MDX
Music Demixing Workshop

Hybrid Spectrogram and Waveform Source Separation

Alexandre Défossez[1]

1 Facebook AI Research

In partnership with
ISMIR2021

Abstract

Source separation models either work on the spectrogram or waveform domain. In this work, we show how to perform end-to-end hybrid source separation, letting the model decide which domain is best suited for each source, and even combining both. The proposed hybrid version of the Demucs architecture (Défossez et al., 2019) won the Music Demixing Challenge 2021 organized by Sony. This architecture also comes with additional improvements, such as compressed residual branches, local attention or singular value regularization. Overall, a 1.1 dB improvement of the Signal-To-Distortion (SDR) was observed across all sources as measured on the MusDB HQ dataset (Rafii et al., 2019), an improvement confirmed by human subjective evaluation, with an overall quality rated at 2.83 out of 5 (2.36 for the non hybrid Demucs), and absence of contamination at 3.04 (against 2.37 for the non hybrid Demucs and 2.44 for the second ranking model submitted at the competition).

Introduction

Work on music source separation has recently focused on the task of separating 4 well defined instruments in a supervised manner: drums, bass, vocals and other accompaniments. Recent evaluation campaigns (F.-R. Stöter et al., 2018) have focused on this setting, relying on the standard MusDB benchmark (Rafii et al., 2017). In 2021, Sony organized the Music Demixing Challenge (MDX) (Mitsufuji et al., 2021), an online competition where separation models are evaluated on a completely new and hidden test set composed of 36 tracks.

The challenge featured a number of baselines to start from, which could be divided into two categories: spectrogram or waveform based methods. The former consists in models that are fed with the input spectrogram, either represented by its amplitude, such as Open-Unmix (F.-R. Stöter et al., 2019) and its variant CrossNet Open-Unmix (Sawata et al., 2020), or as the concatenation of its real and imaginary part, a.k.a Complex-As-Channels (CAC) (Choi et al., 2020), such as LaSAFT (Choi et al., 2021). Similarly, the output can be either a mask on the input spectrogram, complex modulation of the input spectrogram (Kong et al., 2021), or the CAC representation.

On the other hand, waveform based models such as Demucs (Défossez et al., 2019) are directly fed with the raw waveform, and output the raw waveform for each of the source. Most of those methods will perform some kind of learnt time-frequency analysis in its first layers through convolutions, such as Demucs and Conv-TasNet (Luo & Mesgarani, 2019), although some will not rely at all on convolutional layers, like Dual-Path RNN (Luo et al., 2020).

Theoretically, there should be no difference between spectrogram and waveform models, in particular when considering CaC (complex as channels), which is only a linear change of base for the input and output space. However, this would only hold true in the limit of having an infinite amount of training data. With a constrained dataset, such as the 100 songs of MusDB, inductive bias can play an important role. In particular, spectrogram methods varies by more than their input and output space. For instance, with a notion of frequency, it is possible to apply convolutions along frequencies, while waveform methods must use layers that are fully connected with respect to their channels. The final test loss being far from zero, there will also be artifacts in the separated audio. Different representations will lead to different artifacts, some being more noticeable for the drums and bass (phase inconsitency for spectrogram methods will make the attack sounds hollow), while others are more noticeable for the vocals (vocals separated by Demucs suffer from crunchy static noise)

In this work, we extend the Demucs architecture to perform hybrid waveform/spectrogram domain source separation. The original U-Net architecture (Ronneberger et al., 2015) is extended to provide two parallel branches: one in the time (temporal) and one in the frequency (spectral) domain. We bring other improvements to the architecture, namely compressed residual branches comprising dilated convolutions (Yu & Koltun, 2016), LSTM (Hochreiter & Schmidhuber, 1997) and attention (Vaswani et al., 2017) with a focus on local content. We measure the impact of those changes on the MusDB benchmark and on the MDX hidden test set, as well as subjective evaluations. Hybrid Demucs ranked 1st at the MDX competition when trained only on MusDB, with 7.32 dB of SDR, and 2nd with extra training data allowed.

