least need 16 or more. Training is difficult because it quickly fills up your VRAM even with gradient checkpointing enabled” - mesk

Training was also tested by unwa and working on RX 7900 XTX 24GB (gfx1100) on Ubuntu 24.04 LTS using Pytorch 2.6 for ROCm 6.3.3, PyTorch 2.6 for ROCm 6.2.4.

“No special editing of the code was necessary. All we had to do was install a ROCm-compatible version of the OS, install the AMD driver, create a venv, and install ROCm-compatible PyTorch, Torchaudio, and other dependencies on it.”

“To install only the minimum necessary items, I first installed PyTorch, then ran train.py many times to install the missing items little by little.”

“Basically, it is almost no different from PyTorch for CUDA.

For example, when specifying a device in your code, you can just use 'cuda' as is.

Also, Flash Attention can be used by setting the environment variable to 'TORCH_ROCM_AOTRITON_ENABLE_EXPERIMENTAL=1'.”

For now, the only [supported](#) consumer AMD Radeon GPUs for ROCm on Linux are: RX 7900 XTX, RX 7900 XT, RX 7900 GRE and AMD Radeon VII (probably a fuller list of GPUs from [here](#) should have working GPUs with ROCm too), but “even right now hipcc in ROCm 6.3.3 has gfx1200 and gfx1201 targets [namely RX 9070 and RX 9070 XT]. You'll still be able to build and run stuff with ROCm. For whatever reason, AMD feels it's not ready to give its stamp of approval.” (EmergencyCucumber905)
E.g. RX 6700 XT 12GB seems to work with ROCm too, but its performance might turn out to be not good enough, seeing how ZLUDA based on ROCm performed (more on ZLUDA later below, it's rather not feasible for training even in its fork state).

“The 7900XTX is great but probably no match for the 9070XT, which will be optimized in time; the 7900XTX certainly has more VRAM, but RDNA4 has FP8 support and greatly enhanced performance at FP16/BF16.

Also, the 7900XTX is a top-end GPU and generates tremendous heat.

The room becomes unbearably hot after running it for a while.” Unwa

Unwa half a year later:

“If you want to use AI properly with an AMD GPU, the MI300X is the best choice.”

“Honestly, I miss how comfortable CUDA is.”

GPU handling

“I've reduced mine's core clock down to 60% with barely any reduction in performance, but it's much cooler now (stays about 55 degrees while training as opposed to 75-80)” becruily (iirc it was on 3090 or Ti).

Bad thermal paste (e.g. MX-2) can dry out in a year in high temperatures up to 80 degrees. Some GPU brands allow changing thermal paste before warranty period ends.

Consider a PC case with good airflow. Thermal pads might degrade after 5 years or when you disassemble them and, in a result, increase temperatures for memory. Then, check specific pad density for your model and replace it. You should be able to monitor VRAM temperature in e.g. GPU-Z. Sometimes it can be good at stock state even longer.

PyTorch supporting ROCm on Windows natively without WSL was unavailable before ([old documentation](#)), and for consumer GPUs is now supported with ROCm 6.4.4 for only RX 7000/9000.

Seeing by e.g. Stable Diffusion WebUI, any potential DirectML forks will be much slower than ROCm ([src](#)), so for now, ROCm is the only reasonable way to go on Radeons (probably ZLUDA forks are still too much behind in development to be any useful (for now official repo supports only Geekbench, and further maintained [fork](#) of the old base doesn't work with e.g. UVR [or we just haven't tried hard enough yet])).

Unwa: "The transition from GeForce to Radeon was not too difficult.

It may be a bit cumbersome to build the environment."

"So far I have not had any problems. Running the same thing appears to use a little more VRAM than when running on the NVIDIA GPU, but this is not a problem since my budget is not that large and if I choose NVIDIA I end up with 16GB of VRAM (4070 Ti S/4080 S).

Processing speeds are also noticeably faster, but I did not record the results on the previous GPU, so I can't compare them exactly."

MDX-Net v2 (not incl. above, see Kim's MDX-Net v2 training repo [here](#)) - lighter and older arch than Roformers, less effective and aggressive, less filtered

Turned out to be easier in picking out proper parameters for training than VR.

In case of e.g. MDX-Net, you take under consideration how big your model is intended to be by fft parameter determining the cutoff of the model, and also in-out channels (size of the channels long story short) - it increases size of the model and intensifies the resources needed for training.

So if you have a smaller dataset, your model doesn't have to be that large.

If you crank up the model size too much for a small dataset, you're putting yourself into a risk of overfitting. It means that the model will work too well on a data which was trained on, but it will not work so well on unknown songs which the model wasn't trained on.

In case of situation of having large database with small model size, there won't be much training at all. It will basically forget features of larger dataset. You need to find a balance here.

Batch size is the amount of samples that are being fed into the model as it's being trained.

Smaller batch sizes will take longer to learn, but you might get a better result at the end.

Larger batch size will make the model not so good, because it has to learn bigger passages at once, but the model will train faster.

You need to tweak, balance out and find what works for you the best for a model you're training. Also balancing things out might be helpful for end users with slower GPUs, or even CPUs [although bigger MDX23C (v3) models are very difficult to separate on CPU, nearly

impossible on the oldest 4 cores and still noticeably slower than MDX-Net models on GPUs like 3050].

The section continues later below.

Overfitting

"Is when a model is still improving on training data but not on unseen data, and if training is push too far, it can even start to perform worse on unseen data.

It's more important issue when you want a model that generalise well", [e.g. targeting only 909 hihats], you want a model which targets one really precise sound (with some variation, but still 909 hihats, so it's not really about generalisation." jarredou

In terms of training, currently Anjok uses A6000 48GB and Ryzen 7 5800, 128GB RAM, 3TB NVME, you need an SSD for training as the training process is intensive for a massive amount of data.

MDX23C

Noticeably slower for separation than MDX-Net, even for GPUs like 3050.

3000 samples of 3-4 minutes length, it's going to take at least for batch size of 8, a month and a half (?on A6000 and MDX-Net). Anjok didn't want to make models too big, having end users with not the best hardware in mind (hence the choice of the older arch).

(here the interview section ends)

Everything should be trained to min. 200 epochs (at least for a model trained from scratch), and better, for 500 (e.g. MDX-Net HQ_2 was trained to 450 epochs). From e.g. 200 upward, the increase of SDR can be very low for a longer time. Experimentally, HQ_4 was trained to epoch 1149, and it slowly, but consequently progressed further beyond. In general, some people train models up to 750 or 1000 epochs, indeed, but it takes longer.

Somewhere at the beginning of 2023, UVR dataset consisted of 2K songs (maybe for voc_ft, can't remember), and probably more for MDX23C, and 700 pairs for BVE model, but in case of vocal model, the one with 7K songs didn't achieve much better SDR results than 2K.

Could've been a problem of overfitting or no cutoff for vocal model or any other problem with dataset creation we will tackle here later.

The best publicly available archs for training instrumentals/vocals which community already used, are:

MelBand Roformer (faster and can surpass MDX23C and BS-Roformer SDR-wise with e.g. Kim config below [and not only], and can sound better and less muddy than BS),
BS-Roformer (very demanding, better for specific tasks), MDX23C (can produce more residues in instrumentals than MDX-Net v2, but can give a bit more clarity), MDX-Net v2 2021 (instrumentals can get a bit muddy even in fullband models, still more residues than in Roformers), Demucs HT a.k.a. Demucs 4 (Anjok failed at training single stem model for it), vocal-remover (VR) by tsurumeso 5 (good for specific tasks like Karaoke/BVE models or dereverb, and for instrumentals it leaves lots of unpleasant residues), VR 6 (now takes

phase under consideration, so there should be less residues, but it's outperformed by newer archs), VitLarge (probably the fastest), SCNet (still faster than Roformers).

I think on example of HQ_3 and 16.xx models, it's safe to say that MDX-Net v2 fullband models have less vocal residues in instrumentals than newer MDX23C arch, but it is also much more muffled, and it depends on specific song what arch will fit the best.

About BS-Roformer, e.g. the model trained by Bytedance didn't include other stem and is obtained by inversion, and initially the results had lots of vocal residues in instrumentals or instruments in other stem, but it can be alleviated by decreasing volume of input file for separation by 3dB (the best SDR among lots of tested values). Generally, viperx models sounds similar to Ripple. The arch itself has potential for the best SDR currently (although currently there's a small difference between the two best Mel and Rofo models SDR-wise - 2024.08.07 and 2024.10 on MVSep.com, while BS models are more muddy).

There are other good archs like BSRNN which is already better than Demucs, and later released SCNet (but the results weren't as good as Roformers, they had more noise, and training wasn't that straightforward as initially thought). It's faster, than BS-Roformer, but probably due to arch differences, rather not better, although it might be still decent in some cases (you can hear the results on MVSEP).

Viperx trained on far more demanding arch (BS-Roformer) with 8xA100-80GB (half of what ByteDance used), on 4500 songs, and only on epoch 74 they already surpassed all previous UVR and ZFTurbo's/MVSEP models, including ensembles/weighted results (more info on that later below).

Viperx made a private model with Mel-Roformer which reached an epoch of around even 3100. He uploaded the SDR results to MVSEP, but it has been taken down since [presumably by viperx himself]. And even then, the result was not above 9.7 unfortunately, achieving results not much better than MDX23C SDR-wise, but with probably bigger dataset.

Later Kim fixed the issues with low SDR in Mel with her config and released the model which become the base of all the fine-tunes by Unwa/Gabox/Syh-Aname (more below).

- "as training progresses, the metrics will improve slower and slower until a point where it's too slow = stop training" - becruily
- I always stop when [loss, avg_loss=] nan

Q: it went back to SDR 20 again

A: "that progress on valid updates per track, so some tracks are 20 SDR some will be like 2 SDR" (frazer)

Q: yea, some are still at 11 etc

A: "train until either avg loss nan or fixed sdr (example: 14 for me) with 5.0e-05 use best checkpoint from that run with 1.0e-05 to get some boost

idk why it works (and if it works im at step 1 rn xD)" - mesk

Q: is it normal that when i use a checkpoint with 12.53 sdr, then restart training, the results at epoch 0 and 1 drop back to 11.77?

A: yes

only if u restart with a higher LR

so i had a checkpoint that was 12SDR trained with 5e-5, then if i trained that with 1e-5, youd expect it to start at like 11.Xsdr

if i had a checkpoint 12SDR trained 1e-5, and i train with 5e-5, id expect it to either start at 12, then drop, then start to increase

keep it going 5e-5 for literally as long as u can

this is called overfitting - just make sure it doesnt do this

its the point where the model begins to not generalize but memorize the training set - happens on finetuning if u train too long

what u do is just keep the redline score if u still have it ([pic](#)) - frazer

Preparing dataset

Let's get started.

First, check the -

Repository of stems - [section](#) of this document.

There you will find out that most stems are not equal in terms of loudness to contemporary standards, and clip when mixed together.

About sidechain stem limiting guide by Vinctekan

The sidechain limiting method might be not so beneficial for SDR as we thought initially, irc it's explained in the [interesting links section](#) with the given [paper](#).

Other useful links:

<https://arxiv.org/pdf/2110.09958.pdf>

<https://github.com/darius522/dnr-utils/blob/main/config.py>

"You can also just utilize this https://github.com/darius522/dnr-utils/blob/main/audio_utils.py and make a script suited to your own, the one already on this repo is a bit difficult to repurpose.

I just concatenated a lot of sfx music and speech together into 1hr chunks and used audacity tho (set LUFS and mix)

oh and then further split into 60 second chunks after mixing them" - jowoon

"Aligned dataset is not a requirement to get performing models, so you can create a dataset with FL/Ableton with random beats for each stem. Or using loops (while they contain only 1 type of sound).

You create some tracks with only kick, some others with only snare, other with only...etc...

And you have your training dataset to use with random mixing dataloader (dataset type 2 in ZFTurbo script, one folder with all kick tracks, one folder with all snare tracks, one folder with... etc.

Then you have to create a validation dataset accordingly to the type of stems used in training, preferably with a kind of music close to the kind you want to separate, or "widespread", with a more general representation of current music, but this mean it has to be way larger.

The only requirements are:

44.1Hz stereo audio.

Lossless (wav/flac)

Only 1 type of sound by file (and no bleed like it would happen with real drums)

Audio length longer than 30s (current algos use mostly ~6/12 second chunks, but better to have some margin and longer tracks so they can be used in future when longer chunks can be handled by archs & hardware)." jarredou

"You can use flac too; saves space (though make them 44.1 / 16-bit / stereo, even if u use mp3's or whatever other format - convert upfront)

validation set however needs to remain in wav with mixture included." Bas Curtiz

"A quite unknown Reaper script to randomize any automatable parameters on any VST/JS/ReaXXX plugin with MIDI notes. It's REALLY a must-have for dataset creation, adding sound diversity without hassle.

<https://forum.cockos.com/showthread.php?t=234194>" jarredou

(Guides for stem limiting moved to the end of the section for archival purposes - rather outdated approaches due to the statements in the paper above)

FAQ

You shouldn't compare training data against evaluation data, while those being the same.

You can use multisong dataset from MVSEP, and make sure you don't have any of those songs in your dataset.

Q: Does evaluation data matter for the final quality of the model?

A: Absolutely not. It's merely indication

SDR measurement is logarithmic, meaning that 1 SDR is 10x difference.

Q: Why I have negative SDR values (based on HTDemucs)

A: Make sure there are no empty stems in any training dataset and or validation dataset

Below is just a theory for now and probably wasn't strictly tested on any model yet, but seems promising

Q: Can you not calculate the average dB of the stems and fit one limiting value to them all?

A: the stems are divide-maxed prior meaning they are made so, that when joined together, they won't clip but are normalized so they will be kinda standardized already based on that, I should be able to just go with one static value for all Example

<https://www.youtube.com/watch?v=JYwsIDs-t4k>

Q: This is great, I actually used this method before with a few sets of stems, before I decided to try sidechain compression/ Voxengo elephant method, but I'm not too sure if I am on the right path. However, I'm pretty sure this only works best for evaluation, if the resulting mixture has consistent loudness like in today's music.

A: Yeah, it's a different approach than compression/voxengo indeed. But the fact it scored high in SDR and UVR dataset is already compressed/elphanted I think it's a good combo to use both in the set, a bit like new style tracks and oldies [so to use both approaches inside the dataset] some tracks in real life are compressed like fuck - some aren't so it mimics real life situation

Q: if it's true that's awesome, with that the model basically has the potential to work in multiple mixing styles, without having to create new data, or changing it, right?

While still adding new data

A: Yeah, since UVR dataset is already compressed - and then add these one of mines with the more delicate way of mastering (incl. divdemax prior)

Q: Does somebody know the best way to make dataset smaller? I have very huge dataset in flac format, so the one idea is to truncate part in the song where is only music without vocals? Also, I can convert it to opus format, does it worse it? Or maybe there is something better that I don't know?

A (jarredou): If you plan to use random mixing of stems during training (so non-aligned dataset), then you can remove all silent parts from stems pre-training, on instrumental it will not change a lot but for vocals it can save a lot of space (h/t Bas Curtiz for the idea)

Q: Currently dataset is aligned, but does this random mixing is standard approach? I am going to train official SCNet model, so maybe it will require modifications for this?

A: <https://arxiv.org/abs/2402.18407> (Why does music source separation benefit from cacophony?)

<https://www.merl.com/publications/docs/TR2024-030.pdf> (same non-columns formatting)
“It thus appears that a small amount of random mixes composed of stems from a larger set of songs performs better than a large amount of random mixes composed of stems from a smaller set of songs.”

If needed, the training script that ZFTurbo has made does handle random and aligned dataset and has also SCNet implementation:

<https://github.com/ZFTurbo/Music-Source-Separation-Training>

Q: As I know, SCNet supports only for inference here.

A: It does training too, ZFTurbo has recently trained a SCNet-large model on MUSDB18 dataset

Dataset types doc

https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/docs/dataset_types.md

(he only didn't update help string)

Creating dataset - guide by Bas Curtiz

(now also [video](#) available)

How to: Create a dataset for training

1. Download any sample pack that is focused around inst/synth x or y.
(sources to seek on: audioz.download, magesy.blog, rutracker.org, freesounds.org, etc.)
2. Use real-debrid.com or alldebrid.com to speed things up DL-wise
(costs a few bucks but worth it,
so better prepare so u can sign up for a free trial or a 3 bucks access for x amount of days)
3. Unzip all (so they are all in a separate individual folder)
4. Convert all to make it consistent (I use <https://www.dbpoweramp.com/> to batch-process)
 - a) Convert all to WAV/16-bit/Stereo & delete any other audio format like AIFF, MP3, OGG.
 - b) Convert all to FLAC (saves space without degrading quality)
 - c) Delete all *.WAV files
5.
 - a) Move all files to root folder of the individual folder.
(I use a python script for that. Hit me up so I can share.)
 - b) Remove empty directories
(I use <https://sourceforge.net/projects/rem-empty-dir/> to batch-process)
6. Rename files by adding a prefix of the folder they're in.
For convenience, add a tag like [ORGAN] or so to it:
example: '[PERC] - Aaroh South Indian Percussion -
AR_SIP_80_percussion_small_nagara_double_rhythm.flac'
(I use <https://www.bulkrenameutility.co.uk/> to batch-process)

7. Sort on length. If below 11s, move them elsewhere.

(I use Mp3tag to sort and move in batch, but Windows explorer is able to do so too)

8. Loop those up till 11 seconds.

(I use <https://www.dbpoweramp.com/> > Loop DSP for that)

9. Move *.flac files into 1 folder (the looped + untouched audio files) - now u can ditch all unneeded files/folders

10. Undupe

(I use <https://www.similarityapp.com/>, <https://www.duplicatecleaner.com/> or <https://dupeguru.voltaicideas.net/> to batch-process)

(optional - only when applicable)

11. Sanitize based on SDR

a) Process the original files with HTdemucs.

Based on whether your dataset should contain bass/drums/other/vocals, set the proper output.

b) Rename the output so it matches the original filename again (using <https://www.bulkrenameutility.co.uk/>)

c) Use SDRCALC.exe like `sdrcalc "c:\organ" "c:\organ-htdemucs" > sdr-organ.txt`

d) copy over the output in the text-file to a GSheet for convenience, to sort on SDR

e) move all unprocessed/original files above a certain SDR to a new folder

(I use a python script for that. Hit me up so I can share.)

12. Review the content on filename and play some you aren't sure, what this filename would sound like.

They can be hit or miss for your specific dataset. So anything that mentions something unusual or so usual, you know it's part of something totally different,

move them elsewhere, to keep the dataset close to what you try to obtain sound-wise.

(example *timpani* is part of percussion, not so much part of a String Dataset)

You could try to cluster the samples upfront with s/w like <https://www.sononym.net/> - also available at audioz.download

13. Zip those and upload, so you can share with those that have the ability/experience to train.

Also needed when you're going to hire a cloud-gpu setup, to copy over the dataset to its server).

(I use sharepoint/onedrive for that, but u can use buzzheavier.com for unlimited storage)

Done.

"In some cases when there aren't clean versions available, you can use a portion of the song where it doesn't have vocals (but has the bleed instruments) and add random clean vocals

it's not aligned dataset, but works for fine-tuning" - becruily even aufr33 admitted that makes models with isolated tracks

"Lossless is always better (and if needed you can use mp3 encoding as an augmentation during training, based on the lossless files)

But as 320kbps have quite high cutoff (20khz or something), it would be less problematic than more compressed audio with hard cutoff at 16khz or 17khz.

I would say that, like for other less regular stuff in your dataset, make it obvious in filename that it's not lossless if you share these files

Q: So... maybe not? IDK I feel like I could make an entire 20 songs dataset out of those, because the best ones aren't lossless

but would it actually be helpful

a lot of leaked stuff is in 320kbps mp3

A: I think while codec cutoff is around 20khz or above it's ok.

