2025 edition

Deep Learning for Music Analysis and Generation

Source Separation: Quick Notes

(audio → audio)

Yi-Hsuan Yang Ph.D.
yhyangtw@ntu.edu.tw

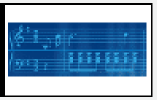
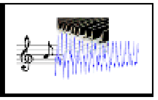
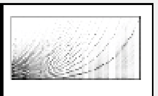
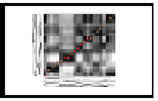
Objectives

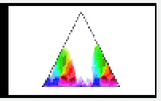
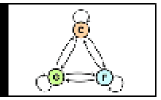
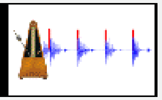
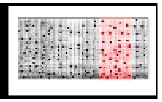
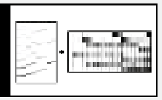
- To be familiar with how people **work on spectrograms**

- It's a type of *conditional* audio **generation** task (with *strong condition*)
  - **"audio (mixture) → audio (stem)"**
  - Will talk about other types of audio generation tasks in the forthcoming lectures

Reference 1: FMP Notebook

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C8/C8.html

Part	Title	Notions, Techniques & Algorithms	HTML	IPYNB
B 	Basics	Basic information on Python, Jupyter notebooks, Anaconda package management system, Python environments, visualizations, and other topics	[html]	[ipynb]
0 	Overview	Overview of the notebooks (https://www.audiolabs-erlangen.de/FMP)	[html]	[ipynb]
1 	Music Representations	Music notation, MIDI, audio signal, waveform, pitch, loudness, timbre	[html]	[ipynb]
2 	Fourier Analysis of Signals	Discrete/analog signal, sinusoid, exponential, Fourier transform, Fourier representation, DFT, FFT, STFT	[html]	[ipynb]
3 	Music Synchronization	Chroma feature, dynamic programming, dynamic time warping (DTW), alignment, user interface	[html]	[ipynb]

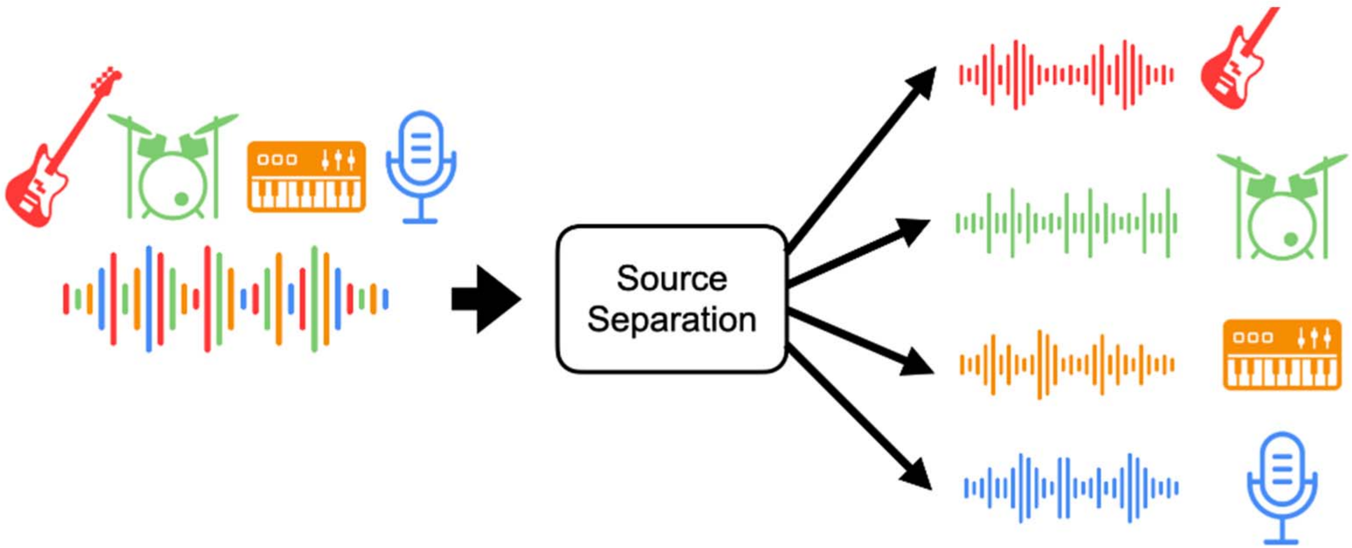
Part	Title	Notions, Techniques & Algorithms	HTML	IPYNB
4 	Music Structure Analysis	Similarity matrix, repetition, thumbnail, homogeneity, novelty, evaluation, precision, recall, F-measure, visualization, scape plot	[html]	[ipynb]
5 	Chord Recognition	Harmony, music theory, chords, scales, templates, hidden Markov model (HMM), evaluation	[html]	[ipynb]
6 	Tempo and Beat Tracking	Onset, novelty, tempo, tempogram, beat, periodicity, Fourier analysis, autocorrelation	[html]	[ipynb]
7 	Content-Based Audio Retrieval	Identification, fingerprint, indexing, inverted list, matching, version, cover song	[html]	[ipynb]
8 	Musically Informed Audio Decomposition	Harmonic/percussive separation, signal reconstruction, instantaneous frequency, fundamental frequency (F0), trajectory, nonnegative matrix factorization (NMF)	[html]	[ipynb]