Related work

There exist a number of spectrogram based music source separation architectures. Open-Unmix (F.-R. Stöter et al., 2019) is based on fully connected layers and a bi-LSTM. It uses multi-channel Wiener filtering (Nugraha et al., 2016) to reduce artifacts. While the original Open-Unmix is trained independently on each source, a multi-target version exists (Sawata et al., 2020), through a shared averaged representation layer. D3Net (Takahashi & Mitsufuji, 2020) is another architecture, based on dilated convolutions connected with dense skip connections. It was before the competition the best performing spectrogram architecture, with an average SDR of 6.0 dB on MusDB. Unlike previous methods which are based on masking, LaSAFT (Choi et al., 2021) uses Complex-As-Channels (Choi et al., 2020) along with a U-Net (Ronneberger et al., 2015) architecture. It is also single-target, however its weights are shared across targets, using a weight modulation mechanism to select a specific source.

Waveform domain source separation was first explored by (Lluís et al., 2018), as well as (Jansson et al., 2017) and (Stoller et al., 2018) with Wave-U-Net. However, those methods were lagging in term of quality, almost 2 dB behind their spectrogram based competitors. Demucs (Défossez et al., 2019) was built upon Wave-U-Net, using faster strided convolutions rather than explicit downsampling, allowing for a much larger number of channels, but potentially introducing aliasing artifacts (Pons et al., 2021), and extra Gated Linear Unit layers (Dauphin et al., 2017) and biLSTM. For the first time, waveform domain methods surpassed spectrogram ones when considering the overall SDR (6.3 dB on MusDB), although its performance is still inferior on the other and vocals sources. Conv-Tasnet (Luo & Mesgarani, 2019), a model based on masking over a learnt time-frequency representation using dilated convolutions, was also adapted to music source separation by (Défossez et al., 2019), but suffered from more artifacts and lower SDR.

To the best of our knowledge, no other work has studied true end-to-end hybrid source separation, although other teams in the MDX competition used model blending from different domains as a simpler post-training alternative.

Architecture

In this Section we present the structure of Hybrid Demucs, as well as the other additions that were added to the original Demucs architecture.

Original Demucs

The original Demucs architecture (Défossez et al., 2019) is a U-Net (Ronneberger et al., 2015) encoder/decoder structure. A BiLSTM (Hochreiter & Schmidhuber, 1997) is applied between the encoder and decoder to provide long range context. The encoder and decoder have a symetric structure. Each encoder layer is composed of a convolution with a kernel size of 8, stride of 4 and doubling the number of channels (except for the first layer, which sets it to a fix value, typically 48 or 64). It is followed by a ReLU, and a so called 1x1 convolution with Gated Linear Unit activation (Dauphin et al., 2017), i.e. a convolution with a kernel size of 1, where the first half of the channels modulates the second half through a sigmoid. The 1x1 convolution double the channels, and the GLU halves them, keeping them constant overall. Symetrically, a decoder layer sums the contribution from the U-Net skip connection and the previous layer, apply a 1x1 convolution with GLU, then a transposed convolution that halves the number of channels (except for the outermost layer), with a kernel size of 8 and stride of 4, and a ReLU (except for the outermost layer). There are 6 encoder layers, and 6 decoder layers, for processing 44.1 kHz audio. In order to limit the impact of aliasing from the outermost layers, the input audio is upsampled by a factor of 2 before entering the encoder, and downsampled by a factor of 2 when leaving the decoder.

Hybrid Demucs

Overall architecture

Hybrid Demucs extends the original architecture with multi-domain analysis and prediction capabilities. The model is composed of a temporal branch, a spectral branch, and shared layers. The temporal branch takes the input waveform and process it like the standard Demucs. It contains 5 layers, which are going to reduce the number of time steps by a factor of $4^5 = 1024$. Compared with the original architecture, all ReLU activations are replaced by Gaussian Error Linear Units (GELU) (Hendrycks & Gimpel, 2016).

The spectral branch takes the spectrogram obtained from a STFT over 4096 time steps, with a hop length of 1024. Notice that the number of time steps immediately matches that of the output of the encoder of the temporal branch. In order to reduce the frequency dimension, we apply the same convolutions as in the temporal branch, but along the frequency dimension. Each layer reduces by a factor of 4 the number of frequencies, except for the 5th layer, which reduces by a factor of 8. After being processed by the spectral encoder, the signal has only one "frequency" left, and the same number of channels and sample rate as the output of the temporal branch. The temporal and spectral representations are then summed before going through a shared encoder/decoder layer which further reduces by 2 the number of time steps (using a kernel size of 4). Its output serves both as the input of the temporal and spectral decoder. Hybrid Demucs has a dual U-Net structure, with the temporal and spectral branches having their respective skip connections.