Because that will not bias model output results.

With first Roformers models from ByteDance on ripple, that were trained on mixed lossless and 128kbps mp3 with hardcutoff around 16~17khz, we could see that bias in separated audio even when it was lossless input.

Maybe it's a question of balance between lossless/compressed content. I remember the first Ripple bsroformer outputs with these incoherencies in high frequencies while we knew it was trained on mixed lossless/128kbps mp3" - jarredou

Bas: diversity is key we learned from this paper: <https://arxiv.org/pdf/2402.18407.pdf> so I take that literally, and as a starting point.

Q: so it should also have compression for it to work better after all?

A: so diversity is also in audio compression

we don't know for sure, but since my model doesn't seem to perform bad, let's pretend

D: I'm not sure if it's really worth to worsen the quality of already lossy stems to create diversity in the dataset artificially. Yes, the model might behave better at lossy inputs, but people should use lossy inputs only occasionally, so I wouldn't sacrifice quality of lossless inputs that way, plus dataset "can be degraded in many ways on the fly during training with augmentations if needed".

Q: Can the training files for dataset be mp3

I added over 2k tracks and deleted the metadata, it keeps only scanning the original 2k tracks, not the 4k+

A: "I think you can add "mp3" extension to the list there in dataset.py

arg... it's not that simple, there are other places with wav/flac hardcoded..." [jarredou](#)

"If you were annoyed by dataset metadata generation step being slow with MSST, update/do that:

<https://github.com/ZFTurbo/Music-Source-Separation-Training/pull/178/files>

it's like hundred times faster than before" - jarredou

Bas Curtiz' Q&A

“1. “Is it really necessary to have vocals for every track when training?
Depends. Do you wanna make a model that can split vocal from instrumental? Or vocal from whatever 'other' is?

2. Could adding more instrumental-only data be beneficial for variety?
My dataset wasn't 50/50 either. Does that benefit? No idea.

3. Like is it okay to just fill up the dataset with instrumentals to complete it or is there a risk it'll start underfitting on vocals?
Underfitting is always on the lure, so make sure u have plenty of data in general.

4. I know in a lot of songs there are instrumental breaks with no vocals, so I'm just wondering.

Hence, see video "my how to create a dataset":

https://www.youtube.com/watch?v=Wmt_0zu94L8

We ditch the silence parts from start/in between/end, from vocal and instrumental.

If you used dataset type 4, then don't throw away silence, since then there's no cohesion any longer between the inst/vocal or drums/bass/vocal/other or whatever u training.”

Q: Why is my model bleeding the least vocals in the instrumental output?

A: Cause I didn't use pre-processed/cleaned up vocals (I did, but only ~10% of the dataset).
[he refers to his fine-tune exclusively on MVSEP]

Q: What has that to do with the absence of vocal bleeding in the instrumental output?

A: Cause the full spectrum is being used to determine what is a vocal. Even low rumble and potential noise.

Q: So why does yours bleeds less compared to other models?

A: As stated, aufr33 for ex. used pre-processed vocals to train on.

This means, a part of the noise/low rumble is already gone.

So it's trained like that stuff isn't part of the vocal. And if there is, it is dumped into the other (the instrumental output).

This is my gut-feeling why we do hear vocal leftovers in the instrumental output.

Q: So what is the solution?

A: My model 😊 If... you want clean instrumental output.

Q: But yours bleeds percussion/noise in the vocal output though...

A: Yes. That's the side-effect of training on non-processed/cleaned up vocals, I think.

Q: Solution?

A: Fine-tune my model, but this time based on de-noised vocals (and add a shitload of percussion samples).

My prediction is that we then have the best of both worlds:

The current model as-is = great for instrumentals (as described by the community several times due, using non-processed/cleaned up vocals as input)

Another fine-tune based on my current model = great for vocals (due to the lack of noise/low rumble/no bleed percussion)

Warning:

This is based on logic and gut-feeling.

Dill: caught it going to nan again when everything was going smoothly very suddenly between epochs|

ZFTurbo: I think it's never happened to me on SCNet, but often on htdemucs as I remember.

use_amp: false can help

After you can switch back on usage

Dry Paint: I can speak from experience that it only kinda helps

I have the exact same issue with SCNet

making amp=false does solve it but causes the loss to skyrocket to like 3.2e32 around the same time nan loss would appear

Q: Can I put a 4-hour file in one of the dataset's songs? Or should I split it?

A: During training the script uses chunks anyway, so yes you can feed it a 4 hour file (in theory)

A: I did this, but I had to modify the training code to check the file length without loading the whole thing into memory

Leading architectures

MDX-Net (2021) architecture (a.k.a. v2) (<https://github.com/kuielab/mdx-net>). Old.

From public archs, before MDX v3 2023, it gave us the best results for various applications like vocal, instrumental, single instruments models compared to VR arch. But denoise and dereverb/deecho model turned to be better using VR architecture, the same goes to Karaoke/BVE models where in contrary to 5/6_HP, MDX model sometimes does nothing.

In times of Demucs 3 there was also e.g. custom UVR instrumental model trained, but it didn't achieve that good results vs MDX-UVR instrumental models.

Once there was UVR **Demucs 4** model coming up, but the training was cancelled due to technical difficulties. Looks like ZFTurbo managed to train his model for SDX23 challenge and also vocal model, but “[the] problem is that Demucs4 HT [traning is] very slow. I think

there is some bug. Bug because sometimes I observe large slow-downs on inference too. And I see high memory bandwidth - something is copying without reason..."

Spleeter might seem to be a good choice, because training is pretty well documented, but it isn't worth it seeing how these models sound (also it was very first AI for audio separation at the time, and even VR arch is better than Spleeter hence UVR team started to train on VR arch with much greater results than Spleeter).

Your starting point to train MDX model would be here:

<https://github.com/KimberleyJensen/mdx-net>

(visit this repo, it has some instructions and explanations)

ZFTurbo released his training code for other various archs here:

<https://github.com/ZFTurbo/Music-Source-Separation-Training>

"It gives the ability to train 5 types of models: mdx23c, htde mucs, vitlarge23, bs_roformer and mel_band_roformer.

I also put some weights there to not start training from the beginning."

"Set up on Colab is simple:

You only have to create one cell for installation with:

```
from google.colab import drive  
drive.mount('/content/drive')  
%cd /content/drive/MyDrive  
!git clone https://github.com/ZFTurbo/Music-Source-Separation-Training  
%cd /content/drive/MyDrive/Music-Source-Separation-Training  
!pip install -r requirements.txt
```

And a cell to run training:

```
%cd /content/drive/MyDrive/Music-Source-Separation-Training  
!python train.py \  
--model_type mdx23c \  
--config_path 'configs/config_vocals_mdx23c.yaml' \  
--results_path results/ \  
--data_path '/content/drive/MyDrive/TRAININGDATASET' \  
--valid_path '/content/drive/MyDrive/VALIDATIONDATASET' \  
--num_workers 4 \  
--device_ids 0
```

Don't forget to edit the config file for training parameters

You can also resume training from an existing checkpoint by adding
--start_check_point 'PATH/TO/checkpoint.ckpt' \
parameter to the command in the training cell

the checkpoints are saved in the path provided by the :
--results_path results/ \
parameter of the command, so here, in "results" folder

With ZFTurbo's script, mixtures are needed for validation dataset, to evaluate epoch performance" - jarredou

"it saves every checkpoint as "last_archname.ckpt" (file is overwritten at each epoch), and also save each new best checkpoint on validation as "archname_epxx_SDRscore.ckpt".

It also lowers the learning rate when validation eval is stagnant for a chosen number of epochs (reduceonplateau), you can tweak the values in model config file."

Q: what does this gradient accumulation step/grad clip mean exactly?

A: "Accumulation lets you train with a larger batch size than what you can fit on your GPU, your real batch size will be batch_size multiplied by gradient_accumulation_steps.

grad_clip clips the gradients, it can stop the exploding gradients problem

Exploding gradients = model ruined basically, i had this problem with Demucs training, but I used weight decay (AdamW) to solve it instead of grad_clip

I don't think grad_clip uses any resources, but accumulation uses a little bit of VRAM, i don't know the exact number" - Kim

Q: Why can't models have like an auto stop feature or something IDK like if the model stops improving it'll stop automatically
or overtraining, but IDK if models can overtrain

A: Nothing stopping you from adding a thing to stop training after seeing SDR (or whatever) is stagnant, some people even represent it in a chart

A: That's easy to get it done in PyTorch, just use EarlyStopping after the overall validation loss computation and the training will stop depending on the patience you set on EarlyStopping...

- [Colab](#) by jazzpear96 for using ZFTurbo's MSS training script. "I will add inference later on, but for now you can only do the training process with this!"

- Training lots of epochs on Colab might be extremely tasking - for free users they currently only give slow GPU with performance of around RTX 3050 in CUDA but with 11GB of VRAM. It's only good enough for inference.

Q: how can I train a heavier MDX-NET model with a higher frequency cutoff like recent UVR MDX models?

KimberleyJSON:

A: these are the settings used for the latest MDX models you can change them at configs/model/ConvTDFNet_vocals.yaml and configs/experiment/multigpu_vocals.yaml
overlap - 3840
dim_f - 3072
g - 48
n_fft - 7680

These seem to be actually parameters for the last Kim ft other instrumental model, while e.g. half of MDX-UVR HQ models without cutoff has n_fft/self n_fft set to 6144.

Alternatively, see this guide:

<https://github.com/kuielab/mdx-net/issues/35#issuecomment-1082007368>

You also need to be aware of a few additional things:

(Bas Curtiz, and brackets mine)

Few [training] key points:

- If you don't have a monster PC incl. a top range GPU [RTX 3080 min?] (or at work), don't even consider. [smaller models than good inst/vocs with fewer epochs of around 50 might be still in your range though]
- If you don't have money to spent renting a server instead, don't even consider.
- If you aren't tech-savy, don't even consider.
- [If training] a particular singer, [then does it have] highly 100 tracks with original instrumental + vocal?
- IDK, but I don't think that will be enough input to get some great results, you could try though [good models so far have varying genres and artists in the dataset, not just one].
- If you need some help setting it up, Kimberly (yes, she's the one who created Kim_vocal_1 model, based on an instrumental model by Anjok), you can ask her (@)KimberleyJSON.

MDX-Net 2023 (v3) a.k.a. MDX23C (not always better than v2)

<https://github.com/ZFTurbo/Music-Source-Separation-Training>

OG repo:

<https://github.com/kuielab/sdx23>

Lots of general optimizations to the quality while keeping decent training and separation performance. Theoretically the go-to architecture over MDX-Net v2, although currently SAMI Bytedance reimplementation (under VR section below) has much less bleeding results for much more compute intensiveness. It was used for trained models by ZFTurbo. On the same if not better dataset than previous V1 models, it received not much worse SDR than V1 arch for narrowband, but with much fuller vocals, although with more bleeding (also in instrumentals). For fullband, SDR was high enough to surpass previous models, but SDR stopped reflecting bleeding on multisong dataset.

"It doesn't need pairs anymore.

This... is HUGE.

It randomizes chunks of 6 seconds from random instrumental and random vocal to learn from.

In other words, no more need to find the instrumental+vocal for track x.

Just plump in any proper acapella or instrumental u can find.

The downside so far is the validation.

It takes way longer." so you might be able to perform evaluation per only, e.g. 50 epochs.

Dataset structure looks like

- train folder > folders with pairs > other.wav + vocals.wav
- validation folder > folders with pairs > other.wav + vocals.wav + mixture.wav"

Libsnd can read FLACs when renamed to WAV. It can save a lot of space.

I think in the old MDX-Net, we didn't have a model with not worse SDR than epoch greater than 464, although 496 with lower SDR also had its own unique qualities (though more vocal residues at times). Also, frequently training is ended on epoch 300, and might not progress SDR-wise for a long time (maybe till 400+).

<https://cdn.discordapp.com/attachments/911050124661227542/1136258986677645362/image.png> (dead)

(written before Roformers) We may already be hitting the wall SDR-wise, as Bas once conducted an experiment with training a model consisting dataset made of the dataset evaluation and the result was only 0.31 higher than the best current ensemble (although it used lower parameters for separation). Generally, to break through that wall, we may need to utilize multi-GPU with batch size "16 or even 8".

"If you did this experiment with batch size 16 or even 8 you would see much better performance I think" - Kim

"mhm but that requires multi GPU" - Bas

“yeah that is the wall I think” - Kim

- “at least for vocals, using the default 8192 n_fft for mdx23c and reducing hop_length from 2048 to 1024 gave better results (it's InstVocHQ config iirc).

In mdx23c paper, they got better score with higher n_fft/hop_length resolution.

Hop_length means the portion of audio that will NOT be overlapped during STFT processing. So the lower, the higher overlap you get in the end. And a bit like the overlap we are doing during inference (which is different, as applied on waveform, on way larger scale) but in the same way, the higher overlap you use, the higher quality you get, but at cost of more resources, and at some point it gets stagnant (or could even reduce quality too if set too high)” - jarredou

- For “ringing and unpleasant artifacts at subband edges, (...) more of a dip, that seems to get filled progressively along training”

> “It seems like overlapped subbands is the thing (not surprised about it) but the dips are maybe a bug, I'll see if I can improve that.

I'm still experimenting to find best way to alleviate the subband artifacts of mdx23c, but here's an already working fork with separatable depthwise convs (model size is divided by ±6)

DL

- “[The] opposite to BS/Mel Rofos, the more you have sub-bands with mdx23c the less resources needed to train a model (until quick collapse of the model if too much sub-bands used, from what I've experimented until now)” - -||-

Frazer: “wouldn't another fix be to change how the transposed convolutions work? So that the input to the TConv includes the other bands first or last N indices

then crop it out afterward or sum/avg those extra indices

surely if what's happening is that as the bands are downsampled either they drop indices at the edges because it's not a perfect crop or that for whatever reason the latents at the edges don't receive proper gradients and don't optimize - then accounting for that by either expanding input bands or allowing the convs to work on the borders maybe fix it. I like your idea more, it's cleaner, but I'm just trying to come up with some system where it's cheaper than just having bigger bands, yk. I mean, your system would fix it, but I think you'd need to probably drop some indices right near the border on the expanded bands since those would be broken in the same way as they are currently, if that makes sense” artefacts

[the modified code] Confirmed with vanilla OG mdx23c code, with only that change and nothing else.”

Q: How about changing it to a depthwise separable convolution + pointwise convolution?

A: “That's what I've done with my latest version.

Duplications “kinda back, differently but still can lead to similar audible ringing artifacts with some configs. Taking it from the opposite side: if it's the low band high energy the issue, reduce the low band high energy (using preemphasis and deemphasis - pic). It seems to get rid of the duplicated stuff and ringing artifacts at training start.

Colab for training MDX23C model (on free T4)

“Can't train multistem model because of limited resources of Colab, so it's one by one.

It's only 1x Tesla T4, 15GB VRAM so lots of [GPUs] can be [much] better!

I can run batch_size = 8 with it

(with n_fft=2048 instead of 8192 in the model config; other archs are using n_fft=2048 too [Demucs, Rofos...]).

When you use full runtime credits one day, the day after, you get only 1h10min GPU time (2 epochs).

1. It's really boring to do
2. You must have multiple Google accounts
3. You must have a dataset and host it on GDrive and share it with all the accounts (and making it accessible at root for each account)
4. Use this fork of ZFTurbo's training script that is allowing better resuming, which is required for Colab as sessions are deleted after 3h~4h max (often less)
<https://github.com/jarredou/Music-Source-Separation-Training/tree/wandb%2Bresume>
5. Edit this config baseline accordingly to your dataset/needs ([click](#))
6. Set parameters accordingly, gdrive connection and run.
7. When you've burnt all credits from one account, switch and rerun. When you've burnt all credits from one account, switch and rerun. When you've burnt all credits from one account, switch and rerun. When you've burnt all credits from one account, switch and rerun. When you've burnt all credits from one account, switch and rerun. When you've burnt all credits from one account, switch and rerun. When you've burnt all credits from one account, switch and rerun. When you've burnt all credits from one account, switch and rerun. When you've burnt all credits from one account, switch and rerun. When you've burnt all credits from one account, switch and rerun.... and do that 7 loop for weeks.

I've started this experiment just to see if a lightweight mdx23c model could be trained with free Colab, but as I saw it was quickly achieving higher SDR than drumsep on my tiny eval dataset, I'm continuing the training, it's almost at 18SDR now for kick” - jarredou

Q: Any particular reason why the num steps is 1268 instead of 1000? Is that a random number or a calculation?

A: I've read that it's better to use a multiple of the number of tracks in the dataset (here 317 * 4) To avoid that some of the tracks are used more than others at each epoch. (jarredou)

jarredou/Aufr33 drumsep model was trained with 3 seconds chunks.

- Here you can find [Colab](#) made by yukunelatyh (dead)
- [Colab](#) by jazzpear96 (with the OG MSST repo; maybe you could just replace the link)

I think later I quote jarredou on training with extremely low parameters configs on other archs as well.

- [Here](#) the community guide one user on training on 3060 12GB ([invite](#))

JFI -

Multi Source Diffusion

<https://github.com/gladia-research-group/multi-source-diffusion-models>

Some results posted by Bytadance were labelled as “MSS” but it’s rather not the same arch. In the original MSS paper above, only Slakh2100 was used.

ByteDance probably expanded it further, and had it was said they had issues with their legal department with making their work public, so they can equally use unauthorized stems just like us, or looking for ways to monetize their new discovery for TikTok, as their company largely invest in GPUs lately, so something might happen maybe in the end of the year, and maybe it will be released in their exclusive service (Ripple and Capcut were released later indeed). TBH, it’s hard to get a good model using only public datasets. For public archs, it’s even impossible. They probably know it too, so it’s kinda grey zone, sadly, and model trained later for Ripple was probably done from scratch and contains only lossy files for training from now on.

Bytedance (Ripple too?) was said to train on 500 songs only + MUSDBHQ

BS-Roformer

(one of) The best, but the slowest tested arch out of the all in this doc SDR-wise. Once considered as SOTA (state-of-the-art algorithm), but it has its own caveats, like very strong denoising (which is double-edged sword and might give too muddy results frequently), but using Mel-Roformer and proper config tweaks and prioritizing stem there helped for the muddiness issues. Also, Mel-Roformer has a bigger SDR according to the Mel paper (with an exception for bass), and it’s better for vocals than BS. Plus, Mel seems to handle creating duet singing separation model better. “BS (...) uses more VRAM; unlike Mel, BS has no overlap between bands, so VRAM usage and model size are smaller.” Unwa More below.

“I find it cute how they call the Transformer based models (which destroy the older convnets) “Roformers” because they use RoPE embeddings. By that naming scheme, all llama-like models are Roformers too...” kalomaze

“By the way, it wasn’t us who started calling Transformer using RoPE “Roformer.

<https://arxiv.org/abs/2104.09864>

MelBand-Roformer and BS-Roformer may also be considered as generative objective models in a sense. The goal of these models is to generate a mask to extract the desired stem from the mixed source” unwa

<https://github.com/lucidrains/BS-RoFormer> (it incorporates both BS and Mel variants, implemented in MSST training repo)

It’s safe to say it’s SAMI Bytedance arch from MVSEP chart reimplemented from their paper - done by lucidrains.