Reference 2: ISMIR 2020 Tutorial

https://source-separation.github.io/tutorial/landing.html

Open Source Tools & Data for Music Source Separation

By Ethan Manilow, Prem Seetharaman, and Justin Salamon



Outline

- **Basics**
- Tools

What is Source Separation?

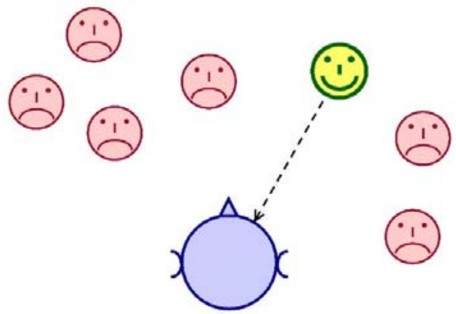
https://source-separation.github.io/tutorial/intro/src_sep_101.html

- The process of isolating individual sounds (*sources*) in an auditory mixture of multiple sounds

- *Underdetermined* problem
  - Fewer observations $y(t)$ (1 or 2; mono or stereo) than the required outcomes $x_i(t)$ (e.g., 4)

$$y(t) = \sum_{i=1}^{N} x_i(t).$$

- In speech: cocktail party effect



Cocktail-Party-Effekt:
Herausfiltern einer Schallquelle bei Anwesenheit mehrere Schallquellen

Demo: Source Separation

https://www.gaudiolab.com/technology/source-separation

Why Source Separation?

https://source-separation.github.io/tutorial/intro/src_sep_101.html

- Benefit **downstream MIR** problems (e.g., *singer classification*)

  - automatic music transcription [PAB+02,MSP20],
  - lyric and music alignment [FGO+06],
  - musical instrument detection [HKV09],
  - lyric recognition [MV10],
  - automatic singer identification [WWollmerS11,HL15,SDL19],
  - vocal activity detection [SED18a],
  - fundamental frequency estimation [JBEW19], and
  - understanding the predictions of black-box audio models. [HMW20a,HMW20b]

Why Source Separation?

https://source-separation.github.io/tutorial/intro/src_sep_101.html

- Benefit **music generation**
  - Re-mix of the sources
  - Up-mix: stereo to 5.1-channel
  - Replacement of some of the sources
  - Audio editing
- **Active music listening**