The output of the spectral branch is inversed with the ISTFT, and summed with the temporal branch output, giving the final model prediction. Due to this overall design, the model is free to use whichever representation is most conveniant for different parts of the signal, even within one source, and can freely share information between the two representations. The hybrid architecture is represented on Figure 1.

Padding for easy alignment

One difficulty was to properly align the spectral and temporal representations for any input length. For an input length L, kernel size K, stride S and padding on each side P, the output of a convolution is of length $(L - K + 2 * P)/S + 1$. Following the practice from models like MelGAN (Kumar et al., 2019) we pad by $P = (K - S)/2$, giving an output of L/S, so that matching the overall stride is now sufficiant to exactly match the length of the spectral and temporal representations. We apply this padding both for the STFT, and convolution layers in the temporal encoders.

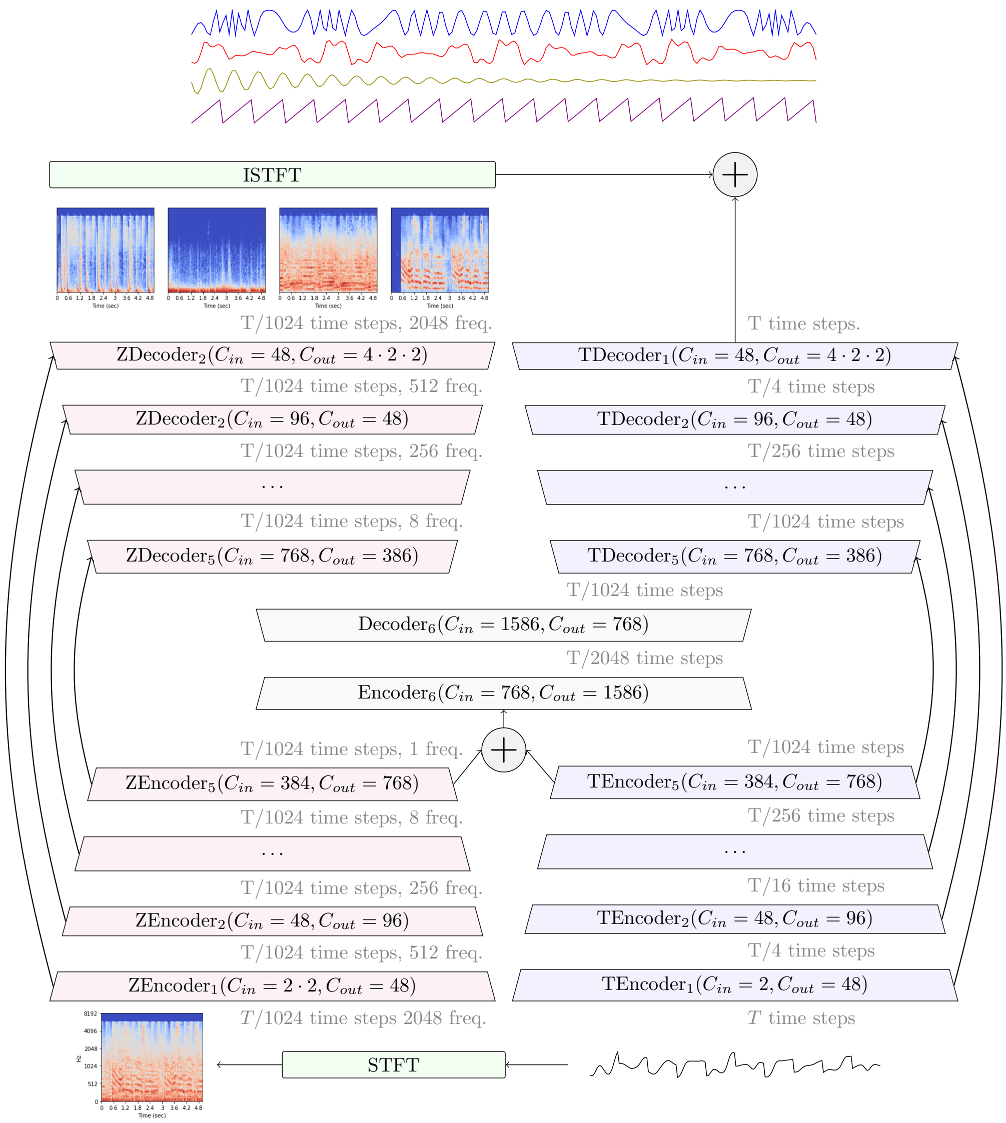

Figure 1: Hybrid Demucs architecture. The input waveform is processed both through a temporal encoder, and first through the STFT followed by a spectral encoder. The two representations are summed when their dimensions align. The opposite happens in the decoder. The output spectrogram go through the ISTFT and are summed with the waveform outputs, giving the final model output. The Z prefix is used for spectral layers, and T prefix for the temporal ones.

Frequency-wise convolutions

In the spectral branch, we use frequency-wise convolutions, dividing the number of frequency bins by 4 with every layer. For simplicity we drop the highest bin, giving 2048 frequency bins after the STFT. The input of the 5th layer has 8 frequency bins, which we reduce to 1 with a convolution with a kernel size of 8 and no padding. It has been noted that unlike the time axis, the distribution of musical signals is not truely invariant to translation along the frequency axis. Instruments have specific pitch range, vocals have well defined formants etc. To account for that, (Isik et al., 2020) suggest injecting an embedding of the frequency before applying the convolution. We use the same approach, with the addition that we smooth the initial embedding so that close frequencies have similar embeddings. We inject this embedding just before the second encoder layer. We also investigated using specific weights for different frequency bands. This however turned out more complex for a similar result.

Spectrogram representation