Arch papers:

BS (band split): <https://arxiv.org/abs/2211.11917>
Mel (mel scale): <https://arxiv.org/pdf/2310.01809.pdf>

“Bytedance didn't give any info of training duration for these scores, but in their last [ISMIR2024] paper for Mel-Roformer:

<https://arxiv.org/abs/2409.04702>

with L=12, they get 12.08 SDR on vocals with Musdb-only by using :

8 Nvidia A-100-80GB GPUs with batch_size=64, and the training stopped at 400K steps (~40 days.)”

32x V100 will require two months of training (most likely for 500 songs only + MUSDBHQ)
“It's better to have 16-A100-80G”, viperx trained BS-Roformer with 4500 songs on
8xA100-80GB and after 4 days achieved epoch 74, and on epoch 162 achieved only 0,0467
better SDR for instrumental.

ZFTurbo having 4x A6000 gave up training on it, having to face 1 year of training time.

After the BS variant, [Mel-Band RoFormer](#) based on the band split was released (“Mel-Roformer uses a Mel-Band spectrogram whereas BS-Roformer doesn't”), which is faster. Initially it achieved worse SDR than BS-Roformer than on the paper. But it was till Kimberley Jensen released her new Mel model, and by the occasion, tweaked config in a way that it made the SDR on par with BS variant, but presumably, by training on even smaller dataset.

Later, viperx trained drums only model, both on BS and Mel-Roformer, and BS-Roformer was still slightly better, but there wasn't such a difference between both anymore (12.52 vs 12.40 SDR).

“Main diff is that BandSplit is using linear Hertz scale for frequency splitting while MelBand is using Mel scale (which is a more close representation of how humans are hearing frequency distances than linear Hertz scale).

MelBand matrixing is by design using overlapping frequency bands, while with BS, there's no overlap in the frequency range.” jarredou

“remember, just because the melscale is more perceptual, it doesn't necessarily translate into the neural net learning the representation better. It might be a good idea to use a modified zipformer” frazer

Kim's Mel training config made for H100 (model on x-minus was trained for 3 weeks) which viperx probably used later for drums model or reworked for his GPU.

Kims says 5.0e-05 is already low enough learning rate, setting it too low may make it too slow to train. Kim “also said that during the end of her training the loss would plateau but the SDR was improving”, “no patience, no LRreduceonplateau at all. While, if set incorrectly, it can ruin your training very quickly” - jarredou

With the unwa's inst v1 model it turned out, prioritizing stem in the config matters a lot, so depending on which model you want (other, instrumental, null, multistem like in duality models or vocal) that's what you should set in the config. Although, prioritizing stem to instrumental gave a noise similar to VR or MDX-Net v2 models (but not the same). We ended up with the [code](#) based on Aufr33 idea, recreated by Becriuly, that copies phase from Mel-Kim model which is deprived of the noise and doesn't prioritize vocal stem like unwa inst 1/1e/v2 models, and it gets rid of some noise in those models. Aufr33 own implementation is added in ensemble on x-minus.pro/uvrline and in UVR latest patches there's becriuly [script](#) rewrite.

According to mesk, fine-tuning a fine-tuned model might be a worse solution than simply fine-tuning the Kim's model (from experience on training on genre-oriented dataset like metal which Mesk tried to train).

"I got an error when I set num_stems to 2." unwa

You can use "target_instrument: null" instead, which is also required for multistem training like on [this](#) example ~jarredou

"increasing num_stems increases model size" "multistem is like having multiple checkpoints in one file (1 for each stem). All model types work like that with ZFTurbo's script AFAIK"

use_torch_checkpoint: true

in the current MSST repo will reduce VRAM usage.

Using various chunk_size during different stages of the training can be helpful, and also using different dataset sizes based (e.g. leaving only more clean or official at certain points).

First BS-Roformer models were trained on ZFTurbo dataset, later viperx trained on his own, presumably larger dataset (and possibly better GPU) and achieved better SDR, then another model was made from fine-tuning viperx model on ZFTurbo dataset, and Kim's Mel model was trained on Aufr33 dataset from UVR, later Unwa trained on Bas Curtiz' dataset (?too).

You can use ZFTurbo code as base for training Roformers:

<https://github.com/ZFTurbo/Music-Source-Separation-Training>

"change the batch size in config tho

I think ZFTurbo sets the default config suited for a single a6000 (48gb) and chunksizes" joowon

So, to sum up, BS-Roformer is the best publicly available arch SDR-wise for now (and in practice, Mel-Roformer scores a bit lower on the same datasets with the same public code

we currently have), although both are very, very demanding compared to MDX23C or MDX-Net v2 or VR (voice-remover by tsurumeso).

E.g. Aufr33 said that BS Roformer turned out to be better for training BVE model. “Although I get about zero or even negative SDR with both, the BS does the job better.

Maybe it's not the architecture but the augmentation, I disabled it for BS ”. Some other explanations:

“In BS-Roformer they don't do any downsampling or compression” hence it's so slow to train.

“I've noticed that it separates mono (i.e., panned in the center) harmonies well if the two voices belong to different people, and much worse if it's the same singer (even though they differ by a third interval).

This leads me to believe that the BS architecture would work well for separating female and male voices.” (after some problems with SDR being in the region of 0 or 1) “I have another thought. Roformer works differently with stereo, not like VR. It's like it partially merges channels, which is bad for backing vocal detection. It seems to me that using a stereo expander would help.” Aufr33. Eventually he later started the training Mel from scratch on 4x RTX 6000 Ada using stereo expander and that lower training rte, and eventually it started to increase SDR, but later negative value was showing, and he continued training BS and SDR was above 1, but it stopped progressing, “It seems that 525 songs is not enough to train a [BS/Mel] model.”

Some things here can be outdated already, as some optimizations were introduced to the training code (read more below):

—

Time-wise, BS vs Mel, instead of 16x a100 in BS-Roformer, it might be like 14x a100 to train in decent time, but at best, without the config tweak, SDR will be only in pair with MDX23 and MDX-Net archs v2 archs, and BS-Roformer will achieve better SDR than Mel-Band.

Might be some issue in Mel-Band Roformer reimplementation, maybe paper lacking something. Only in BS-Roformer some of the original authors from Bytedance took part in some reviewing of the reimplementation code made by lucidrains.

On Mel-Band, epoch 3005 took 40 days on 2xA100-40GB with the previous viperx model.

Viperx trained their own vocal model, using BS-Roformer on +4500 songs (studio stems * +270h) using 8xA100-80GB, and only on epoch 74 they almost surpassed sami-bytedance-v.0.1.1 result (which was actually multistem model iirc), achieving 16.9279 for instrumental, and 10.6204 for vocals.

With epoch 162, they achieved 16.9746 and 10.6671, which for instrumental, is now only 0.0017 difference in SDR vs v.0.11 result.

Training settings:

chunk_size 7.99s

dim 512 / depth 12
Total params: 159,758,796
batch_size 16
gradient_accumulation_steps: 1

Since epoch 74 there were “added +126 songs to my dataset”

Training progress:

<https://ibb.co/1zfFX82>

Source:

https://web.archive.org/web/20240126220641/https://mvsep.com/quality_checker/multisong_leaderboard?sort=instrum

https://web.archive.org/web/20240126220559/https://mvsep.com/quality_checker/entry/5883

It sounds similarly to the Ripple model.

“7 days training on 8xA100-80GB”: $7 \times 24 \times 15.12$ (runpod 8xa100 pricing) = \$2540.16”

viperx trained on dataset type 2, meaning that he had 2 folders:
vocals and other and no augmentations

“For more detailed infos, you can read ZFTurbo’s doc about dataset types handled by his script

https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/docs/dataset_types.md

viperx trained on faster Mel-Roformer arch variant before, and on epoch 3005 trained 40 days on 2xA100-40GB with 4500 songs, he achieved only 16.0136 for instrumentals, and 9.7061 which is in pair with MDX-Net voc_ft model (2021 arch).

“Each epoch [in Mel-Roformer] with 600 steps took approximately 7 to 10 minutes, while epochs with 1000 steps took around 14 to 15 minutes. These are estimated times.

Initially, I suspected that the SDR was not improving due to using only 2xA100- 40GB GPUs. After conducting tests with 8 x 80GB A100 GPUs, I observed that the SDR remained stagnant, suggesting that the issue might be related to an error in the implementation of the Mel-Roformer architecture.” [More info \(copy\)](#). Probably it was the issue (at least partially?) fixed by Kim config tweaks.

Later, the viperx’ BS-Roformer model was further trained from checkpoint by ZFTurbo, and it surpassed all the previously released models, and even ensembles, at least SDR-wise. Then it was finetuned on different dataset. Still, as all Roformers, it might share some characteristic features, like occasional muddiness, and filtered sound at times, but Mel variant seems to be less muddy.

More insides:

<https://github.com/lucidrains/BS-RoFormer/issues/4#issuecomment-1738604015>
<https://media.discordapp.net/attachments/708579735583588366/1156700109682262129/image.png?ex=6516952c&is=651543ac&hm=988a5acc32f075988c1701d41c2090321a25955c4ffedd64516e0062fa1002e0>
<https://cdn.discordapp.com/attachments/708579735583588366/1156700305069707315/image.png?ex=6516955b&is=651543db&hm=bf5737f95f3a93fd3e3a23a679e2ad0031e0feb6c622fb85eafa053ed483e08>
<https://media.discordapp.net/attachments/708579735583588366/1156700453585829898/image.png?ex=6516957e&is=651543fe&hm=06ed766b39c3c7f4a8329420a22bcc572e856116a6e1cea030d158c984c46825>

More hints/FAQ for training Roformers

ZFTurbo:

"1) Best [BS-Roformer] model with the best parameters can be trained only on A100, and you need several GPUs. The best is use 8. It reduces possibilities of training by enthusiasts. [later it was found out that checkpointing decreased VRAM usage allowing using probably more modest GPUs]

All other models like HTDemucs or MDX23C can be trained on single GPU. Lower parameter BS-Roformers don't give the best results. But maybe it's possible. Solution: We need to try train smaller version which will be equal to current big version. Lower depth, lower dim, less chunk_size. We need to achieve at least batch 4 for single GPU. Having such model can be useful as starting point for fine-tuning for other tasks/stems [perhaps Unwa's 400MB exp value residue model meets that requirements).

2) I also noticed a strange problem I didn't solve yet. If you try to finetune version trained on A100 on some cards other than A100 then SDR drops to zero after first epoch. Looks like "Flash attention" has some differences (???).

3) Training is extremely slow. And I noticed BS-Roformer more sensitive to some errors in dataset.

[probably for 4090]

chunk_size: 131584

dim: 256

depth: 6

I think these settings can give batch_size > 4

For example, I can't finetune viperx model on my computer with 48GB A6000 because the model is too large.

chunk_size is what affect model size the most, I think. And I saw it's possible to get good result with small chunk size.

I put the table here:

https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/docs/bs_roformer_info.md

[see also <https://lambdalabs.com/gpu-benchmarks> batch size chosen in Metric, fp16, but ZFTurbo said that training on fp32 is also possible]

The 0 SDR issue was later fixed: “the non A100 issue is fixed with latest torch, but I think the general rule is that batch size 1 (like in my case with 3090) won't give good results on Roformers.

But I've been doing batch size 2-4 with A6000 and A100 and no issues there.

But the model is also large, when I finetuned a smaller Mel-Roformer the 3090 worked there”

“Full sized Roformers are very heavy to train and unless you have crazy hardware like A6000 (the very minimum), A100 etc, training from scratch will take months to get a good model (with SDR similar to current ones, and this is without considering the dataset). Maybe someone with 4090 can give more insight, but I personally can't train/fine tune a full sized Roformer model with my 3090, it's way too weak, I'd have to make the model smaller meaning the quality won't be as good as the original checkpoint” becruilily

Q: Can we change model size of existing model and fine tune it? Or it must have been trained from scratch with the same chunk size

A: 1) If you just decrease chunk size, it will work almost the same as with larger (as I remember)

2) If you decrease dim or depth, score will drop very much”

Don't forget, each time you change something in dataset, you have to delete metadata_x.pkl file to create new database on training launch taking into account new changes (it made me become crazy during my first tests when forgetting to delete it)

I've just checked ZFTurbo's code, and for dataset type 2, the ".wav" extension is still required for the script to find the files (it doesn't work with any other)

Q: It is possible to finetune on 3080 Ti 8GB? unwa did it [on 3070 Ti 8GB]

A: “By reducing the chunk_size and using AdamW8bit for the optimizer, I was able to train even with 8GB of VRAM.” Usually such fine tune on inferior GPU was decreasing SDR. Here it only dropped by 0.1 SDR after few thousands steps of training (so only a few epochs) on musdb18hq+moises+original dataset (41 songs (In total, 432 tracks, 16.5GB, FLAC, very small dataset, OG was trained on 5K songs) ~becruilily

Used config to finetune:

https://drive.google.com/file/d/1gK1_n_bpRHD1i02VA2bgUc3TrpEJUcg9/view?usp=drive_link

train.py with optimizer (probably it's already integrated into MSST code):

<https://drive.google.com/file/d/1jLSTDajYxZRSb5wLOwyVuRJrayNLIWUZ/view?usp=sharing>

Changes were pushed to ZFTurbo training dataset, but the optimizer turned out to not save too much VRAM ([diagram](#)).

“adamw8bit” for training.optimizer in config). Also added possibility to provide optimizer parameters in config file using keyword ‘optimizer’.” ZFTurbo Optimizers explained later below.

And it’s not enough to point the model trained by unwa turned out to have the same 0 SDR issue becruily had, but unwa didn’t notice it due to lack of validation dataset. So inference worked correctly despite 0 SDR, but the model was getting worse gradually during finetuning.

Frazer suggestion (probably already implemented by ZFTurbo):

“I think the issue with the checkpointing / 0 SDR bug is due to

|At least one input and output must have requires_grad=True for the reentrant variant. If this |condition is unmet, the checkpointed part of the model will not have gradients. The |non-reentrant version does not have this requirement.

So I think the fix is either assigning

`x.requires_grad=True`

in train.py to the batch tensor before passing into the model or passing

`use_reentrant=False`

to all torch.utils.checkpoint.checkpoint calls

I think it’s probably better to use the non-reentrant variant, since Pytorch will default to this in later versions

<https://pytorch.org/docs/2.1/checkpoint.html#torch.utils.checkpoint.checkpoint>

also this point here looks interesting to test whether it causes a significant performance hit or not

|The logic to stash and restore RNG states can incur a moderate performance hit depending |on the runtime of checkpointed operations. If deterministic output compared to |non-checkpointed passes is not required, supply preserve_rng_state=False to checkpoint or |checkpoint_sequential to omit stashing and restoring the RNG state during each |checkpoint.”

- It was discovered, that using different chunk_sizes at various stages of training can be beneficial (iirc esp. for training time at initial stages without much SDR sacrifice).

- “I added code to train using accelerate module:

https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/train_accelerate.py

It’s useful on multi GPU systems. In my experiments, speed improved ~50%.

~1.57 sec per iteration goes down to ~1.07 sec per iteration.

But I think the script has some flaws - my validation score during training is lower than in reality. I didn’t find the reason yet.

Also, script allows training across multiple machines without changes in code.

More information here:

<https://huggingface.co/docs/accelerate/index>

<https://huggingface.co/docs/accelerate/quicktour>" - ZFTurbo

- What unwa said later in October 2024 what allowed the fine-tunes to be made on 8GB VRAM and RTX 3070, is they used gradient checkpointing - “time and space are a trade-off, and [gradient checkpointing](#) saves memory at the expense of computation time”

Q: And you don't get the 0 SDR issue?

A: “Yes. I'm using L1Freq metric now. As it turns out, it was not a failure to train properly, but just a problem with the validation function.” - unwa

[More](#) on the issue. “The validation issue should be solved now [in valid.py] but not sure if it was the same issue ZFTurbo was facing”

- “<https://huggingface.co/pcunwa/Mel-Band-Roformer-small>

In the experiments with the Mel-Band Roformer big model, it was confirmed that increasing the number of parameters for the Mask Estimator did not improve performance.

Therefore, I conducted an experiment to see if I could reduce the number of parameters while maintaining the performance.

It even runs well on 4 GB cards due to the reduced memory used.” - unwa

ZFTurbo: “I looked onto unwa code for small Roformers. Roformers have one parameter mlp_expansion_factor which couldn't be change[d] from config and fixed as 4. It uses a lot of memory:

```
| └─MaskEstimator: 2-8           [1, 1101, 7916]      --
|   | └─ModuleList: 3-73          --                  201,465,304
```

if set to 1 (memory reduced 10 times 200 MB to 23 MB):

```
| └─MaskEstimator: 2-8           [1, 1101, 7916]      --
|   | └─ModuleList: 3-73          --                  23,836,120
```

Yesterday I already added in my repository possibility to change mlp_expansion_factor from config.

Unfortunately, while overall number of weights is greatly reduced, it won't allow to greatly increase speed or batch size for training:

Mel band (384, 6, chunk: 352800)

mlp_expansion_factor = 4, normal training: batch size: 2 (1.27 s/it)

mlp_expansion_factor = 3, normal training: batch size: 2 (1.20 s/it)

mlp_expansion_factor = 2, normal training: batch size: 2 (1.19 s/it)

mlp_expansion_factor = 1, normal training: batch size: 2 (1.16 s/it)

Even in last case, batch size 3 is not possible

- I will check how "checkpointing" method works

OMG checkpointing technique impressed me a lot!!

It reduced required memory ~20 times

Mel band (384, 6, chunk: 352800) Single A6000 GPU 48 GB

mlp_expansion_factor = 4, normal training: batch size: 2 (1.27 s/it) - 0.635 sec per image

mlp_expansion_factor = 3, normal training: batch size: 2 (1.20 s/it) - 0.600 sec per image

mlp_expansion_factor = 2, normal training: batch size: 2 (1.19 s/it) - 0.595 sec per image

mlp_expansion_factor = 1, normal training: batch size: 2 (1.16 s/it) - 0.580 sec per image

mlp_expansion_factor = 1, low mem training: batch size: 2 (0.60 s/it) - 0.300 sec per image

mlp_expansion_factor = 1, low mem training: batch size: 40 (6.32 s/it) - 0.158 sec per image

mlp_expansion_factor = 4, low mem training: batch size: 40 (6.87 s/it) - 0.171 sec per image

So my batch size for single GPU grew from 2 to 40

[So maybe there won't be necessary to train a good model without x16 A100]

And speed per image increased ~ 4 times.

Ok changes in repo. To train with low memory, you need to replace only one thing:

mel_band_roformer -> mel_band_roformer_low_mem. And increase batch_size in config. All weights and model parameters are the same.

The same can be done for BSRoformer as well (need to add).

With current improvements for memory, we can try big depths for training

BS-Roformer with depth 12 now has batch_size: 32

We can add sum of inputs, for example for every 3 blocks of freq and time transformer blocks.

Or even use DenseNet approach.

I found a problem. If internal loss calculation for Roformers is used based on FFT. Batch size reduced to 12 instead of 40.

Loss calculation inside the model consumes too much memory." - ZFTurbo

unwa: The core of the model is the Roformer block, and the Mask Estimator probably did not need that many parameters.

According to the paper, the entire model has 105M parameters, whereas when the mlp_expansion_factor is 4, the Mask Estimator alone exceeds that number by a wide margin. Sorry, I forgot about this, I had removed 4096 from multi_stft_resolutions_window_sizes.

Q: Is the speed also faster despite the use of gradient checkpointing because memory bandwidth was the bottleneck?

A: I don't know, maybe yes. Now we need to ensure it doesn't affect the training process.

And the quality of models stays the same.

So there is no need to decrease mlp_expansion_factor from 4 to 1 currently (may be later for train new models).

I will add possibility to train with low mem in my repo in several minutes

I think speed up is because of the benefits of large matrix multiplication (because it's calculated for 40 images at the same time).

Q: Can the gradient checkpointing be applied to other architectures? For example SCNet

A: I think yes, but maybe with lower benefits.