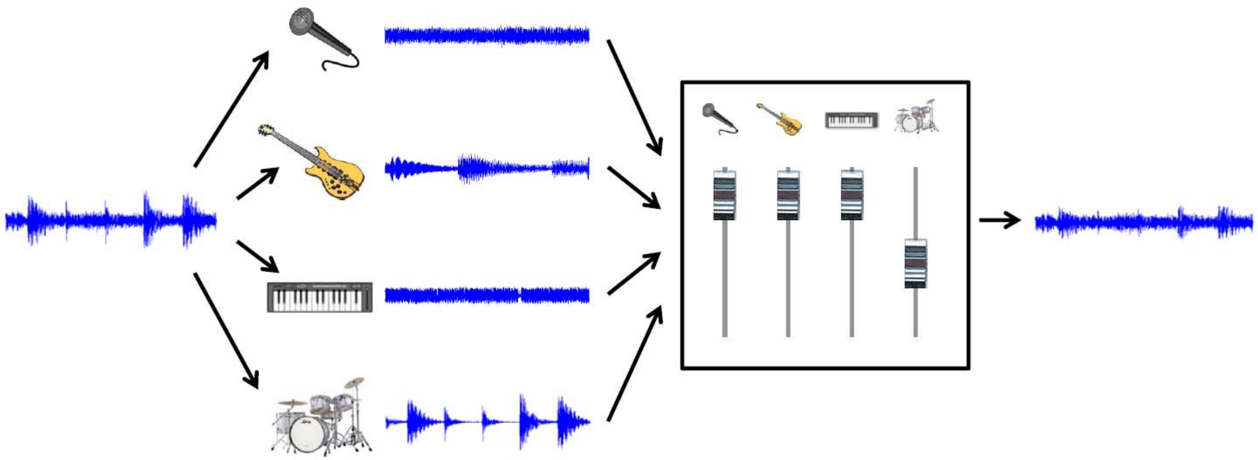
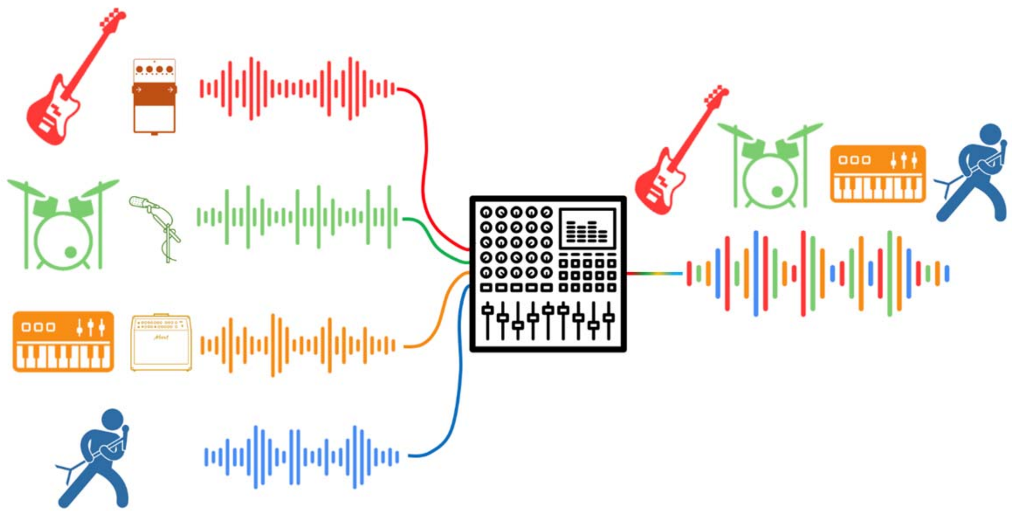


Figure from [Mueller, FPM, Chapter 8, Springer 2015]

Why Source Separation is Difficult?