We investigated both with representing the spectrogram as complex numbers (Choi et al., 2020) or as amplitude spectrograms. For this second option, we use Wiener filtering (Nugraha et al., 2016). We use Open-Unmix differentiable implementation of this filtering (F.-R. Stöter et al., 2019), which uses an iterative estimation procedure. Using more iterations at evaluation time is usually optimal, but sadly doesn't work well with the hybrid approach, as changing the spectrogram output, without the waveform output being able to adapt will drastically reduce the SDR, and using a high number of iterations at train time is prohibitively slow. In all cases, we differentiably transform the spectrogram branch output to a waveform, summed to the waveform branch output, and the final loss is applied in the waveform domain.

Compressed residual branches

The original Demucs encoder layer is composed of a convolution with kernel size of 8 and stride of 4, followed by a ReLU, and of a convolution with kernel size of 1 followed by a GLU. Between those two convolutions, we introduce two compressed residual branches, composed of dilated convolutions, and for the innermost layers, a biLSTM with limited span and local attention. Remember that after the first convolution of the 5th layer, the temporal and spectral branches have the same shape. The 5th layer of each branch actually only contains this convolution, with the compressed residual branch and 1x1 convolution being shared.

Inside a residual branch, all convolutions are with respect to the time dimension, and different frequency bins are processed separately. There are two compressed residual branch per encoder layer. Both are composed of a convolution with a kernel size of 3, stride of 1, dilation of 1 for the first branch and 2 for the second, and 4 times less output dimensions than the input, followed by layer normalization (Ba et al., 2016) and a GELU activation.

For the 5th and 6th encoder layers, long range context is processed through a local attention layer (see definition hereafter) as well as a biLSTM with 2 layers, inserted with a skip connection, and with a maximum span of 200 steps. In practice, the input is splitted into frames of 200 time steps, with a stride of 100 steps. Each frame is processed concurrently, and for any time step, the output from the frame for which it is the furthest away from the edge is kept.

Finally, and for all layers, a final convolution with a kernel size of 1 outputs twice as many channels as the input dimension of the residual branch, followed by a GLU. This output is then summed with the original input, after having been scaled through a LayerScale layer (Touvron et al., 2021), with an initial scale of $1\mathrm{e}{-3}$. A complete representation of the compressed residual branches is given on Figure 2.