- “One thing to keep in mind too, is that Rofos are using flash attention by default, which is not compatible with all GPUs (not with small ones), and this flash attention is greatly reducing training duration, from what I get. Non-compatible GPUs use lower performing attention.

<https://github.com/Dao-AILab/flash-attention>

Supported GPUs: <https://imgur.com/a/QGtSuKR> (src) - half a year later, Flash Attention 2 is still unsupported on RTX 2000 series, and FA3 beta is available for Hopper GPUs (e.g. H100)

Q: The training script defaults to memory efficiency unless you use A100 though, does this mean ZF didn't implement flash att for the other GPUs listed there

A: I don't know, I think it's more related to the custom flash attention module made by lucidrains, he doesn't use this flash attention repo. frazer and unwa had a discussion about that in devtalk some days ago, I didn't understand everything, just learnt that it was custom implementation

anvnew made a pull request (“Enable flash attention for compute capability >= 8.0” so GPUs from RTX 3000 onward)

<https://github.com/ZFTurbo/Music-Source-Separation-Training/pull/52> (merged already)

Q: Is adam or adamw better for Mel-Roformer?

A: “This is not particularly relevant if weight decay is not used.

In Adam it was implemented with L2 normalization; in AdamW it is implemented in its original form.” - unwa

Bas Curtiz later conducted some experiments on the best optimizer, and the winner is: Prodigy (with LR 1.0) on the same amount of steps (~14K) and ~20 hours in (at least when training from scratch). [More](#)

- Unwa’s

“1) v1e model was trained with a custom loss, if you don't use the same or similar multi resolution loss, your results will be bad

2) because of [the] 1[st], SDR is not the best metric to keep track of how good the model is, SDR will be lower when training with mrstft losses

3) you must set the target instrument to other, and yes you need more vocals” becruily

Unwa “reduced mask estimator depth from 3 to 2, and he said that it didn't hurt the quality but reduced the size significantly. also there's some additional line 'mlp_expansion_factor: 1' in his small model config. Maybe that helped somehow too.”

“The mask estimator is already 2 by default on Kim model for example (and in bort's config too)”

“I have unwa's beta and duality models and on batch size 1 they don't eat that much memory”

“inference maxes out the GPU mem and sits at 0% ckpt file is 3GB.

Moral of this story is try running inference before you get to epoch 80"

Q: Does anyone here know how much VRAM is required to train a Roformer model with the same specs as Unwa's and Kim's models?

A: ZFTurbo has made this small benchmark some months ago with BS-Roformer:

https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/docs/bs_roformer_info.md

and [newer](#) one for Mel-Roformer

ZFTurbo experimented with 4 stems Mel model creation on MUSDB18, and struggled with getting good results like in the paper. [Here](#) he evaluated various parameters and achieved SDR.

Eventually, he released checkpoint with different parameters [here](#).

Later, he trained BS-Roformer 4 stem model on MUSDB18.

Q: Is it not possible to emphasize both SDR and Aura scores?

A: "Training the model using [I1_freq and AuraMRSTFT] metrics is prone to phase problems. Like my 5e and v1e models" Unwa

- "You can enable spectral phase with auraloss (never really tried)" J.

A: It makes it less stable in training; Loss is more likely to be NaN
It is difficult to optimize a phase spectrogram that looks like noise.

A: "You don't emphasize a model with them, they're just metrics, and you're tracking how well each epoch scores

the metrics will go up and down, but it doesn't specifically emphasize the chosen metric B."

"Phase also has a significant impact on sound quality and cannot be ignored. However, these metrics ignore phase; models that emphasize fullness are those that compromise phase optimization to some degree in favor of optimizing the amplitude spectrogram, and clearly these metrics favor such models.

I would endorse the log_wmse metric.

It is a relatively new time-domain metric over SDR and SI-SDR that is not overly sensitive to low frequencies like SDR and can accurately evaluate silent intervals.

In addition, time-domain metrics can be evaluated for both amplitude and phase."

- **LoRA training repository** by frazer - for only Mel and BS models at the moment
(merged into ZFTurbo's MSST training repo already)

<https://github.com/fmac2000/Music-Source-Separation-Training-Models/tree/lora>

"LoRA could specialize in a particular singer or genre."

"You can use LoRA as a replacement for full-weight fine-tuning

for now, all I can say is that it'll be faster to train and way more memory efficient than fine-tuning - whether the performance competes with full fine-tuning up is yet to be determined"

"did a small test, and it achieved results in hours that took me days to achieve with A100"

"I trained it for a week - 180 epochs

I took Kim Melband and just trained a LoRA on the standard MUSDB - it took SDR vocals from 11 to around 12.3 after 100 epochs on batch_size = 1.

I didn't save each epoch so that 100 epoch voc12.3 isn't there.

(...) my bets it sucks ass since it's overtrained" frazer

- Bytedance and Asriver prepared some **enhancements** for Roformer arch, and already published white paper which will be presented on ISMIR2024:

<https://arxiv.org/abs/2409.04702>

lirc, sami-bytedance-v.1.1 model is already some derivative of above with higher parameters, settings and from what I remember, trained on 16xA100, which cannot be even rented. Bas tried to train a model (model_mel_band_roformer_ep_617_sdr_11.5882) better than that, by just training purely on mutlisong dataset, but he couldn't surpass that score.

- At the end of December 2024, Lucidrains implemented "**Value Residual Learning**" into his BS-Roformer [repo](#), based on the following paper:

<https://arxiv.org/abs/2410.17897>

"The paper argues that this mechanism can reduce the over-focus of attention and further reduce the vanishing gradient problem."

Unwa trained a small 400MB experimental instrumental [model](#) based on it. Doesn't work in UVR.

Now it's also added into ZFTurbo MSST repo.

- CFM

Before the middle of 2025, somewhere in probably #dev-talk of our Discord server, jarredou "posted a repo from the ByteDance team which explained a method but didn't implement it in the code"

Becruily: "Gemini did a quick pseudo implementation, and it trains 4 times slower"

Frazer :"chat models can't even code javascript let alone AI"

B: yeah the hallucinations and unnecessary edits are insane but 2.5 pro is prob the best I've seen as long as it has all the context

F: "just standard CFM it doesn't need something audio specific for it to work
time = torch.randn(B, device=device, dtype=dtype).sigmoid()
time = time.view(B, 1, 1)

```
x = time * OUTPUTTENSOR + (1 - time) * INPUTTENSOR  
target = OUTPUTTENSOR - INPUTTENSOR
```

```

condition = self.timeembedder(time)
condition2 = self.someconditioningembedder(otherinfo here)
somecondition = condition + condition2

x = model(x, somecondition)

loss = F.mse_loss(x, target)

```

you need time embeddings for it to work, but it's copy-paste 10 minutes work"

Q: what about the train/inference/valid, don't they have to be adapted to cfm as well or nah
 F: yeah - so inference will have to be something like this

```

def evaluate(self, x: torch.Tensor, N: int = 50) -> torch.Tensor:

    x = x.transpose(1, 2) #X must be B T C (if that's what bsroformer uses idk)
    t_span = torch.linspace(0, 1, N + 1, device=x.device, dtype=x.dtype)

    dt = t_span[1] - t_span[0]

    for t in t_span[:-1]:
        k = self.forward_eval(x, t)
        x = x + dt * k

    x = x.transpose(1, 2)
    return x

```

where you then define a new eval forward

```

def forward_eval(self, x: torch.Tensor, time: torch.Tensor):
    B, T = x.shape[0], x.shape[1]

    time = torch.full((B,), time, device=x.device, dtype=x.dtype)
    time = time.long()
    time = self.embedding_time(time).view(B, 1, -1)

```

```

conditioning = time #can add in other embeddings here
x = self.forward_model(x, conditioning)
return x

```

if u want different losses you have to adapt them, what ur trying to do is to iteratively change the input by adding something to the latent across N timesteps"

Q: Is there a version of Mel-Roformer without all the nonsense value residuals and stuff like that

A: Yeah zturbo separated those a few weeks ago on a new file

[“\[https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/models/bs_roformer/mel_band_reformer.py\]\(https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/models/bs_roformer/mel_band_reformer.py\)](https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/models/bs_roformer/mel_band_reformer.py)

or I guess lucidrains file is as bare/og as it gets”

Later, frazer's blocks are written somewhere in #dev-talk.

- More Roformer training insides in far right of [this](#) Bas Curtiz' sheet (phase image may overlap the text, navigate by arrows to read it down below).

“bleedless/fullness metrics are stft magnitude-only based and as they are discarding the phase data, they have some kind of blind spots.

I guess this noise could be also reduce by using higher n_fft values for model (smaller bins, finer freq separations, but way more ressources needed to train models)” - jarredou

Q: High n_fft values increase the frequency resolution but decrease the time resolution

A: Yeah. But Roformers are trained with 2048, it's not high value. MDX23C original models are using 8192 by default, with improved results compared to lower values. (ZFTurbo has made some tests back then comparing different n_fft/hop_length config)

Q: does it matter that much when the model is trained with multi resolution 🤔 it should cover both low and high nfft values

A: n_fft=2048 is around 21.53 Hz resolution per bin (on linear scale)

while n_fft=8192 gives 5.39 Hz resolution per bin

This should benefit most of the stems types (maybe not drums and transient heavy content tho)

We don't have multi-resolution arch yet, even if it could be interesting. Only the loss in multi-resolution, not the model. - jarredou

Q: Does finetuning need to reach hundreds of epochs?

A: “Not always. Depends on the amount of data - if it's small you could literally get away with training for a day or half a day” 40 hours is enough for just fine-tuning (frazer)

Q: Is there something wrong with the config I'm using? I've already tried training 3 times, and the pattern is always the same, after the training metrics improve about 6 times, it becomes really difficult for them to improve any further in the next epochs. i've already tried changing the lr to 5e-5 and 1e-5, but it's still the same

A: “try fiddling with adam's betas - change from betas=(0.9, 0.999) to betas=(0.8, 0.99) adamfactor might be a cool thing to test as well add this into the config, it might work”

optimizer:

betas: (0.8, 0.99)

(frazer)

Ident like in a new line equally with training and inference, placed below training (so actually none)

Q: So far the metrics keep improving, unlike the previous training. It didn't just cap like last time.

Q: After e.g. 14 epochs, you can try out restarting the checkpoint from the best SDR weight because it will do this "it's weird. The optimizer for whatever reason gets fucked up after a while": [pics \(src\)](#)
(frazer)

A: how to setup this graphs?

—wandb_key YOURKEY

- Loss 0 issue

<https://discord.com/channels/708579735583588363/1220364005034561628/1416389629866672261>

Q: is this right? i resumed yesterday's training, but it went back to epoch 0

A: yea that's normal because you're restarting, i think there's something to make it so that it saves that data but IDK

- 2nd epoch takes a long time issue

<https://discord.com/channels/708579735583588363/708912597239332866/1415577958646808636>

"When VRAM runs out and begins using main memory, processing becomes extremely slow."

Q: how do i make more metrics show up like this during training?

A: --metrics sdr si_sdr log_wmse l1_freq aura_stft aura_mrstft bleedless fullness

- "inverts can bring good quality shit if you clean the remaining instrumentals in your vocals with a roformer (...) (ofc this only works if you possess both the original songs and instrumental version which you have to align perfectly)" - mesk

If you already [prepared your dataset](#), here is a step-by-step guide by Bas Curtiz on:

Setting up training on your local machine

(read also [mesk's guide](#))

"Make sure you have all dependencies installed for ZFTurbo's Training & Inference script:

<https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/docs/gui.md>

Update Nvidia drivers: <https://www.nvidia.com/en-us/drivers/>

Set power plan to best performance + never fall asleep.

Determine what the fastest drive is on/in your PC:

<https://www.guru3d.com/download/crystal-diskmark-download/>

Put your dataset on the fastest drive.

Read and apply the steps at:

<https://github.com/ZFTurbo/Music-Source-Separation-Training?tab=readme-ov-file#how-to-train>

Notes:

a) ZFTurbo's repo has a lot of config files to start with. Pick the one based on the model type, you want to use, inside the folder /configs

b) Alternatively, if you are going to fine-tune an existing model, use the .yaml associated

(unsure, implementation looks unstable or smth wonky)

To train locally, assuming you don't have a powerhouse of a GFX card, add this line in config:use_torch_checkpoint: True

a) Learning rate wise, If you are training a model from scratch, you want to set it higher: 5.00e-04 is used by ByteDance for example (=0.0005)

b) If you are fine-tuning an existing model, set it lower: 5.00e-06 is recommended (=0.000005)

Save the altered config and add your handle for ex.

For full overview of (optional) parameters that can be used:

<https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/train.py#L106>

- optional but recommended-

Create a free account at <https://wandb.ai> - shows u more insight on the training progress with graphs.

A free personal/cloud-hosted account should suffice.

Add parameter --wandb_key YOUR_API_KEY (which u can get from <https://wandb.ai/authorize>)

Save the full command in a text-file, handy for future usage. Hereby mine, which u can alter:

```
python train.py --model_type mel_band_roformer --config_path  
configs/config_musdb18_mel_band_roformer_bascurtiz.yaml --dataset_type 2 --results_path  
results --data_path datasets/train --valid_path datasets/validation --num_workers 4  
--device_ids 0 --wandb_key e304f2CENSOREDSOYOU NEEDTOUSEYOUR OWNdecc122e
```

Run the command in the root folder with CMD.

Check your progress/graphs at [https://wandb.ai/\[yourusername\]/projects](https://wandb.ai/[yourusername]/projects)

Latest update of repo gives u insight into fullness/bleedless too using parameter:
--metrics sdr bleedless fullness l1_freq si_sdr log_wmse aura_stft aura_mrstft"

"the augmentations help tho
even tho it's slower
gives it much more situations for real songs"
"ye if your dataset isn't from the biggest it will help"

"If you plan to stop/resume training many times, it could be interesting to also save optimizer (and scheduler) state with checkpoint, it can help train faster when you stop/resume a lot (as you resume everything in the state it was when you stopped training, instead of restarting optimizer from scratch each time)."

Patience parameter

"If set too low, it will reduce learning rate way too fast and lead to stagnant learning".
So, if "the model did not improve for like 10 epoches (weights did not save)" while patience is set to default 2, "you should set it to like 1000 to disable it for now."
"patience = X means that if during training, X consecutive epochs are not giving improvement (using sdr metric by default), it will reduce learning rate. If not set correctly, it can kill a training run by reducing too fast and too early learning rate."

When not sure, it's better to set it to really high value (like 1000 here) so it will be never triggered." - jarredou

Made from scratch training script by Dill "<https://github.com/dillfrescott/mvsep-beta>
"it uses a neural operator architecture with something I call Kernel Scale Attention to capture a range of details. I'm training it now. No guarantees tho on the quality but it's def working"

Troubleshooting of ZFTurbo's training code by Vinctekan

Issue: GPU isn't available, using CPU instead it will be very slow

"Turns out that the group and order of specific python packages that Turbo listed in the [.txt] is pretty cursed.

At the very end, pip just decides to remove your previous instances of torch, torchvision, and torhcaudio for some reason, and replaces it with the CPU versions, even if you decide to install pytorch CUDA beforehand. Tried removing torch==2.0.1 from the requirements but somehow it still stuck.

If you try to install pytorch CUDA AFTER installing the requirements, then it register as already installed. I thought about it for a while as to how could that be possible, but I slowly figured out that the CPU versions were installed because of it.

The way I found the fix is by pip uninstalling all 3 packages, and then reinstalling pytorch with the command on the website. It ultimately does not matter if it's 118 or 121."

Q: if I change something in the audio, model, training augmentations, inference section, or if I decide to remove augmentations entirely, will that still start training from where it left off, or is it going to start all over again?

A: "as long as you provide a starting checkpoint in the training code, it will continue where it left off" becruily

A: "Don't change "audio", "model" config, this must be the same as base checkpoint when resuming/fine-tuning, I think, but for "augmentations" part, you can edit as you want as it's pre-processing of the audio and done on the fly. mp3 encoding, pitchshifting and timestretching are quite resource heavy augmentations and can slow down training, other type of augmentations are more lightweighted.

For "inference", you can reduce overlap value if you want the validation step between each epoch to be a bit faster (overlap=1 will create clicks at chunk boundaries [fixed in newer MSST code])

"Training" part, you'll probably have to edit batch_size to find the max value your GPU can handle." jarredou

Q: Isn't changing chunk size in audio fine as long as it's divisible by the hop length?

I recently tried a lower chunk size while keeping everything the same (for melband) to help with VRAM issues, and it seemed to work (didn't train for long, just wanted to try)

A: I have never tried, but yeah, I think you're right, that's probably why we can also use different chunk_size/dim_t values for inference and the models are still working.

The lightest arch and still performing great seems to be **Vitlarge**.

"This arch is more tricky than other, even if lighter." jarredou

(?) segm_model in the script (or something like that)

musdb configs are for 4stem training, vocals ones are for 2stem

Q: What's the minimum length requirement

A: "Default segment_size in ht demucs config is 11 seconds audio chunks, so your training audio files should be longer or equal to 11 second length.

It can be lower, if there's no other choice."

- [Here](#), one user is being helped with training hi hat model from scratch using ZFTurbo code on an example of RTX 3060 12GB

- "I believe a length of about one minute per song is appropriate for the validation dataset."

Q: "My avg loss is always in the 130–120 range, is it worth the time to keep waiting for the training? The previous training is also like this, never touched 110 or under 100"

A: Don't worry about avg loss, look at the SDR on the metrics - is it improving?

Q: No improvement so far, the last one was at epoch 9, now I'm heading into epoch 13

A: "Yeah, don't worry, so what you're seeing is the loss curve.

You've been shooting down that ramp, but it slows improvements after a while"

[pic](#) (frazer)

How to get fast GPUs for training

By Bas Curtiz

"Budget" option - 4090, or

Buy A6000, preferably multiple.

Or hire them in the cloud.

Best bang for your buck for now

<https://vast.ai/>

[<https://www.tensordock.com/>] similar prices (although for November 2024, worse for 8xA100]

<https://www.runpod.io/>

<https://app.hyperbolic.xyz/compute> "5x and 8x H100 GPU instances with 1.8 TB storage for \$5 and \$8 per hour" - Kim

"hunder compute - A100XL (80GB) on \$1.05/hour - Essid]

"The easiest would be Colab, if you pay for the compute units the v100 is identical to training with 3090 locally, but Colab can get expensive quickly" - becruily

Paid Colab has now Nvidia A100 vs free Tesla T4. It's also faster than v100 and L4.

Most in-depth and handy article:

<https://timdettmers.com/2023/01/30/which-gpu-for-deep-learning/>

GPU performance chart:

https://i0.wp.com/timdettmers.com/wp-content/uploads/2023/01/GPUS_Advanced_performance3.png?w=1703&ssl=1

tldr; <https://nanx.me/gpu/>

[dtn: Performance in training

NVIDIA H100>A100 (40/80GB)>RTX 4090>RTX A6000 Ada>Nvidia L40 (also 18K CUDA cores)>prob. RTX 5000 Ada (12,8K CUDA cores)>RTX 4080 (9728K)>3090 Ti (10752)>V100 (32/16GB)>RTX 3090

https://i0.wp.com/timdettmers.com/wp-content/uploads/2023/01/GPUS_Ada_raw_performance3.png?w=1703&ssl=1

Cheaper GPUs for training

2x GTX 3090s used are cheaper than 4090 (but irc, the performance for multi-GPUs doesn't scale linearly, so might be not that affordable)

RTX 3090 24GB (CUDA cores: 10496)

4070 Ti Super 16GB (CUDA cores: 8448)

RTX 4070 Ti 12GB (CUDA cores: 7680, it's (still) tasking to train on it, and Roformers will achieve worse SDR due to necessity to start training with lower parameters)

2x GTX 1080 (if dual GPU scaling would be decent enough, it's not linear)

Not mentioning these:

RTX 2080 Ti (CUDA cores: 4352)

RTX 3060 12GB (CUDA cores: 4864)

GTX 1080 Ti (CUDA cores: 3584)

Sign up for an account at <https://sites.google.com/site/vultrfreecredit?pli=1>

Get 250 bucks free.