Local attention

Local attention builds on regular attention (Vaswani et al., 2017) but replaces positional embedding by a controllable penalty term that penalizes attending to positions that are far away. Formally, the attention weights from position i to position j is given by

$$w_{i,j} = \mathrm{softmax}(Q_i^T K_j - \sum_{k=1}^{4} k\beta_{i,k}|i-j|)$$

where Q_i are the queries and K_j are the keys. The values $\beta_{i,k}$ are obtained as the output of a linear layer, initialized so that they are initially very close to 0. Having multiple $\beta_{i,k}$ with different weights k allows the network to efficiently reduce its receptive field without requiring $\beta_{i,k}$ to take large values. In practice, we use a sigmoid activation to derive the values $\beta_{i,k}$.

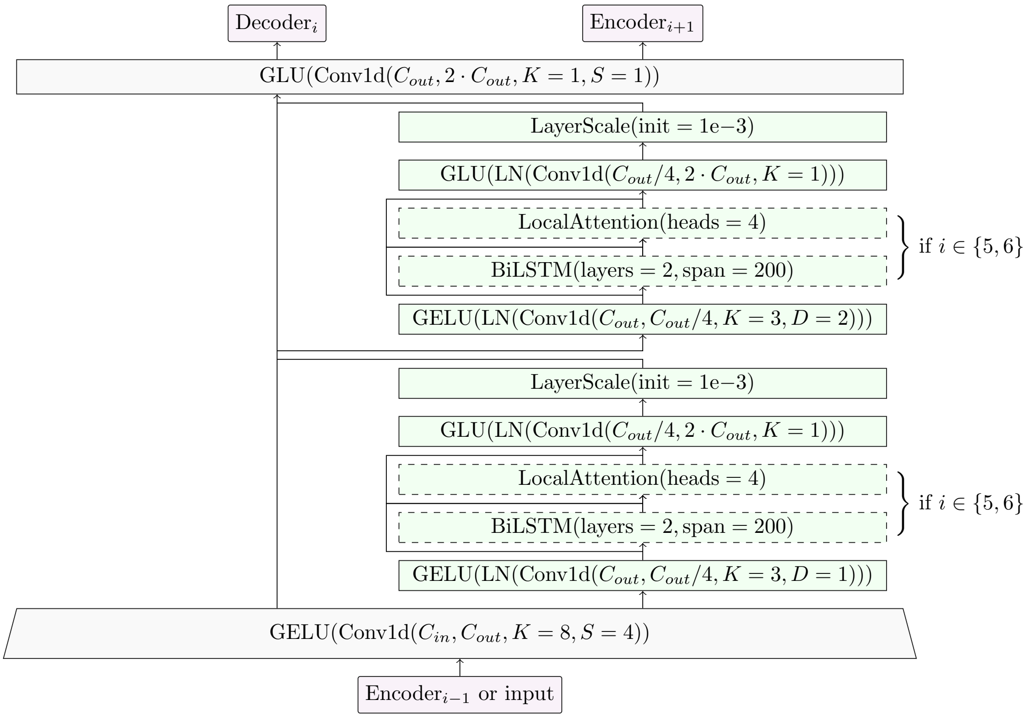

Figure 2: Representation of the compressed residual branches that are added to each encoder layer. For the 5th and 6th layer, a BiLSTM and a local attention layer are added.

Stabilizing training

We observed that Demucs training could be unstable, especially as we added more layers and increased the training set size with 150 extra songs. Loading the model just before its divergence point, we realized that the weights for the innermost encoder and decoder layers would get very large eigen values.

A first solution is to use group normalization (with 4 groups) just after the non residual convolutions for the layers 5 and 6 of the encoder and the decoder. Using normalization on all layers deteriorates performance, but using it only on the innermost layers seems to stabilize training without hurting performance. Interestingly, when the training is stable (in particular when trained only on MusDB), using normalization was at best neutral with respect to the separation score, but never improved it, and considerably slowed down convergence during the first half of the epochs. When the training was unstable, using normalization would improve the overall performance as it allows the model to train for a larger number of epochs.

A second solution we investigated was to use singular value regularization (Yoshida & Miyato, 2017). While previous work used the power method iterative procedure, we obtained better and faster approximations of the largest singular value using a low rank SVD method (Halko et al., 2011). This solution has the advantage of always improving generalization, even when the training was already stable. Sadly, it was not sufficient on its own to remove entirely instabilities, but only to reduce them. Another down side was the longer training time due to the extra low rank SVD evaluation. In the end, in order to both achieve the best performance and remove entirely training instabilities, the two solutions were combined.

Experimental Results

Datasets

The 2021 MDX challenge (Mitsufuji et al., 2021) offered two tracks: Track A, where only MusDB HQ (Rafii et al., 2019) could be used for training, and Track B, where any data could be used. MusDB HQ, released under mixed licensing[1] is composed of 150 tracks, including 86 for the train set, 14 for the valid, and 50 for the test set. For Track B, we additionally trained using 150 tracks for an internal dataset, and repurpose the test set of MusDB as training data, keeping only the original validation set for model selection. Models are evaluated either through the MDX AI Crowd API[2], or on the MusDB HQ test set.

Realist remix of tracks

Metrics

The MDX challenge introduced a novel Signal-To-Distortion measure. Another SDR measure existed, as introduced by (Vincent et al., 2006). The advantage of the new definition is its simplicty and fast evaluation. The new definition is simply defined as

$$SDR = 10 \log_{10} \frac{\sum_n \|\mathbf{s}(n)\|^2 + \epsilon}{\sum_n \|\mathbf{s}(n) - \hat{\mathbf{s}}(n)\|^2 + \epsilon}, \tag{1}$$

where s is the ground truth source, $\hat{s}$ the estimated source, and n the time index. In order to reliably compare to previous work, we will refer to this new SDR definition as *nSDR*, and to the old definition as *SDR*.

Note that when using nSDR on the MDX test set, the metric is defined as the average across all songs. On the other hand, the evaluation on the MusDB test set follows the traditional median across the songs of the median over all 1 second segments of each song.

Models

The model submitted to the competitions were actually bags of 4 models. For Track A, we had to mix hybrid and non hybrid Demucs models, as the hybrid ones were having worse performance on the bass source. On Track B, we used only hybrid models, as the extra training data allowed them to perform better for all sources. Note that a mix of Hybrid models using CaC or Wiener filtering were used, mostly because it was too costly to reevaluate all models for the competition. For details on the exact architecture and hyper-parameter used, we refer the reader to our Github repository facebookresearch/demucs.

For the baselines, we report the numbers from the top participants at the MDX competition (Mitsufuji et al., 2021). We focus particularly on the KUIELAB-MDX-Net model, which came in second. This model builds on (Choi et al., 2020) and combines a pure spectrogram model with the prediction from the original Demucs (Défossez et al., 2019) model for the drums and bass sources. When comparing models on MusDB, we also report the numbers for some of the best performing methods outside of the MDX competition, namely D3Net (Takahashi & Mitsufuji, 2020) and ResUNetDecouple+ (Kong et al., 2021), as well as the original Demucs model (Défossez et al., 2019). Note that those models were evaluated on MusDB (not HQ) which lacks the frequency content between 16 kHz and 22kHz. This can bias the metrics.

Results on MDX

We provide the results from the top participants at the MDX competition on Table Table 1 for the track A (trained on MusDB HQ only) and on Table Table 2 for track B (any training

[1] https://github.com/sigsep/website/blob/master/content/datasets/assets/tracklist.csv
[2] https://www.aicrowd.com/challenges/music-demixing-challenge-ismir-2021

data). We also report for track A the metrics for the Demucs architecture improved with the residual branches, but without the spectrogram branch. The hybrid approach especially improves the nSDR on the `Other` and `Vocals` source. Despite this improvement, the Hybrid Demucs model is still performing worse than the KUIELAB-MDX-Net on those two sources. On Track B, we notice again that the Hybrid Demucs architecture is very strong on the `Drums` and `Bass` source, while lagging behind on the `Other` and `Vocals` source.

Table 1: Results of Hybrid Demucs on the MDX test set, when trained only on MusDB (track A) using the nSDR metric. "improved Demucs" consist in a bag of Demucs models without any hybrid model, i.e. only residual branches etc.

Method	All	Drums	Bass	Other	Vocals
Hybrid Demucs	**7.33**	**8.04**	**8.12**	5.19	7.97
improved Demucs	6.82	7.58	7.79	4.70	7.21
KUIELAB-MDX-Net	7.24	7.17	7.23	**5.63**	**8.90**
Music_AI	6.88	7.37	7.27	5.09	7.79

Table 2: Results of Hybrid Demucs on the MDX test set, when trained with extra training (track B) using the nSDR metric.