Add 50 bucks.

Now GPU rental is unlocked. Start there and vast.ai and wait for a server that has a6000 x8 for a good price.

But if you have enough time at hand, RTX 4090 is cheaper in the long run.

Depends on your electricity costs, though, which varies per country.

Training and inference performance for GPU per dollar

https://i0.wp.com/timdettmers.com/wp-content/uploads/2023/01/GPUs_Ada_performance_per_dollar6.png?ssl=1

Be aware that multi GPU configurations don't scale linearly.

We had an interesting discussion on the server on the choice between GTX 1080 Ti vs RTX 3060 12GB in training. We're yet to find out the final result, but unwa claims that despite having more CUDA cores, 1080 Ti might turn out to be slower. The possible reasons:

- Pascal's are "limited by FP16 performance by a factor of 64"
- No tensor cores ("Normally, AMP (Automatic Mixed Precision) is turned on when training a model, but AMP uses Tensor Core to speed up the computation." plus "Tensor Core generation is also newer [in 3000 series], with support for more precisions, including BF16.")
- "3060 is one generation newer than RTX 20xx / GTX 16xx and can use Flash Attention2. This is not very relevant for music source separation model, but may be very useful for LLM inference. (Roformer models have a Flash Attention entry in the configuration, but Memory Efficient Attention is used unless A100 GPU(s) are used.)"

Bas Curtiz is probably yet to find out the final verdict.

- "This is an extreme example, but it's a table comparing video generation times for each GPU under the settings of Wan2.2 Q6 K, 1280x704, 84 seconds, and 8 steps. The 5060Ti 16GB even outperforms the 3090" - Unwa

[pic](#)

Q: How long it takes to train a model?

A: "Depends on input and parameters and architecture.

MDX old version:

5k input (15k actually: inst/mixture/vocals) + 100 validation tracks (300, same deal), fullband, 300 epochs would have taken 3 months on a RTX 4090.

You can speed it up by going multiple GPUs and more memory, therefore:

A6000 (48gb) x 8 was like 14 days.

Damage on 300 epochs: ~700 bucks."

"7 days training of e.g. BS-Roformer on 8xA100-80GB": $7 \times 24 \times 15.12$ (runpod 8xa100 pricing) = \$2540.16"

4 days achieved epoch 74, and on epoch 162 for ~4200/4500 songs

Q: "4070 [8GB] works, but I would only use for testing IMO

A: I've trained some convtasnet in the past with really decent times [on 4070 8] (the new Ada Lovelace on 40 series makes faster tensor cores, which kinda compensates the less number of cores compared to 30 series)

A: [4070 8GB] is fine for non transformers.

If mamba blocks are used good it could be fine TBF.

The thing with transformers is that it is really reliant on VRAM.

A: Depends on what's inside the transformer, if it's flashatten then you need Ada.

Mamba has custom kernels, but I'm pretty sure 4090 can run it - what'll be cool is mamba + reversible net, super memory efficient in training, but it ends up being slower per step (around 2x compared to backprop).

I guess in reversible net you can have gigantic batch sizes which kinda circumvent the problem of a slow step speed"

There is a potential alternative to GPUs -

Training using TPU

<https://sites.research.google/trc/about/>

"(...) equivalent in performance to an a100

I'm not sure how good torch_xla support is now (...)" Cyclcrclicly

Turns out total usable VRAM for Pro is sadly 16GB, and 48GB of system memory

24 hours of interrupted training is possible, and 12 hours for free users.

It turned out to be extremely convoluted to fix compatibility issues.

Bcr: models need to be rewritten in JAX, I think; can't just train like this

Fr:

```
import torch_xla
model.to('xla')
for inputs, labels in train_loader:
    with torch_xla.step():
        inputs, labels = inputs.to('xla'), labels.to('xla')
        model(blah blah)
```

```
#after above epoch is finished
```

```
+ torch_xla.sync()
```

```
+
```

Anv: there's jax version <https://github.com/flyingblackshark/jax-bs-roformer>

Fr: You dont need JAX - all you need to do is import torchxla and move it to xla device wherever the model is instantiated, I think train.py probs, add that import at the top, where the model is moved to device - change that from .to('cuda') to .to('xla'), then in the train loop you add the with torch_xla step, move inputs to xla then after that train loop you put xla sync

Bas stuck during fixing it before, but here's the convo where he stuck:

Why does training the bs_roformer model with 8s chunksize, 256 dim, 8 depth consume only 13GB of VRAM now, compared to 21GB last time [they could decrease VRAM since then]

Stuck troubleshooting of TPU training by Bas Curtiz (Q) and frazer, DJ NUO, jarredou and Cyclcrclcy (A):

Q: is it as simple as adding pytorch lightning though?

A: try using "xla" as the device instead of cuda and if you're lucky everything will Just Work™

Q: The Pytorch's implementation of stft does not work on the XLA (TPU), because it internally uses some unsupported functions.

There are not feasible workarounds for it available.

Only some 3x PhD discussion, which discusses the underlying function not working, which would require forking pytorch to get it working, IF the solution was actually even feasible:

(hacky super slow workaround, or just "use different shit").

Only "realistic" solution I've found is porting the mel band roformer to tensorflow.

Which is bruh, but the thing is in their docs STFT says:

Implemented with TPU/GPU-compatible ops and supports gradients..

Also tensorflow is by google, the TPU as well, so yk, it might have better support.

The same error basically is described here:

<https://github.com/pytorch/xla/issues/2241>

A: As frazer said, you'll have better luck with jax than tensorflow

A: Can you try putting data to CPU & running it there, and then put the result back on TPU?

I encounter similar issues when running on Mac MPS (GPU), and this code helps to alleviate the issue:

```
stft_repr = torch.stft(raw_audio.cpu() if x_is_mps else raw_audio, **self.stft_kw_args,
window=stft_window.cpu() if x_is_mps else stft_window, return_complex=True).to(device)
```

(of course, in your case the code might be a bit different, but it demonstrates the idea)

Q: obviously slow

it is called in a loop in the forward function (= very slow)

...if it was like only once / before each step, but not inside step.

we'll try anyways, thanks

Timed-out after 10 mins, 0 steps were finished.

Imagine doing 4032 steps.

JAX is like an optimizer/JIT.

STFT of it, is just Scipy's STFT but running under JAX.

Scipy's implementation is CPU-based.

So it expects CPU data. Not Tensor/GPU/TPU data.

A: Or this might help (custom implementation of STFT):

<https://github.com/MasayaKawamura/MB-iSTFT-VITS/blob/main/stft.py>

A: There's also <https://github.com/qiuqiangkong/torchlibrosa/> that has a stft implementation

Q: Hmm both use numpy which is cpu based

A: yeah its some weird operation in the torch spec, i use

<https://github.com/adobe-research/convvmelspec> anytime incompatibility occurs

Q: May be we need to replace mel spec with this in MelRoformer.

I got a boilerplate/minimal production ready, but 2 things...

no TPU for me right now to test - maybe someone else has better luck / paid Colab sub.

Last outcome, which might be fixed by now: RuntimeError: Attempted to call

variable.set_data(tensor), but variable and tensor have incompatible tensor type.

A: you can use kaggle for tpuv3 with probably better availability

Q: <https://github.com/qiuqiangkong/torchlibrosa/> result:

Calling "torch.nn.functional.fold" just gets stuck, when interrupting, the error stack has mentions of copying to CPU.

...smth to do with the fold function.

Numpy only in initialization (cpu), so that's fine.

<https://github.com/MasayaKawamura/MB-iSTFT-VITS/blob/main/stft.py> result:

Numpy and basically cpu all-the-way, so no/go.

<https://github.com/adobe-research/convvmelspec> result:

Not a STFT library / whole spectrogram, don't wanna dissect it, the STFT part seems internal,

didn't notice (would have to double-check) the inverse, but wasted 2 days already. done with it.

A: Just a guess (have no experience with Tensorflow): what if STFT portion of the code can be executed by TensorFlow code -> convert result to numpy CPU -> convert to PyTorch tensor

Q: Problem is... It simply takes too much time, copying to cpu and back is expensive resource-wise

A: In some part of torchlibrosa they use a workaround for nn.functional.fold function, maybe that can be reproduced/adapted to the other failing part where fold used.

A: line 239 is the drop in, you have to make sure the settings are the same from what i remember <https://github.com/adobe-research/convvmelspec/blob/main/convvmelspec/stft.py>

Q: It got thru the whole ass forward step. But now it's stuck at backward step.

yk, recalculate the weights based on this step to improve the model.

replaced the backwards function of stft with empty one, and yet: stuck.

so since backwards step of stft/istft is disabled...
the problem is elsewhere.

No idea where, no idea how to debug, out of my expertise.
A: I might be 100% wrong here, but I think you should disable the backward pass through
that class if it is type nn.Module
stft.requires_grad=False
or when you call stft use a decorator with indentation
with torch.no_grad():
x=stft(x)

Other archs

SCNet: Sparse Compression Network

Large models by ZFTurbo turned out to sound between Roformers and MDX.
“SCNet is maybe a bit more bloody than MDX23c” and/or possibly noisy, judging by the
MVSEP model(s). “seemingly impossible” to train guitars

[July 10th 2024] “Official SCNet repo has been updated by the author with training code.

<https://github.com/starrytong/SCNet>

“ZF’s script already can train SCNet, but currently it doesn’t give good results”

<https://github.com/ZFTurbo/Music-Source-Separation-Training/releases/>

The author’s checkpoint:

<https://drive.google.com/file/d/1CdEllqsoRfHn1SJ7rccPfyYioW3BIXcW/view>

June 2025

ZFTurbo: “I added new version of SCNet with mask in main MSST repository. Available with
key ‘scnet_masked’. Thanks to becruiy for help.”

“the main thing is removing the SCNet buzzing” - Dry Paint

How heavily undertrained weights looks on spectrograms with mask vs without: [click](#)

“One diff I see between author config and ZF’s one, is that dev has used learning rate of
5e-04 while it’s 4e-05 in ZF config. And main issue ZF was facing was slow progress (while
author said it worked as expected using ZF training script

<https://github.com/starrytong/SCNet/issues/1#issuecomment-2063025663>)”

The author:

“All our experiments are conducted on 8 Nvidia V100 GPUs.

When training solely on the MUSDB18-HQ dataset, the model is
trained for 130 epochs with the Adam [22] optimizer with an initial
learning rate of 5e-4 and batch size of 4 for each GPU. Nevertheless,
we adjust the learning rate to 3e-4 when introducing additional data
to mitigate potential gradient explosion.”

“Q: So that mean that you have to modulate the learning rate depending on the size of the dataset ?

I think it's first time I read something in that way

A: Yea, I suppose because the dataset is larger you need to ensure the model sees the whole distribution instead of just learning the first couple of batches”

Paper: <https://arxiv.org/abs/2401.13276>

<https://cdn.discordapp.com/attachments/708579735583588366/1200415850277130250/image.png> (dead)

On the same dataset (MUSDB18-HQ), it performs a lot better than Demucs 4 (Demucs HT). “melband is still sota cause if you increase the feature dimensions and blocks it gets better you can't scale up scnet cause it isn't a transformer it's a good cheap alt version tho”

ZFTurbo “I trained small model because author post weights for small. Now I'm training large version of model, but it's slow and still not reach quality of small version.

I use the same dataset for both models

My SCNet large stuck at SDR 9.1 for vocals. I don't know why

My small SCNet has SDR 10.2

I added config of SCNet to train on MUSDB18:

https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/configs/config_musdb18_scnet_large.yaml

Only changes comparing to small model are these parts:

Small:

dims:

- 4
- 32
- 64
- 128

band_SR:

- 0.175
- 0.392
- 0.433

Large:

dims:

- 4
- 64
- 128
- 256

band_SR:

- 0.225

- 0.372
- 0.403"

ZFTurbo eventually trained SCNet large model, but it turned out to sound similar to Roformers, but with more noise. You can test the model on MVSEP.com

SCNet Large turned out to be good for piano (vs MDX23C and MelRoformer) and also drums models according to ZFTurbo.

"He also said SCNet didn't work that well for strings, Aufr didn't have luck with BV model as well"

"MDX23c is already looking better on guitar after 5 epochs than scnet after 100 epochs"

"with SCNet I've had the fastest results with prodigy [optimizer]" becruily

Later, ZFTurbo released SCNet 4 stems (in his repo) and exclusive bass model on MVSEP.

There was also an older, an unofficial (not fully finished yet, it seems) implementation of SCNet: <https://github.com/amateur/SCNet-PyTorch>

Experimental **BS-Mamba**

git clone <https://github.com/mapperize/Music-Source-Separation-Training.git> --branch workingmamba

TS-BSmamba2

<https://arxiv.org/abs/2409.06245>

<https://github.com/baijinglin/TS-BSmamba2>

Added to ZFTurbo training repo:

<https://github.com/ZFTurbo/Music-Source-Separation-Training/>

At this moment, training works only on Linux or WSL.

SDR seems to be higher than all the current archs, maybe besides Mel/BS Roformers (weren't tested). "It's in between SCNet and Rofos but maybe more lightweight than them. (...) From the scores from MelBand paper [it seems] the Rofos are still like +0.5 SDR average above the other archs when trained on musdb18 only."

But it's great to finally see some mamba-inspired MSS arch with great performance". As for 22.09.24, ZFTurbo had problems with low SDR during training.

<https://discord.com/channels/708579735583588363/1220364005034561628/1286650425596186645>

<https://discord.com/channels/708579735583588363/1220364005034561628/1284221988294099102>

Another three very promising archs for the moment:

Conformer

“performs just as well if not better than a standard Roformer”

<https://arxiv.org/pdf/2005.08100.pdf>

<https://github.com/lucidrains/conformer>

(people already train with it, and its implementation might be pushed to the MSST repo in not distant future)

<https://github.com/ZFTurbo/Music-Source-Separation-Training/issues/169>

Essid pretrain:

<https://huggingface.co/Essid/MelBandConformer/tree/main>

https://mvsep.com/quality_checker/entry/9087

“Due to cost issues, I'm discontinuing the Mel-Band-Conformer MUSDB18HQ-based train. I'm sharing the ckpt and config, so anyone who wants to continue can use them.”

It has [shown](#) steady improvement in training in the last 12 hours from epoch 0 to 83 (1 SDR increase on private validation dataset, probably on A100XL (80GB) on “thunder compute” \$1.05/hour) and the shared weight is epoch 200+.

TF-Locoformer

<https://arxiv.org/abs/2408.03440>

https://github.com/merlresearch/tf-locoformer/blob/main/espnet2/enh/separatortflocoformer_separator.py

“I see only now that the tf-locoformer repo was updated to include the variants published few months ago (TF-Locoformer-NoPE and BS-Locoformer)

<https://github.com/merlresearch/tf-locoformer>” - jarredou

dTNet

<https://github.com/junyuchen-cjy/DTTNet-Pytorch>

“They report very good performance on vocals with low parameters” - Kim

Back in the end of 2023, one indie pop song from multisong dataset (of the two there) received the best SDR - Bas Curtiz

“better than scnet imo, remains to see if it can beat rofos”

“not fast to train. I'm back with vanilla mdx23c

Trying a config to train model with less than 4GB VRAM, almost at 7 SDR for vocals in 8 hours of training (on moisesdb+musdb18, and using musdb18 eval, with my 1080Ti and batch_size=1, chunk_size is around 1.5sec”

Modification of the training code for MSST by becruily ([DL](#); [old](#)).

Breaks compatibility with the authors' checkpoint.

“Also keep in mind authors trained with l1 loss only, default in msst is masked loss”

“from what I read, l1 loss when dataset is noisy, mse loss when dataset is clean”

“the loss is defined from msst, but in the original dttnet it was in the code itself

you can just --loss l1_loss”

@jarredou “I copied your tfc and tfc_tdf classes to my files (and used that latest stft/istft I sent) - and seems to be better, just like the og dttnet
the tfc/tdf fixed the nan issue for me” - becruily

Installation instruction:

“In the latest MSST [at least for 13.10.25]

add the ddtnet folder to "models" and replace your settings file in utils with this”

“The weird thing is, it sounds like a fullness model despite not being one, I barely can find dips in instrumentals

ddtnet vs kim melband, if anyone is curious

<https://drive.google.com/drive/folders/12an8wnKC-FKE48gVu9pHvUaLSxzpC6C8?usp=sharing>

Keep in mind ddtnet was trained only with musdb and has 10-20x less params while being comparable in quality”

“the authors checkpoints had 16khz cutoff because dim_f was smaller than nfft/2

if you want to train model with cutoff it's fine, if you want fullband then dim_f must be half of nfft + 1”

“I've found the issue in my DTTNet version leading to the "noisy" outputs. It was just the * changed to a + in forward [here](#)” - jarredou

Everything uploaded at the top.

Mesk's config for training instrumental model (achieved from SDR 6 to 9.3 in a third epoch [counting the first as 0]):

```
python train.py --model_type dttnet --config_path config-ddtnet-other.yaml
--start_check_point results/vocalsg32_ep4082.ckpt --results_path results/ --data_path
[YOUR DATASET] --valid_path [YOUR VALIDATION DATASET] --dataset_type [TYPE
1/2/3/4/5] --num_workers 8 --device_ids 0 --metric_for_scheduler sdr --metrics fullness
bleedless l1_freq
change these accordingly:
>data_path
>valid_path
>dataset_type
```

On the side.

ZLUDA is a translation layer for CUDA allowing to use any CUDA-written app to be used with AMD (and formerly Intel) GPUs, and without any modifications to such app. Weaker GPUs than 7900 XT might show its weaknesses considerably, [compared](#) to better GPUs. The example came from ZLUDA in Blender, but rather from AMD period code, so before the takedown and rollback to pre-AMD codebase so now ZLUDA is more crippled. With never released code, at certain point it was even made to support Batman Arkham Knight, with general plans to support DLSS, but it will probably never see a day light. Maybe [this](#) repo still has the old base forked - version 3 codebase is still being updated there. Utilizing it on 6700 XT in [stable-diffusion-webui-amdgpu](#), it was performing slowly like DirectML, but on 7900 XT it sped up the process from 3-4 to ~1 minute. The first execution can be slow due to need of creating cache. Then it can surpass ROCm performance-wise if you manage to make it work. Plus, ZLUDA works on Windows and supports older AMD GPUs, like even RX 500 series (use [lshqytiiger's repo](#), check e.g. ROCm 5 version if your app doesn't start, but it might crash anyway), while for ROCm on Linux and older GPUs, e.g. RX 5700 XT should work with some quirks (e.g. HIP 5.7 and ROCm around 5.2.* - [src](#), although you can try out 6.21 or 6.2.x to ensure, as it could happen that some earlier 6.x wasn't supporting RX 5700 XT correctly, while e.g. for RX 6000 ROCm 6.24 should be used at the moment).

It could be interesting to see utilizing training repo using ZLUDA e.g. on Windows instead of ROCm Pytorch on Linux but Unwa notice in the ZLUDA repo fork, "PyTorch: torch.stft does not always return correct result." and it might be problematic during training, so ZLUDA might be not a good solution for training currently, but who knows whether for inferencing on e.g. Windows using MSST or UVR, although the latter crashes for me during separation with nvcuda.dll. But I haven't tried messing with HIP SDK mentioned in the release page or other fork's ZLUDA versions than the newest. I don't even have anything in C:\Program Files\AMD\ROCM (if it wasn't even futile without it), but I have amdhip64.dll v. 5.5 in system32 (if 5.7 isn't shipped with newer drivers and required).