Method	All	Drums	Bass	Other	Vocals
Hybrid Demucs	8.11	**8.85**	**8.86**	5.98	8.76
KUIELAB-MDX-Net	7.37	7.55	7.50	5.53	8.89
AudioShake	**8.33**	8.66	8.34	**6.51**	**9.79**

Results on MusDB

We show on Table Table 3 the SDR metrics as measured on the MusDB dataset. Again, Hybrid Demucs achieves the best performance for the `Drums` and `Bass` source, while improving quite a lot over waveform only Demucs for the `Other` and `Vocals`. Interestingly, the best performance on the `Vocals` source is achieved by ResUNetDecouple+ (Kong et al., 2021), which uses a novel approach based on complex modulation of the input spectrogram.

Table 3: Comparison on the MusDB (HQ for Hybrid Demucs) test set, using the original SDR metric. This includes methods that did not participate in the competition. "Mode" indicates if waveform (W) or spectrogram (S) domain is used.

Method	Mode	All	Drums	Bass	Other	Vocals
Hybrid Demucs	S+W	7.33	**8.04**	**8.12**	5.19	7.97
Original Demucs	W	6.28	6.86	7.01	4.42	6.84
KUIELAB-MDX-Net	S+W	**7.41**	7.09	7.38	**6.29**	8.88
D3Net	S	6.01	7.01	5.25	4.53	7.24
ResUNetDecouple+	S	6.73	6.62	6.04	5.29	**8.98**

Human evaluations

We also performed Mean Opinion Score human evaluations. We re-use the same protocol as in (Défossez et al., 2019): we asked human subjects to evaluate a number of samples based on two criteria: the absence of artifacts, and the absence of bleeding (contamination). Both are evaluated on a scale from 1 to 5, with 5 being the best grade. Each subject is tasked with evaluating 25 samples of 12 seconds, drawn randomly from the 50 test set tracks of MusDB. All subjects have a strong experience with music (amateur and professional musicians, sound engineers etc). The results are given on Table Table 4 for the quality, and Table 5 for the bleeding. We observe strong improvements over the original Demucs, although we observe some regression on the bass source when considering quality. The model KUIELAb-MDX-Net that came in second at the MDX competition performs the best on vocals. The Hybrid Demucs architecture however reduces by a large amount bleeding across all sources.

Table 4: Mean Opinion Score results when asking to rate the quality and absence of artifacts in the generated samples, from 1 to 5 (5 being the best grade). Standard deviation is around 0.15.

Method	All	Drums	Bass	Other	Vocals
Ground Truth	4.12	4.12	4.25	3.92	4.18
Hybrid Demucs	**2.83**	**3.18**	2.58	**2.98**	2.55
KUIELAB-MDX-Net	**2.86**	2.70	2.68	**2.99**	**3.05**
Original Demucs	2.36	2.62	**2.89**	2.31	1.78

Table 5: Mean Opinion Score results when asking to rate the absence of bleeding between the sources, from 1 to 5 (5 being the best grade). Standard deviation is around 0.15.

Method	All	Drums	Bass	Other	Vocals
Ground Truth	4.40	4.51	4.52	4.13	4.43
Hybrid Demucs	**3.04**	**2.95**	**3.25**	**3.08**	**2.88**
KUIELAB-MDX-Net	2.44	2.23	2.19	2.64	2.66
Original Demucs	2.37	2.24	2.96	1.99	2.46

Conclusion

We introduced a number of architectural changes to the Demucs architecture that greatly improved the quality of source separation for music. On the MusDB HQ benchark, the gain is around 1.1 dB. Those changes include compressed residual branches with local attention and chunked biLSTM, and most importantly, a novel hybrid spectrogram/temporal domain U-Net structure, with parallel temporal and spectrogram branches, that merge into a common core. Those changes allowed to achieve the first rank at the 2021 Sony Music DemiXing challenge, and translated into strong improvements of the overall quality and absence of bleeding between sources as measured by human evaluations. For all its gain, one limitation of our approach is the increased complexity of the U-Net encoder/decoder, requiring careful alignement of the temporal and spectral signals through well shaped convolutions.

References

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv Preprint arXiv:1607.06450*.

Choi, W., Kim, M., Chung, J., & Jung, S. (2021). LaSAFT: Latent source attentive frequency transformation for conditioned source separation. *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 171–175.

Choi, W., Kim, M., Chung, J., Lee, D., & Jung, S. (2020). Investigating u-nets with various intermediate blocks for spectrogram-based singing voice separation. In ISMIR (Ed.), *21th international society for music information retrieval conference*.

Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017). Language modeling with gated convolutional networks. *Proceedings of the International Conference on Machine Learning*.

Défossez, A., Usunier, N., Bottou, L., & Bach, F. (2019). Music source separation in the waveform domain. *arXiv Preprint arXiv:1911.13254*.

Halko, N., Martinsson, P.-G., & Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, *53*(2), 217–288.

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv Preprint arXiv:1606.08415*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Isik, U., Giri, R., Phansalkar, N., Valin, J.-M., Helwani, K., & Krishnaswamy, A. (2020). Poconet: Better speech enhancement with frequency-positional embeddings, semi-supervised conversational data, and biased loss. *arXiv Preprint arXiv:2008.04470*.

Jansson, A., Humphrey, E., Montecchio, N., Bittner, R., Kumar, A., & Weyde, T. (2017). Singing voice separation with deep u-net convolutional networks. *ISMIR 2018*.

Kong, Q., Cao, Y., Liu, H., Choi, K., & Wang, Y. (2021). Decoupling magnitude and phase estimation with deep ResUNet for music source separation. *22th International Society for Music Information Retrieval Conference*.

Kumar, K., Kumar, R., Boissiere, T. de, Gestin, L., Teoh, W. Z., Sotelo, J., Brébisson, A. de, Bengio, Y., & Courville, A. C. (2019). MelGAN: Generative adversarial networks for conditional waveform synthesis. *Advances in Neural Information Processing Systems*.

Lluís, F., Pons, J., & Serra, X. (2018). End-to-end music source separation: Is it possible in the waveform domain? *arXiv Preprint arXiv:1810.12*.

Luo, Y., Chen, Z., & Yoshioka, T. (2020). Dual-path rnn: Efficient long sequence modeling for time-domain single-channel speech separation. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 46–50.

Luo, Y., & Mesgarani, N. (2019). Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Mitsufuji, Y., Fabbro, G., Uhlich, S., & Stöter, F.-R. (2021). Music demixing challenge 2021. *arXiv Preprint arXiv:2108.13559*.

Nugraha, A. A., Liutkus, A., & Vincent, E. (2016). Multichannel music separation with deep neural networks. *Signal Processing Conference (EUSIPCO), 2016 24th European*.

Pons, J., Pascual, S., Cengarle, G., & Serrà, J. (2021). Upsampling artifacts in neural audio synthesis. *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3005–3009.

Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., & Bittner, R. (2019). *MUSDB18-HQ - an uncompressed version of MUSDB18*. https://doi.org/10.5281/zenodo.3338373

Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., & Bittner, R. (2017). *The MUSDB18 corpus for music separation*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*.

Sawata, R., Uhlich, S., Takahashi, S., & Mitsufuji, Y. (2020). *All for one and one for all: Improving music separation by bridging networks*. http://arxiv.org/abs/2010.04228

Stoller, D., Ewert, S., & Dixon, S. (2018). Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv Preprint arXiv:1806.03185*.

Stöter, F.-R., Liutkus, A., & Ito, N. (2018). The 2018 signal separation evaluation campaign. *14th International Conference on Latent Variable Analysis and Signal Separation*.

Stöter, F.-R., Uhlich, S., Liutkus, A., & Mitsufuji, Y. (2019). Open-unmix - a reference implementation for music source separation. *Journal of Open Source Software*. https://doi.org/10.21105/joss.01667

Takahashi, N., & Mitsufuji, Y. (2020). D3Net: Densely connected multidilated DenseNet for music source separation. *arXiv Preprint arXiv:2010.01733*.

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., & Jégou, H. (2021). Going deeper with image transformers. *arXiv Preprint arXiv:2103.17239*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 5998–6008.

Vincent, E., Gribonval, R., & Févotte, C. (2006). Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech and Language Processing*.

Yoshida, Y., & Miyato, T. (2017). Spectral norm regularization for improving the generalizability of deep learning. *arXiv Preprint arXiv:1705.10941*.

Yu, F., & Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. *ICLR*.