Also, didn't follow these instructions yet, and they might be useful and contain some older GPUs workaround:

<https://github.com/vladmandic/sdnext/wiki/ZLUDA>

All gfx names with corresponding GPU models:

<https://llvm.org/docs/AMDGPUUsage.html#processors>

More ZLUDA research and workarounds (may work for UVR, not have too):

<https://github.com/comfyanonymous/ComfyUI#amd-gpus-experimental-windows-and-linux-rdna-3-35-and-4-only>

(RDNA 3-4 instructions for Python manual installation using DirectML branch of UVR)

https://github.com/comfyanonymous/ComfyUI#for-amd-cards-not-officially-supported-by-roc_m

(flags for RDNA 2-3)

<https://github.com/patientx/ComfyUI-Zluda>

(Instructions for GCN4-RDNA4;

RDNA 2 with HIP 6.2.4 and experimental 6.4.2)

Using hip 5.7.1 and corresponding ZLUDA should be possible on RDNA2 too

<https://github.com/CS1o/Stable-Diffusion-Info/wiki/Webui-Installation-Guides#amd-fooocus-with-zluda>

(Step 5 in "Setting up Zluda" a bit below - for GPUs below RX 6800 or 9070/60, and instructions above the point are there for 6800 or higher too

<https://github.com/CS1o/Stable-Diffusion-Info/wiki/Webui-Installation-Guides#rocm-hip-sdk-5-7-with-zluda-setup>

(Instructions for GCN4 [RX 400/500]; it contains a step with
pip install torch==2.2.1 torchvision==0.17.1 torchaudio==2.2.1 --index-url
<https://download.pytorch.org/whl/cu118>)

<https://github.com/advanced-lvl-up/Rx470-Vega10-Rx580-gfx803-gfx900-fix-AMD-GPU#important-notes-on-installation>

(Instructions for GCN4 [GFX803 & GFX900])

<https://github.com/ROCM/ROCM/issues/4749#issuecomment-3117083336>

(some old 7900 XTX (gfx11100) troubleshooting

- HIP SDK 6.5 might a bit less crashy,

"Follow the instructions at <https://github.com/patientx/ComfyUI-Zluda> then do the patches at <https://github.com/patientx/ComfyUI-Zluda/issues/222>

Possible complement for RX 6600/6700 (if wasn't mentioned in the instructions above already)

<https://github.com/YellowRoseCx/koboldcpp-rocm/releases/download/deps-v6.2.0/rocblas-6.2.0.dll.7z>

extract that zip file into C:/Program Files/AMD/ROCM/6.1/bin/ it should merge with the "rocblas" folder that's already in there.

KAN-Stem

That might be interesting to train multistem, it's based on Demucs:

<https://github.com/waefrebeorn/KAN-Stem>

VR architecture by tsurumeso <https://github.com/tsurumeso/vocal-remover>

(VR models in UVR, use modified v5 training code in order to support e.g. 4 bands, inferencing v6 models is not yet supported in UVR)

The arch is obsolete for instrumentals - bleeding and vocal artefacts.

Not really recommended anymore, unless for specific tasks like de-noise, de-reverb or Karaoke or BVE when MDX V1 wasn't giving that good results.

(guide by Joe)

Q: How do I train my own models?

A:

Model Training Tutorial

Requirements:

- Windows 10
- Nvidia GeForce Graphic card (at least 8 GB of VRAM)
- At least 16GB of Ram
- Recommend 1 - 2TB of hard drive

Setup your dataset

1. You need to know...

Attention:

- Although you can train your model with mp3, m4a, flac file, but we recommend convert those file to wav file.
- For high-resolution audio sources, the samples are reduced to 44.1kHz during conversion.
- If possible, match the playback position and volume of the OnVocal and OffVocal sound sources.
- The dataset required at least 150 pairs of songs

2. Rename the file...

Attention:

Create "mixtures" folder with vocals / "instruments" folder without vocals

Please separate the sound sources with and without vocals as shown below.

There is also a rule for file names, please make the file names numbers and add "_mix" / "_inst" at the end.

Example:

Instrumental with vocal:

D:\dataset\mixtures\001_mix.wav
D:\dataset\mixtures\002_mix.wav
D:\dataset\mixtures\003_mix.wav

.

.

Instrumental only:

```
D:\dataset\instruments\001_inst.wav  
D:\dataset\instruments\002_inst.wav  
D:\dataset\instruments\003_inst.wav...  
.  
.
```

3. Download the vocal-Remover from GitHub

Link: <https://github.com/tsurumeso/vocal-remover/releases/>

4. Install the program (Use this command down below)...

```
pip install --no-cache-dir -r requirements.txt
```

5. Start learning

```
python train.py --dataset D:\dataset\ --reduction_rate 0.5 --mixup_rate 0.5 --gpu 0
```

Attention:

If you want to pause, press Ctrl+Shift+C

6. Continue learning

Example:

```
python train.py --dataset D:\dataset\ --pretrained_model .\models\model_iter(number).pth  
--reduction_rate 0.5 --mixup_rate 0.5 --gpu 0
```

Compared to VR5 arch, VR6 now can handle phase. Although I'm not sure if it implements Aufr33 multiband functionality which models trained for UVR5 on VR5 utilize (I'm not sure if that training code is in the old CML UVR5 code).

MedleyVox

Excellent for training duet/unison and separately main/rest vocals.

The original code is extremely messy and broken at the same time, and dataset is big and hard to obtain. Cyrus was to publish their own repository with fixed code and complete dataset at some point.

The problem of the model trained by Cyrus was training cutoff used while training.

"The ISR_net is basically just a different type of model that attempts to make audio super resolution and then separate it. I only trained it cuz that's what the paper's author did, but it gives worse results than just the normal fine-tuned" ~Cyrus

Apart from training code, there wasn't any model released by the authors. Only result snippets.

<https://github.com/JusperLee/TDANet>

"I think this arch should worth a try with multiple singer separation, as it's performing quite well on speaker separation, and it seems it can be trained with a custom number of voices (same usual samplerate & mono limitations tho)" jr

MossFormer2 may perform better

"These archs are not implement in ZF's script but are really promising for multiple speakers separation, and should be working for multiple singers separation if trained on singing voice:

<https://github.com/dmlguq456/SepReformer> (current SOTA)

<https://github.com/JusperLee/TDANet>

<https://github.com/alibabasglab/MossFormer2>

" jarreodu

Bas Curtiz

"Few takeaways I learned from the issues at its GitHub,

<https://github.com/dmlguq456/SepReformer/issues>

currently only supports 8khz sample rate (so downsample your 44.1khz samples to this prior)

samples only: max 10-20 seconds input, otherwise potential memory issues (so chunk a full song into such segments prior)

Individual samples are not supported (so it's folder-based, put your samples in there)"

Also, there are various errors which some users tend to encounter, at least on Windows machines.

New sep algo

<https://github.com/OliverRensu/xAR>

[This repository includes the official implementation of our paper "Beyond Next-Token: Next-X Prediction for Autoregressive Visual Generation"]

Might be potentially useful for any training in Colab (by HV, 2021):

```
function ConnectButton(){
  console.log("Connect pushed");
  document.querySelector("#top-toolbar >
colab-connect-button").shadowRoot.querySelector("#connect").click()
}
setInterval(ConnectButton,60000);
<- enter this on console (not cell)
```

and keep Colab on foreground.

It's not really good to train in Colab at all, due to its limitations.
If you're training because you want a better model than v5/v4 mgm models, stop it, you won't surpass mgm models with just Colab. However, you could subscribe to <https://cloud.google.com/gcp> and watch some YouTube tutorials how to utilise its resources to Colab."

(archived)

Sidechain stem limiting guide by Vinctekan

Hello all, I am here to share the definitive answer to exporting sets of stems with consistent and loudness and brickwall like mixing, when a manual mixture of pairs/stems are too loud or are modified.

Even though, pairs like this probably won't have to be used for training in the future, it's still going to be super important for evaluation for said models, or techniques that any of you may discover in the future.

I discovered this through this video, the details and specifics behind are explained in this if you would like to recreate it manual

<https://www.youtube.com/watch?v=Hv8nENoNvbk&t>

This is basically a Side Chain Stem Limiting method that uses FabFilter's Pro-L 2 limiter plugin in the REAPER DAW to mix your stems in a way that when you mix them together in Audacity with the "Mix and Render" option, you get a perfect waveblock like mix, with no clipping and no distortion.

Decided to help you all out and created two REAPER templates where this mixing method is used, so you don't have to make it manually. I'll give out a 4 stem template and a 2 stem template for vocals, and instrumental that you all can use to recreate the above.

The steps to make the above happen aren't exactly the same as in the video, in addition there are a lot of things you don't need to do (since I have already done it), so here is a step-by-step guide:

Requirements:

1. REAPER (DAW) [in the video, it says you can use any DAW]
2. FabFilter Pro-L 2 limiter plugin (preferably the regular VST version, instead of VST3)

Steps:

1. Open the REAPER Project File of your choice (if you're exporting 2 stems, use the 2 stem version, if you're exporting 4 stems, use the 4 stem template)
2. Drag your stems into the corresponding channels, you also have to drag it into the channels labeled: "DUPE"
 - Your vocal stem to "VOCALS", and "VOCALS DUPE"
 - Your drum stem to "DRUMS", and "DRUMS DUPE"
 - Your other stem to "OTHER", and "OTHER DUPE"
 - Your bass stem to "BASS", and "BASS DUPE"
 - Your instrumental to "INSTRUMENTAL", and "INSTRUMENTAL DUPE" [For the 2 stem template]
3. Check the settings of the limiter to make sure it suits your needs.
 - You can set the gain on the left side of the UI, if you think your mix it still isn't loud enough.
 - I used 8x oversampling as default, if you feel like your CPU can handle more or less, you can adjust it to suit your needs.
 - If the exported stems have distortion (by any chance), you can set the limiting mode to SAFE, which prioritizes transients, and keeps unwanted sounds to ABSOLUTE ZERO.
 - You can also think about adjusting the attack, release, and channel linking settings if it's not good enough, but I think the settings in the templates are good for any form of limiting.
 - Make sure "True Peak Limiting" is always on, if it isn't, distortion might become a factor again in the final results
4. Now it's time to export the stems in the first track folder individually. You can do this by soloing them with the yellow "S" button next to the tracks.

4.5 In REAPER: File>Render... and render. Rinse and repeat for all of the stems.
 These are the settings I recommend using, if you plan to further edit the results, and also for retaining the quality of the sources:

- No Tail
- 44100hz or 48000hz sample rate
- Channels: Stereo
- Resample mode: r8brain free (highest quality, fast)
- Format: WAV
- WAV bit depth: 24 bit PCM
- Nothing else

Done!

+5. You can check your work by opening Audacity, importing the exported stems, and mixing them together by pressing CTRL+A, going inside: Tracks > Mix > Mix and Render.
 If everything is done correctly, you should have a mix of stems which sound nice to the ears, and has absolutely zero clipping. You can see if it clips or not by checking: View > Show Clipping (on/off). Or you can press CTRL+A, go inside Effects > Volume and Compression > Amplify. If it's correct, the Amplification bar should show 0.0 DB.

Clipping bug workaround

https://cdn.discordapp.com/attachments/708579735583588366/1139206772092051496/Instrumental_Fix.mp4

In addition to Safe Mode, I set the release to the max, and it worked that way, but the dynamic were shite.

More:

https://discord.com/channels/708579735583588363/708579735583588366/1139189181873_143869

In conclusion to below: The instrumental clipping wasn't the Fabfilter Pro-L 2 VST's fault, or any sidechain limiter for that matter. This is just how digital audio works, unfortunately. (And PS. - 32-bit float exporting might prevent clipping).

Trivia

Ugggh, just checked out both Pedalboard, and DawDreamer, from what it looks like: It's not really possible to recreate stem limiting with a RAM loaded mixture as a reference/auxiliary input.

The only 2 remaining possibilities that I am thinking of is using pydub, librosa, scipy or pyo to do it without the use of a DAW.

If that's not possible, then the only option left is to control REAPER with reapy + reascript.

I also think I now understand why the peak amplitude of the instrumental is decreased when you re-mix the acapella back in to the mix:

Since music is basically just about, 22000 different sine waves going off at the exact same time with changing amplitudes, the pressure waves of all of these sine waves interfere with each other constantly:

If at any given time the pressure waves of these sine waves have a perfectly aligned value of +1, then they add up together, creating a strong signal

On the flip side: there are times when they cancel each other out, because the amplitude are different (e.g 1st being +1 and the 2nd being -1)

I watched a video guide in Fourier Transform, and the concept is visually demonstrated really well:

<https://www.youtube.com/watch?v=spUNpyF58BY&t=50s>

In a nutshell: If you take away the vocals, certain frequencies of the instrumental get amplified, because now the vocal isn't there to dampen it/cancel it out.

You can recreate this by taking a brickwall limited recording of your choice, lowering the DBFS by at least -2. Then you can process it through an MDX model, and then compare the peak amplitudes of:

1. the mixture
2. the Instrumental
3. and the separated instrumental and acapella mixed back together

<https://github.com/jeonchangbin49/musdb-XL/>

From what I can understand, they applied a maximizer to all the mixtures, then calculated the differences of amplitude in a sample by samples by basis, and applied the difference to all the stems at once.

I think I could do that.

Update

"Even though I have found out that using Pro-C 2 [Sidechain compressor, not a limiter] totally fixes the issue of mixes clipping after turning down just about any stem, the trade-off is that the LUFS [short term] suffers by at least -2 DB"

(older techniques from before the guide above)

jarredou's guide:

Here's 2 "proof-of-concept" python dynamic range compressor/limiter I've made recently and that are working with sidechain and multiple stems inputs:

1st one "pydub_comp_fork.py" is a fork of pydub's dynamic range compressor
(line79 to change the audio inputs)

You can set attack/release/ratio/threshold settings like any other compressor

2nd one "limiter.py" is a fork of this Safety Limiter: <https://github.com/nhthn/safety-limiter/>
(54line to change the audio inputs)

You have "release" and "hold_time" settings.
(no threshold here, you just gain the input)

Even if they were sounding "ok" with normal settings, the speed performances were not satisfying for any of them for the planned use, I will not develop them more, consider them as abandonware. But they can maybe be usefull for someone else.

from pydub import AudioSegment, effects

<https://cdn.discordapp.com/attachments/773763762887852072/1167555636272316467/limiter.py>

https://cdn.discordapp.com/attachments/773763762887852072/1167555635928367265/pydub_comp_fork.py

You can use this technique to make the loudness of your stems consistent:

<https://github.com/jeonchangbin49/musdb-XL> to get better results with your model, where usually there's a problem with proper isolation of overly compressed music.

You can also read Aufr33 short guide

on his approach toward this problem (plus more explanations [here](#))

For a problem of inconsistent volume in mixture vs stems when a limiter is used, sidechain mixture to a limiter.

Other option, more close to real world processing :

- * apply (strong) compressor/limiter to individual stems to mimic the mixing process
- * and then apply (softer/lighter) compressor/limiter on mixture (with sidechain trick) to mimic the mastering process.

Because if you apply too much limiting on the mixture, it will destroy the sound. 2-stage dynamics processing is more transparent.

The only problem with the technique is that there could be clipping if we invert one or more stems over the mixture.

Unless the AIs work with [32 bit] floating point (not integer!)

Exemplary step-by-step guide

I used side-chain on the 2 stems with source 3 as input.

Somehow I had to set the threshold to -12db (instead of the OG -24db) i applied to the mixture (prolly coz 12 x 2 stems)

Used the same Ratio/Attack/Release settings as used with compressor prior, this time on the side-chain compressor.

Two templates for Reaper. One with better LSP Sidechain Limiter Stereo which is Linux Studio Plugin and the other with free reacomp. Target is around -7.5 ilufs, but anything between 8.5 and 9 will do fine.

https://cdn.discordapp.com/attachments/708595418400817162/1108853386608136252/Pair_Limiter.RPP

https://cdn.discordapp.com/attachments/708595418400817162/1108861182506455170/Pair_Limiter_-_ReaComp.RPP

These may not be the final files. ReaComp struggled more at some point. Consider using e.g. also iZotope RX9/10 Maximiser IRC IV for more transparent results.
E.g. Aufr33 used Voxengo and sometimes ReaXComp in 4 channel mode.

Alternatively, you can experiment with:

KSHMR Chain method by Sam Hocking

"I too get some residual that doesn't null when comparing Master Bus v Distributed Stem Mastering."

"The way gainmatch works is it exists on your before processing chain and after processing chain and real-time communicates the difference between the two (does the part knock is showing), so the adjustment is made dynamically as a gain match calc, or you can use it as a target match too. While the loudness adjusting could all be an offline one click process, you would still have to set it all up manually in a DAW. There are some cool duplication chain-style solutions in ProTools that could achieve it more easily, however. My personal favourite is a tool called KSHMR Chain which will work in any Stereo DAW and that allows one plugin instance to be effective on hundreds of tracks at the same time but controlled from one master plugin. This way you could actually adjust every single audio to a common master LUFS dynamically and click export stems and all would be dynamically adjusted at once and offline exported."

<https://www.excite-audio.com/kshmr-chain>

Short guide of Aufr33 approach

<https://cdn.discordapp.com/attachments/900904142669754399/1090876675966894142/sm.png>

"If anyone is wondering how I create pairs. Here's what my project looks like in REAPER.

Before the master bus is the limiter plugin, which works with 4-channels. After rendering a pair for one dataset (in this case, for Karokee), I swap audio items and render the pairs for other datasets: BVE, Strings, etc."

"For training, just make sure that all pairs have a margin of about 0.3 dB. Storing pairs larger than 16 bits can be useful for further editing." aufr33

Local SDR testing script

https://drive.google.com/file/d/1GC9pwch0WQZXwBNTz_QnXE_UyxdKmQF/view?usp=sharing by Dill

<https://drive.google.com/file/d/1BeqNw3TnRTDMwnoQGMbOgwrcGRwe4Zht/view?usp=sharing> GUI by zmis (but it scores a bit lower for some reason):

“Here's a handy little python script I made using the help of Ai that can calculate the SDR of a track based off of the actual instrumental or vocal of the song.

You can do `python sdr.py --help` for an explanation on how to use the script.

You just need numpy and scipy for it to work, and python ofc!

I'm not sure if you would like to pin this or not, but I've been using this script to help me improve my separation methods.

<https://github.com/ZFTurbo/Audio-separation-models-checker/tree/main>

Based on MUSDB18-HQ dataset”

“Q: Why SDR goes <0 in silence parts? (song_006)

A: SDR and SISDR behave weirdly when 1 of the input is silent, and that's why log_WMSE was made: <https://github.com/crlandsc/torch-log-wmse/>

Interestingly, L1freqMag metrics is giving same results than some users here (1296 a bit better for instrumentals, 1297 a bit better for vocals).” jarredou

Best ensemble finder for a song script

by Vinctekan

<https://drive.google.com/file/d/1LUtBsCSym1iDHqADEusmACs-LF2INYLw/view?usp=sharing>

Currently, this optimized version can find the best combo of 9, 3 minute audio files in about 2 minutes and 40 seconds in Colab.

Refactored best *weighted* ensemble finder by jarredou

<https://drive.google.com/file/d/1Rm09z1wpj0Pi-6bFQ15u767n1XV95pDz/view?usp=sharing>

“That's what I've used to find optimal weights for my MDX23 fork v2.5 update.
It's still Nelder-Mead based optimizer, but code is way more simple/clean than 1st version.

To use it, you need:

A dataset of clean sources (with exact same filename scheme than mvsep multisong dataset).

Process dataset mixtures with all the models you want to ensemble and put the outputs in different folders, 1 for each model (and still with exact same filename scheme than mvsep multisong dataset).

librosa and scipy python libs

then run (for example):

```
weight_finder_v2.py \
--ref c:\reference_dataset \
--est c:\InstVoc c:\bsrofo1296 c:\kimrofo \
--stem vocals \
--extension flac \
--tracks 100
```

--ref is clean sources' folder path

--est is estimates (separations) folder paths (multiple inputs)

--stem is stem name (based on multisong dataset filename scheme)

--extension is audio file extension (flac/wav...)

--tracks is a number of tracks in a dataset.

It will process the datasets many times and change weights each time until it find the best balance. When finished, it will output weights scaled to 10 max value.

Warning: it can take hours (or even days, depending on the number of models to ensemble, size of dataset and resources of computer)

A python lib to align audio:

<https://github.com/nomonosound/fast-align-audio>"

Universal function to make different types of ensembles by ZFTurbo

<https://cdn.discordapp.com/attachments/911050124661227542/1192220574982881320/ensemble.py>

I think it's the same or newer:

<https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/ensemble.py>

“In my experiments SDR for avg_wave always the max.”

Now also jarredou made his Colab with the above implemented with comfy GUI:

https://colab.research.google.com/github/jarredou/Music-Source-Separation-Training-Colab-Inference/blob/main/Manual_Ensemble_Colab.ipynb

Volume compensation for MDX v2 models

How to automate calculation of volume compensation value for all older MDX models
(results are not perfect and need to be fine-tuned)

by jarredou

So, I have maybe a protocol to find accurate volume compensation:

- Use a short .wav file of just noise (I've used pink noise here) and pass it through the model you wanna evaluate

- Take the resulting audio, the one that will have all the noise in it and compare it to the original noise with this little python script that will give you the difference in dBTP and the quivalent VC ratio (you'll need to

pip install librosa

if you don't have it installed already). The results I've found with it are coherent with the ones you've found by ears ! (1.035437 for HQ2 / 1.022099 for KimFT other)

Here's the script :

```
import numpy as np
import argparse
import librosa
```

```
def Diff_dBTP(file1,file2):
```

```
    y1, sr1 = librosa.load(file1)
    y2, sr2 = librosa.load(file2)
```

```
    true_peak1 = np.max(np.abs(y1))
    true_peak2 = np.max(np.abs(y2))
```

```
    difference = 20 * np.log10(true_peak1 / true_peak2)
```

```
    print(f"Diff_dBTP : The difference in true peak between the two audio files is
{difference:.6f} dB.")
```

```
    ratio = 10 ** (difference / 20)
```

```
    print(f"The volume of sound2 is {ratio:.6f} times that of sound1.\n")
```

```

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Find volume difference of two audio
files.")
    parser.add_argument("file1", help="Path to original audio file")
    parser.add_argument("file2", help="Path to extracted audio file")
    args = parser.parse_args()

Diff_dBTP(args.file1, args.file2)

```

**Volume compensation values for various models (in reality they may differ +/- e.g. by
0.00xxxx, but maybe not much more)**

All values according to the script made by **jarredou**

(All default but Spectral Inversion - Off; Denoise Output: On; - the latter shouldn't affect the results if turned off):

- Kim Vocal_1 - 1.012819
- Kim Vocal 2 - 1.009
- voc_ft - 1.021
- Kim ft other - 1.020 (Bas' fine-tuned and SDR-validated)
- UVR-MDX-NET 1 - 1.017194
- UVR-MDX-NET Inst 2 - 1.037748
- UVR-MDX-NET Inst 3 - 1.043115
- UVR-MDX-NET Inst HQ 1 - 1.052259
- UVR-MDX-NET Inst HQ 2 - 1.047476
- UVR-MDX-NET Inst Main - 1.037812 (actually it turned out to be 1.025)
- UVR-MDX-NET Main - 1.002124
- UVR-MDX-NET-Inst_full_292 - 1.056003
- UVR-MDX-NET_Inst_82_beta - 1.088610
- UVR-MDX-NET_Inst_90_beta - 1.151219 (wtf)
- UVR-MDX-NET_Main_340 - 1.002742
- UVR-MDX-NET_Main_406 - 1.001850
- UVR-MDX-NET_Main_427 - 1.002091
- UVR-MDX-NET_Main_438 - 1.001799
- UVR_MDXNET_9482 - 1.007059

"denoise is just processing twice with the second try inverted, after separation reinverted, to amplify the result, but remove the noise introduced by MDX, and then deamplified by 6db, so it still the same volume, just without MDX noise.

Basically HV noise removal trick"

UVR-MDX parameters & hashes decoded by Bas Curtiz

https://github.com/Anjok07/ultimatevocalremovergui/blob/master/models/MDX_Net_Models/model_data/model_data.json - the link with hashes possess MDX models parameters.
The above probably still doesn't possess all the models added in the update, e.g. Foxy model, but there are only 4-5 combinations of settings so far.

File with newer models parameters:

https://raw.githubusercontent.com/TRVlvr/application_data/main/mdx_model_data/model_data_new.json

All MDX-Net model parameters in UVR consist of these combinations:

- HQ_4:

self.n_fft = 6144 dim_f = 2560 dim_t = 8

- All older HQ fullbands:

self.n_fft = 6144 dim_f = 3072 dim_t = 8

- kim vocal 1/2, kim ft other (inst), inst 1-3 (415-464), 427, voc_ft:

self.n_fft = 7680 dim_f = 3072 dim_t = 8

- 496, Karaoke, 9.X (NET-X)

self.n_fft = 6144 dim_f = 2048 dim_t = 8 (and 9 kuielab_a_vocals only)

- Karaoke 2

self.n_fft = 5120 dim_f = 2048 dim_t = 8

- De-reverb by FoxJoy

self.n_fft = 7680 dim_f = 3072 dim_t = 9

UVR model hash decode

"I've made this little script a while back to find those hashes.

Use with model_hash_finder.py path_to_model_file."

<https://drive.google.com/file/d/1D4TNKjuObNn6MSiss1PtXPQoR3XJOwJ/view?usp=sharing>

It's a checksum hash but based only on the last 10MB of model files." (jarredou)

```
full_band_inst_model_new_epoch_309.onnx fea6de84f625c6413d0ee920dd3ec32f
full_band_inst_model_new_epoch_337.onnx 4bc04e98b6cf5efeb581a0f382b60499
kim_ft_other.onnx b6bccda408a436db8500083ef3491e8b
```

Kim_Vocal_1.onnx 73492b58195c3b52d34590d5474452f6
Kim_vocal_2.onnx 970b3f9492014d18fefeedfe4773cb42
UVR-MDX-NET-Voc_FT.onnx 77d07b2667ddf05b9e3175941b4454a0
kuielab_a_bass.onnx 6703e39f36f18aa7855ee1047765621d
kuielab_a_drums.onnx dc41ede5961d50f277eb846db17f5319
kuielab_a_other.onnx 26d308f91f3423a67dc69a6d12a8793d
kuielab_a_vocals.onnx 5f6483271e1efb9bfb59e4a3e6d4d098
kuielab_b_bass.onnx c3b29bdce8c4fa17ec609e16220330ab
kuielab_b_drums.onnx 4910e7827f335048bdac11fa967772f9
kuielab_b_other.onnx 65ab5919372a128e4167f5e01a8fda85
kuielab_b_vocals.onnx 6b31de20e84392859a3d09d43f089515
Reverb_HQ_By_FoxJoy.onnx cd5b2989ad863f116c855db1dfe24e39
UVR-MDX-NET-Inst_1.onnx 2cdd429caac38f0194b133884160f2c6
UVR-MDX-NET-Inst_2.onnx ceed671467c1f64ebdfac8a2490d0d52
UVR-MDX-NET-Inst_3.onnx e5572e58abf111f80d8241d2e44e7fa4
UVR-MDX-NET-Inst_full_292.onnx b06327a00d5e5fbc7d96e1781bbdb596
UVR-MDX-NET-Inst_full_338.onnx 13819d85cad1c9d659343ba09ccf77a8
UVR-MDX-NET-Inst_full_382.onnx 734b716c193493a49f8f1ad548451c48
UVR-MDX-NET-Inst_full_386.onnx 2e4fc9ec905f35d2b8216933b5009ff
UVR-MDX-NET-Inst_full_403.onnx 94ff780b977d3ca07c7a343dab2e25dd
UVR-MDX-NET-Inst_HQ_1.onnx 291c2049608edb52648b96e27eb80e95
UVR-MDX-NET-Inst_HQ_2.onnx cc63408db3d80b4d85b0287d1d7c9632
UVR-MDX-NET-Inst_HQ_2.onnx 55657dd70583b0fedfba5f67df11d711
UVR-MDX-NET-Inst_Main.onnx 1c56ec0224f1d559c42fd6fd2a67b154
UVR-MDX-NET_Inst_187_beta.onnx d2a1376f310e4f7fa37fb9b5774eb701
UVR-MDX-NET_Inst_82_beta.onnx f2df6d6863d8f435436d8b561594ff49
UVR-MDX-NET_Inst_90_beta.onnx 488b3e6f8bd3717d9d7c428476be2d75
UVR-MDX-NET_Main_340.onnx 867595e9de46f6ab699008295df62798
UVR-MDX-NET_Main_390.onnx 398580b6d5d973af3120df54cee6759d
UVR-MDX-NET_Main_406.onnx 5d343409ef0df48c7d78cce9f0106781
UVR-MDX-NET_Main_427.onnx b33d9b3950b6cbf5fe90a32608924700
UVR-MDX-NET_Main_438.onnx e7324c873b1f615c35c1967f912db92a
UVR_MDXNET_1_9703.onnx a3cd63058945e777505c01d2507daf37
UVR_MDXNET_2_9682.onnx d94058f8c7f1fae4164868ae8ae66b20
UVR_MDXNET_3_9662.onnx d7bff498db9324db933d913388cba6be
UVR_MDXNET_9482.onnx 0ddfc0eb5792638ad5dc27850236c246
UVR_MDXNET_KARA.onnx 2f5501189a2f6db6349916fabe8c90de
UVR_MDXNET_KARA_2.onnx 1d64a6d2c30f709b8c9b4ce1366d96ee
UVR_MDXNET_Main.onnx 53c4baf4d12c3e6c3831bb8f5b532b93

VR de-reverb models decode

UVR-De-Echo-Normal.pth = f200a145434efc7dcf0cd093f517ed52
UVR-De-Echo-Aggressive.pth = 6857b2972e1754913aad0c9a1678c753

UVR-DeEcho-DeReverb.pth = 0fb9249ffe4ffc38d7b16243f394c0ff
So they're all "4band_v3.json" config file (from [here](#))

More thorough chart by David Duchamp a.k.a. Captain FLAM:

https://docs.google.com/spreadsheets/d/1XZAYKmgJkKE3fVKrJm9pBGIXlcSQC3GWYYI90b_uI1M

Voice Cloning

“RVC and some of its forks ([Apilio](#), Mangio, etc) are genuine free, open source ones for inference and training. For realtime voice changer that uses RVC models, there's w-okada: <https://rentry.co/VoiceChangerGuide>” no guide for Linux though.

<https://www.tryreplay.io/>

“Url downloads, local files, massive database of models, both huggingface and weightsgg, in built separation models, options to skip that part if you have vocals, ability to use multiple ai models for one particular result, and the option to either merge or just get multiple results at the end, plus whatever else, de-reverb and stuff” it has voc_ft vocal model from UVR5.

“even my old laptop still can inferencing using apilio
i3 3217u 1.8ghz
intel hd 4000”

And you're probably aware already that RVC Colabs to train voice cloned models are banned.

Stable Audio Open Gen

Available on MVSEP in the Experimental section. It's not for separation, but generating sounds.

ZFTurbo: “Algorithm based on model:

<https://huggingface.co/stabilityai/stable-audio-open-1.0>

Audio is generated in Stereo format with a sample rate of 44.1 kHz and duration up to 47 seconds. The quality is quite high. It's better to make prompts in English.

Example prompts:

1) Sound effects generation: cats meow, lion roar, dog bark

- 2) Sample generation: 128 BPM tech house drum loop
- 3) Specific instrument generation: A Coltrane-style jazz solo: fast, chaotic passages (200 BPM), with piercing saxophone screams and sharp dynamic changes

Examples:

Cat meow: <https://mvsep.com/result/20250612092110-b297c082fb-generated.wav>

Dog bark: <https://mvsep.com/result/20250612115517-b297c082fb-generated.wav>

128 BPM tech house drum loop:

<https://mvsep.com/result/20250612115841-b297c082fb-generated.wav>

Violin solo: <https://mvsep.com/result/20250612120111-b297c082fb-generated.wav>

Woman sing song "Happy Birthday to you":

<https://mvsep.com/result/20250612120433-b297c082fb-generated.wav>"

Links for research on separation

<https://github.com/facebookresearch/AudioMAE>

<https://arxiv.org/abs/2310.02802>

<https://github.com/pbelcak/fastfeedforward>

<https://github.com/corl-team/rebased>

<https://github.com/bowang-lab/U-Mamba/tree/main>

<https://www.unite.ai/mamba-redefining-sequence-modeling-and-outperforming-transformers-architecture/>

<https://github.com/state-spaces/mamba>

https://github.com/apapiu/mamba_small_bench

("this one is actually exciting because it runs faster and leaner than transformers and promises to surpass them in quality

>What makes Mamba truly unique is its departure from traditional attention and MLP blocks. This simplification leads to a lighter, faster model that scales linearly with the sequence length – a feat unmatched by its predecessors. Mamba has demonstrated superior performance in various domains, including language, audio, and genomics...")

"mamba is real fucking complicated. like reaaaally complicated (...) hyper params do seem hard to adjust tho."

"mamba is kinda sick but its early days in the SSM space, so lots of the tricks that you can do with transformers you can't do with SSMs because they haven't become mainstream

but mamba has two very cool properties

it has positional information by its nature - i.e. no extra computation is required to embed positional info

linear time complexity - so in audio it's super useful because audio data hits the $O(n^2)$ complexity (if the chunksize is large enough)"

"i personally don't trust any of the mamba papers - they either say how mamba is the best thing since sliced bread or worse than 3 year old transformers although the paper I read for that was questionable"

<https://www.harvard.edu/kempner-institute/2024/02/02/repeat-after-me-transformers-are-better-than-state-space-models-at-copying-2/>"

"they don't even replace the mask estimator thing in bs-roformer with mamba"

<https://arxiv.org/abs/2404.02063>

<https://arxiv.org/abs/2401.09417>

<https://github.com/hustvl/Vim>

<https://github.com/RobinBruegger/RevTorch>

<https://huggingface.co/blog/rwkv>

Why does music source separation benefit from cacophony?

<https://arxiv.org/abs/2402.18407>

It makes our side chain stem limiting thing irrelevant.

"As the paper demonstrate that using only randomly mixed stems is more efficient for training than using only real paired stems (from the same song and sync), in that random mix config, the individual stems will never be against the mixture that was used to limit them, so making that process irrelevant" jarredou

MDX23C training code by ZFTurbo has the mix randomization feature built-in - dataset type 1 is random mix, dataset type 4 is the real mix (aligned).

"I think now after reading that paper that once you have a dataset large enough and using the random mixing with some simple augmentations like gain changes/channel swap/phase inversion/EQ/soft-clipping(tanh), you're good to go and can forget more resource intensive augmentations like pitch shifting and time stretching, that can really slow down training. Maybe just reverbs can be still usefull even if it need more resources than simple math processing.

So really go fast/minimal on pre-processing in fact." -||-

"The only good paper about SDR I have in mind is "SDR - half-baked or well done?"

<https://arxiv.org/abs/1811.02508>

from 2018, but maybe there are some more recent ones on the subject

There's also that thesis that is interesting but maybe also outdated now (as based on the old OpenUnmix), about loss functions effect on source separation learning:

<https://discord.com/channels/708579735583588363/911050124661227542/1191134740284190750>

My go-to URLs to follow publications:

<https://arxiv.org/list/cs.SD/pastweek?show=2000>

(weekly list)

<https://arxiv.org/list/eess.AS/pastweek?show=200>

(weekly list)

I've registered to <https://www.scholar-inbox.com>

recently (it's free), which can be handy (but lots of duplicate if you follow already arxiv publications above)

and also: <https://twitter.com/csteinmetz1/>

for sure"

Griffin: Mixing Gated Linear Recurrences with Local Attention for E...

<https://arxiv.org/abs/2402.19427>

new tweak to a modern transformer architecture improves performance

<https://github.com/IAHispano/gdown>

If you have some issues with downloading files from GDrive on Colab

Q: Can you recommend something to automate adding effects (and if possible randomized)

A:

<https://pytorch.org/audio/stable/generated/torchaudio.io.AudioEffect.html#torchaudio.io.AudioEffect>

maybe even <http://ccrma.stanford.edu/planetccrma/man/man1/sox.1.html>

<https://github.com/iver56/audiomentations>

(which uses random parameters by design)

<https://github.com/spotify/pedalboard>

(take a look at the augmentations in ZFTurbo script (dataset.py), it uses both libs with randomized parameters also for pedalboard)

Q: What Transformer and Mamba is

A: <https://www.youtube.com/watch?v=XfpMkf4rD6E>

<https://www.youtube.com/watch?v=9dSkvxS2EB0>

Side-note: with a bit of tweaking, ZFTurbo training script can be edited to train a reverb model, generating the randomized reverb on the fly with pedalboard.Reverb

(<https://spotify.github.io/pedalboard/reference/pedalboard.html#pedalboard.Reverb>) and using reverbs IRs to have more diversity

https://openaccess.thecvf.com/content/CVPR2022/papers/Mangalam_Reversible_Vision_Transformers_CVPR_2022_paper.pdf

<https://arxiv.org/abs/2306.09342>

Fork of ZFTurbo training code, but I don't know with what changes (by frazer):

<https://github.com/fmac2000/Music-Source-Separation-Training-Models/tree/revnet>

Another (by joowon)

<https://github.com/mapperize/Music-Source-Separation-Training>

Another (not so new) paper with maybe interesting concept improving separations quality that could be reproduced:

VocEmb4SVS: Improving Singing Voice Separation with Vocal Embeddings

<http://www.apsipa.org/proceedings/2022/APSIPA%202022/TuAM1-7/1570836845.pdf>

There's also a demo site for the 4-stem version, but I haven't found any publication/code

<https://cathy0610.github.io/2023-SrcEmb4MSS/>

"Demucs employs a combination of L1 loss and deep clustering loss to optimize source separation." (<https://github.com/facebookresearch/demucs/issues/458>) I've found this paper few months ago, its findings are based only on openunmix arch, the observed behaviour could be different with other archs, but it's still very interesting:

<https://arxiv.org/abs/2202.07968>

Not really in MDX23 code made by ZFTurbo:

"By default, my code uses loss proposed by kueilab team. They use MSE but skip sample with worst loss (to avoid problems in dataset). mse loss can be used directly with --mse_loss argument.

Also, auraloss is included in my code too. I experimented with it, but it didn't allow to gain additional profit comparing to standard loss function."

Useful lib to experiment with different loss functions:

<https://github.com/csteinmetz1/auraloss>

I've seen that paper in my feed last month, doing real-time source separation (23ms latency):

<https://arxiv.org/abs/2402.17701>

Mamba: Linear-Time Sequence Modeling with Selective State Spaces

<https://www.youtube.com/watch?v=9dSkvxS2EB0>

Vocal restoration research

<https://github.com/facebookresearch/AudioMAE>

<https://carlosholivan.github.io/demos/audio-restoration-2023.html>

<https://google.github.io/df-conformer/miipher/>

<https://arxiv.org/abs/2403.05393>

<https://github.com/vikastokala/bcctn>

<https://github.com/espnet/espnet>

https://github.com/manosplitsis/hifi-gan-bwe/tree/train_with_music

"new vocoder replacing hifi-gan, vocos, bigvgan etc
compared to other ones, high freq smearing practically doesn't occur"
EVA-GAN - another breakthrough over HiFi-GAN

<https://arxiv.org/abs/2402.00892>

<https://arxiv.org/pdf/2402.00892.pdf>

What can potentially help on training on inferior GPUS with large model size is LORA.

'especially since bsrofromer is transformer

then you can allow users to finetune even the largest

loras work by going through a model and replacing all the linear projections with a wider projection (then i think projecting back to the original size)

so imagine you got a linear projection thats trained, ie its 4 neurons in 4 neurons out - lora works by adding X neurons either side of the projection (i cant really remember but its somethign like this)

then you freeze all the other stuff around the linears and only train the new linears (the lora)
if you open up the lora source code you'll see what i mean, theres a loop that just iterates all the weights and replaces the linears with a loralinear (and thats the entire method)

lora will allow for better SDR on whatever you trained (albeit a small sample set)

so itd be super smart to treat it like how image gen treats it, so u can say make a lora for crowd removal mixed with the lora for vocals

then ppl in this disc can try create the best lora for their specific usecase and mix match with other's" frazer

More: <https://radekoslowski.com/how-to-fine-tune-a-transformer-pt-2/>

"I think here it can be useful if we will have very great multistem model and after finetune it on rare instruments." ZFTurbo

<https://github.com/Human9000/nd-Mamba2-torch>

Visit our [#dev-talk](#) channel for more

[Anjok's interview](#) on YT

TL;DW: UVR's documentary + training, archs and demudder explained

Anjok is the developer of Ultimate Vocal Remover 5 (UVR5 GUI).

He intended UVR to be a Swiss army tool - to contain everything you need for separation, and also contain models made by the community (e.g. dereverb/denoise/deecho).

History of UVR

Anjok in times where Spleeter was still a thing, found a VR arch made by Japanese developer, tsurumeso, and received better results than Spleeter. He started to make his own model on laptop 1060 6GB on 100 or 150 pairs with the absolute minimum parameters, and it turned out to be a better model than tsurumeso's one. Later he transitioned to faster GPU (probably before 3090 yet).

Anjok wanted GUI for VR, and found BoskanDilan on Fiver and simply contracted him, paying to build the foundations of what UVR is today. BoskanDilan turned out to be a very good and talented coder.

They put the work on GitHub, and Aufr33 contacted Anjok with ideas on how the VR models can be improved etc.

Then BoskanDilan left in mid 2021 for personal reasons. Then the GUI work was taken by Anjok who was mentored by BoskanDilan to improve on understanding the coding. Anjok started to working on UVR exclusively, spending 10 hours a day for UVR in 2022.

He decided to make a simple installer in one package, as he received lots of issues on GitHub, from people not knowing how to install it. He also re-coded the UVR to make the code easier to maintain. Then Bas Curtiz helped Anjok on design aspects of UVR, e.g. designed new logo, and gave some advice, and good amount of feedback from UVR user perspective. Early 2022 phase of UVR development took a lot of advice from early users of UVR.

In May 2022 there was a first installer released to make UVR more accessible without e.g. installing Python or other dependencies and specialized programming knowledge to set up a proper environment.

Anjok was still in charge of introducing other archs than VR into UVR, being simply the only one behind the process, while normally bigger teams work on projects of that scale, when e.g. different archs could be coded into UVR by different developers. It was a stressful period of time, because Anjok intended to make the software which is free of bugs, and still not fully rely on the community in terms of bug reporting.

Then the Mac version came out and M1/M2/M3 support for faster GPU acceleration. Anjok found out in Demucs repo a part of the code, making it easier to port UVR to Macs, and it is used by every model. Music community is pretty Mac-centered, and he devoted a considerable amount of time to make it work reliably on Macs too.

In the new UVR version there's a planned demudder to be introduced (described later), and possibly translations.

Anjok currently trains a new model coming in several weeks.

It's intended to be a little smaller in order to be not so resource intensive, but also better than the best current MDX-Net model.

Update 01.03.24

"I'm going to allow HQ4 to continue training beyond 1500+ epochs as an experiment (it's currently at 1200), and interestingly, the SDR has been steadily increasing. It has significantly surpassed HQ3 in terms of SDR and listening tests, and it also outperformed MDXC23 in listening tests, though not in SDR (yet!). The most recent evaluation on the multi-dataset showed a score of 15.85, using the default settings. Clearly, there's a limit to how much further training can enhance performance, but up to this point, improvements are still being observed. This model has been in training since October! I'm chipping away at the next GUI update as well, and the demudder will be in it."

The model was released, with already HQ_5 scheduled in following month/s.

The archs in UVR and their technicalities summarized

VR

VR uses audio spectrograms and converts them to FFT spectrograms.

VR uses only magnitude spectrograms, not phase.

Phase represents timing where the data is, while magnitude represents the intensity of each frequency.

Phase is much harder to predict.

Actually VR uses original phase from the mixture and saves it during the process "and it just does the magnitude".

That's the reason why VR tends to have more artefacts in it. The smearing in instrumentals of VR is because the phase from the mixture is still in there.

Aufr33 later introduced 4 bands support for UVR.

Let's say for first of three bands between 0-700Hz there will be different resolution, for all other frequency ranges there will be different. E.g. knowing that vocals are in specific frequency range, you can optimize it further.

That feature made UVR and VR arch much better.

Later they introduced -

Ensembling

So a way to use multiple models to potentially get better results.

The three ways of ensembling:

avg - gets the average of vocals/instrumentals

max - is maximum result of each stem, e.g. in a vocal you'll get the heaviest weighted vocal from each model, and the same goes for instrumental, giving a bit cleaner results, but more artefacts

min

MDX-Net

Uses full spectrogram with phase and magnitude

Tradeoff is muddier results, but natural, cleaner sound.

Training

Anjok separated on nearly every genre you can think of, and stated that the hardest genre for separation is metal and vocal-centered mixes. Also, if the instrumental has lot of noise, e.g. distorted guitars, the instrumental will come out muddier.

MDX-Net was the arch, addressing lots of VR issues in its core.

Tracks from 70-80s can separate well. 50-60s will be harder, e.g. recorded in mono. Early stereo era gets a little better.

A good model needs to be as good as the dataset for a model.

There was lots of work scrapping it from the internet.

Aufr33 was the mastermind behind Karaoke model and its dataset.

Demucs model wasn't as successful, as probably was more meant for more stems, and MDX-Net gave better results for 2 stems.

Training details covered in this interview can be found at the top of [Traning models guide](#) section of the doc

The biggest issue in terms of archs and the source of muddiness, is phase. Currently, in audio separation there's not a great way to calculate phase in a model like the phase spectrogram as it's not as obvious as the magnitude spectrogram.

You take the vocal out of a heavy rock track, but the process is not perfect, so it will take some part of the instrumental with it. Even if you don't hear instrumental in vocals, there's still instrumental data in there in the phase of that vocal track.

In the end of the day, source separation is prediction. It's predicting where it thinks it is, but there will be always some imperfections, e.g. whenever you hear muddier sound in a track which has more noise like metal tracks.

Anjok emphasizes on (currently) lack of correlation between SDR and the fact that bigger SDR metric doesn't necessarily mean better. He tried some top of the SDR chart result before, and wasn't quite happy about them.

Because phase is a big part of the issue, now the new upcoming -

Demudder

A UVR feature incoming (it was also explained before on the server by Anjok - if something is not clear, try to find his messages there)

It uses lots of phasing tricks. It processes the track twice. The first takes instrumental from the first go around and compares it against the original mixture. It chops the mixture into 3 seconds chunks and ?inerts over that lists of chunks and for each segment, it cuts out where

that segment is in the instrumental, and it finds similar events that aren't at the exact same place. It takes those chunks, and it analyses them against the instrumental that was generated, and it tries to find the most similar events it can from the instrumental, that aren't at the same place from that segment, and it finds similar events, and then it phases it, it does a phase invert of that instrumental

(56:30) If the volume or DB threshold isn't past the certain point because it's too loud then it means it does not cancel out and doesn't make phase invert, if it reaches a certain threshold like if it is below certain threshold it'll phase that, and then it will basically stitch together a new mixture that is kind of phased from that original instrumental output, and it reprocesses that new stitch together, mixture with the phase with the instrumental phase changed, and it processes that through the second pass, and then it takes that vocal and then phase inverts it with the mixture, with the original mixture and then what you end up having is some of the parts that are similar from the other parts of the track, you end up having those fill in the spectral holes.

Sam remarks find some similarity with probably how Izotope Imager works.

Anjok says: I'm trying to get a similar part, but also try to take it and phase it with that segment. Because it's not the exact same part of the segment, it's not gonna be a perfect phase, because it would be an original vocal output.

So it's kind of still finding the bit of instrumental that is still in the vocal.

Sam remarks about frequent situations where you perform separation, and it can lead to decrease of e.g. hihat volume levels in instrumental, referring to what information separated vocal stem can wear. It's part of the muddiness Anjok tried to address with the feature.

Anjok didn't want to compromise vocal quality, and in some cases it makes vocal better too, but it also depends on how the track was mixed originally. If it's analog track recorded in one session or even live track, it won't work so good. The problem is with e.g. 10 minutes track, when demudder won't find phase similarities so effectively. It will work the best on music made with samples. If the track is digital, it is more likely to work better.

Anjok currently works on it to make it work for all tracks.

The more he works on it, the more breakthroughs are made, but due to his day job, he had less time to work on it lately.

Anjok gives his appreciation on the group of very talented developers who made MDX-Net arch in the University of Korea. It's his favourite network. He's a big fan of Woosung Choi work.

Later, Aufr33 invented his own:

Simpler demudder

Published for paid users of x-minus.pro (when you pick Roformer model for instrumentals, buttons with methods appear; it is only applied for instrumentals, not vocals)

In his own words:

"

1. Separate the song into vocals and music
2. Invert the phase of the vocal and mix it with the music
3. Now separate this mix
4. Mix the vocals with the input song

It actually works more complicated than that. I added a high pass filter since the demudder is not needed at low frequencies."

Probably something from the 100-250 Hz range.

Actually Aufr33 used following ffmpeg command:

" -filter_complex
"[0:a]highpass=f=900[hp1];[0:a][hp1]amerge,pan=stereo|c0=c0-c2|c1=c1-c3[lp];[1:a]highpass
=f=900[hp2];[lp][hp2]amix=inputs=2:duration=longest:normalize=0[out]"

Rephrased by becruily

"use roformer on a song
phase invert the vocal file and combine it with the instrumental
separate again using the same model
combine the original song and vocals (no inversion or anything) and you will get demuddled
inst

this is for instrumental, if you want demuddled vocals just switch the two words (acapella and instrumental)"

[Video](#) how to apply demudder method

Notes

- For HQ 4 and at least denoise model enabled, the method seems to produce more vocal residues, so it might be feasible more for Roformers (it's used optionally for Kim Mel-Roformer on x-minus).
- "xminus demudder is more pleasing to the ears" isling

- Some people might still prefer max_mag ensemble on x-minus or mel-roformer + bs-roformer ensemble in UVR

Phase fixer on x-minus for unwa inst v1 model copies phase from Kim Mel-Roformer model.

UVR Demudders released in [beta Roformer patch](#)
(in Anjok's words)

- **Phase Rotate:**

The fullest sounding, but can leave a lot of artifacts with certain models. I only recommend that method for the muddiest models. Otherwise, Combined Methods is the best”

- First, a filtered instrumental is created, and the left and right channels are swapped.
- The phase is shifted by 90 degrees.
- This modified filtered instrumental is then inverted with the original mixture, and another inference pass is performed on the resulting mixture.
- Finally, the vocal from the second pass is phase-inverted and combined with the original mixture, creating a cleaner instrumental.

- **Phase Remix** (similar to X-Minus):

I don't recommend using phase remix on the Instrumental v1e model. I recommend combined methods or phase rotate for models producing fuller instrumentals.

- The mixture is first separated into stems.
- The phase of the vocal stem is inverted and mixed with the filtered instrumental to produce a modified "mixture." Another inference pass is performed on this new mixture.
- The vocal stem extracted from the modified mixture is then reintroduced into the original mixture, creating a cleaner instrumental.
- This method is *only* recommended for models that produce very muddy instrumentals!

- **Combine Methods:**

- It's basically a weighted mix of the final instrumentals generated by "Phase Rotate", "Phase Remix," and the initial instrumental.

Demudder in UVR doesn't work on 4GB VRAM Intel/AMD GPUs.

Phase fixer/swapper
(decreases vocal residues in instrumentals)

The method was invented by Aufr33 to fix noise in Roformers models trained with instrumental stem target. It copies phase from a model trained with vocal stem target which usually gives muddy instrumentals (e.g. Kim's or becruily's vocal) and copies it into the instrumental model. Initially it was added only to x-minus, and later becruily wrote his own script doing the same (torch and librosa implementations).

Later Anjok implemented it into one of the UVR Roformer beta patches (Tools>Phase Swapper), although there it only allows changing high and low cutoff, but no high frequency weight, and santilli_ found out that increasing it from 0.8 to 2 is beneficial for phase swapping from becruily vocal to instrumental model and that it's much better than manipulating with high and low cutoff, you can use their forked Colab [here](#).

You can use and edit original phase swap Python scripts for previously separated files [here](#). The result is - less noise in the instrumental, but a bit more muddiness (usage is described in the link).

Optionally, in Phase Fixer you could set 420 for low and 4200 for high or 500 for both and Mel-Kim model for source; and bleed suppressor (by unwa/97chris) to alleviate the noise further (e.g. phase fixer on its own works better with v1 model to alleviate the residues). Besides the default UVR default 500/5000 and Colab default 500/9000 values, you could potentially “even try like 200/1000 or even below for 2nd value.” “I would say that the more noisy the input is, the lower you have to set the frequency for the phase fixer.” - jarredou

“The optimal FFT size I found is 896 or maybe 1024. The default that UVR and MSST uses is 2048. At least for ensembling different versions of a song in different codecs (like from YT or SoundCloud). I haven't actually tried this with stem outputs, but I probably will tomorrow.”

Phase-fixed output used in UVR’s Manual ensemble (method explained by objectbed)

On example of dca’s "0) Unwa BS Roformer Resurrection Inst (BS 2025.07 as a reference for phase fix) + MVSEP BS Roformer 2025.07 (Max Spec)

—> the least vocal crossbleeding.

Alternatively, you can use becruily vocal model instead of 2025.07 for the ensemble - “Becruily vocal correctly recognize instruments far better than the instrumental one” - dca100fb8”

“This would equate to the following steps:

1. Separate your mixture using the Unwa BS Roformer Resurrection Inst model.
 - Output: inst_unwa.wav (instrumental) + optional vocal.
2. Separate your mixture using the MVSEP BS Roformer 2025.07 model.
 - Output: inst_mvsep.wav + vox_mvsep.wav.
3. Phase fix with UVR’s Phase Swapper:

- 3a. Target Audio = inst_unwa.wav (from Step 1).
- 3b. Reference Audio = vox_mvsep.wav (from Step 2).
- 3c. Click Start Processing.
- 3d. Note the resulting new file (inst_unwa_phaseswapped.wav or similar).

4. Build the ensemble with UVR's Manual Ensemble mode:

- 4a. Inputs =
 - inst_unwa_phaseswapped.wav (from Step 3d)
 - inst_mvsep.wav (from Step 2)
- 4b. Set Algorithm = "Max Spec."
- 4c. Click Start Processing.

The resulting file = your final instrumental stem (the one referenced in the Google Doc instructions as having the least cross-bleed)."

Q: Question about MVSEP BS Roformer 2025.07. Is this a model I can download, or do I have to do it online on their site? I can't find it.

A: It's mvsep site only. No download. So you want to use this model, must go mvsep.com

Q: What exactly are we doing with a Phase Fixer/Swapper? As an audio engineer, I understand phase as how it relates to frequency over time. When a vocal stem is used as a "reference" for the target instrumental stem, what's actually happening?

A: "There is phase in waveform domain, like audio engineers experience it every day, and there is phase in STFT domain. In STFT domain phase means how each STFT bins content is organized with other bins.

The math concept is similar to phase in waveform domain, but instead of having 1 phase value for 1 waveform, you have 1 value for each STFT bin and for each STFT frame, so it's way more complex... (to make it really short).

The phase fixer/swapper thing is operating in the STFT domain, so for each STFT bins, it will use the magnitude data of 1 model, and it will use the phase data from another model, and hopefully this can improve the final result". - jarredou

"In short, what the script does is blend the stft phase of the "donor" file into the target, it uses different blending scales for different frequencies so that it'll only affect the parts that are directly related to the perceived noise" - santilli_ / Michael

Q: Apparently the original Resurrection model only has two models (BS-Roformer 1296/1297 by viperx and BS Large V1 by unwa) that work for phase fixer, but the Gabox one says to use 2025.7 for phase fixer (which works almost perfectly) How does that happen? How did the working models change for Gabox's fine tune?

A: ML models are black boxes. You never know what you'll get. Like in Forrest Gump. It depends on specific instrumental model, which vocal model as source for phase fixer will work the best.

Pitch shifting algorithms' comparison

<https://www.youtube.com/watch?v=gaSFt0tT2u4>
https://www.youtube.com/watch?v=s-5g4I30_eY
<https://www.youtube.com/watch?v=WH8KDQALYQY>

“I've had much better results with Izotope RX than Studio One for example for stretching.”

Also, Bitwig can be good.

You can also try out paid Lossless Pitch AI on dango.ai (tuanziai.com/en-US).

Research:

<https://discord.com/channels/708579735583588363/911050124661227542/1303058610934382675>

Restoring hi-end in pitched-down tracks - [click](#)

What does changing batch_size from 1 to 2

(it wasn't used in Rofo beta UVR for 9 Jan 2025, but maybe it got changed)

“if your input is batch time sequence, it looks like this:

[batch1->[time->[sequence],time2->[sequence]..],
batch2->[time->[sequence],time2->[sequence]..]]

As you increase batch_size you increase the amount of data the model gets to churn through.

So higher batch_size allows the model to see more data before you do a thing called backward prop which calculates another thing called gradients, which are used to improve the model by tuning loads of little values inside the neurons so that the next pass through is more accurate” frazer

Q: They told me that increasing batch size to 2 makes it process faster

A: “So when that user says you can increase the batch_size what they mean is you can use more than one song to process - i.e. instead of running a single song at batch_size = 1 you can run 2 at the same time (batch_size = 2)

Q: Ah so batch_size param is used for the amount of chunks of the input, so if I set [batch_size] to 4 my audio is chunked into 4"

A: "No, the chunks are split based on defined chunk_size in config (which is more related to STFT settings), and then the script is stacking 'batch_size' number of chunks in same tensor to process them at same time (for inference)." jarredou

A: "Increasing the batch size increases the number of chunks that can be processed at one time, which may speed up processing, but also increases memory usage. It will probably not affect quality." unwa

Inference Colab by jarredou forces batch_size=1. lirc the clicking issue with such value was fixed in MSST repo later, and you can stick to it. Probably in UVR too, since latest patches where newer inference code from MSST was implemented.

Someone was once telling that value not bigger than 2 takes no more than 4GB of VRAM, but it will rather differ from AMD/Intel when the VRAM usage is higher due to lack of garbage collector present in CUDA.

Done by deton24, 2021-2025

*Special thanks to [Audio Separation Discord](#)
(and all the people mentioned in the credits section)*