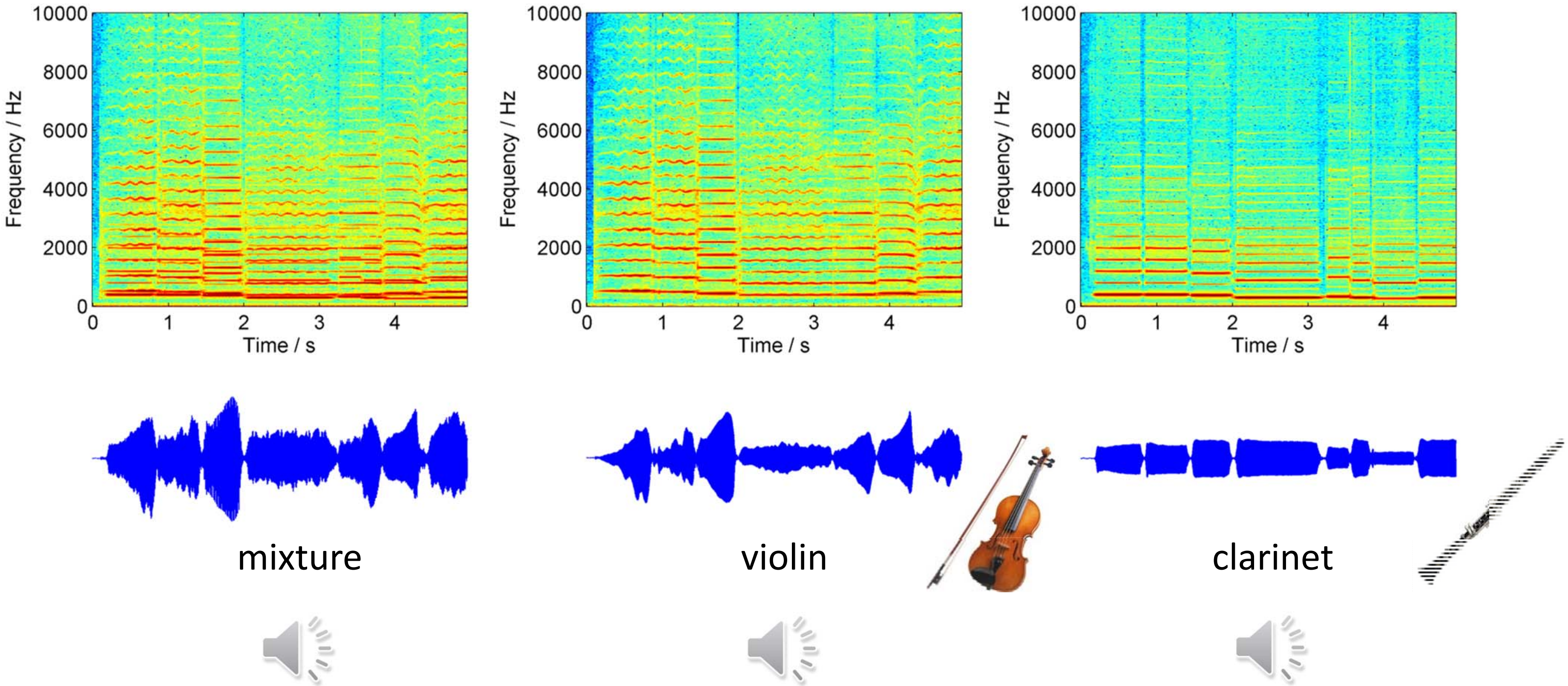
https://source-separation.github.io/tutorial/intro/src_sep_101.html

- Sources in music are **highly correlated** (harmonically and rhythmically)
- The **mixing** of music signals is **complex** and **non-linear**
  - Reverb, EQ, …
  - Don't know how the mixing was done
- It's actually an **audio-domain music generation** problem
  - Instrument recognition (*discriminative*) vs. instrument separation (*generative*)
  - The bar for quality can be high



Why Source Separation is Difficult?

- Sources in music are **highly correlated** (harmonically and rhythmically)



Why Source Separation is Difficult?

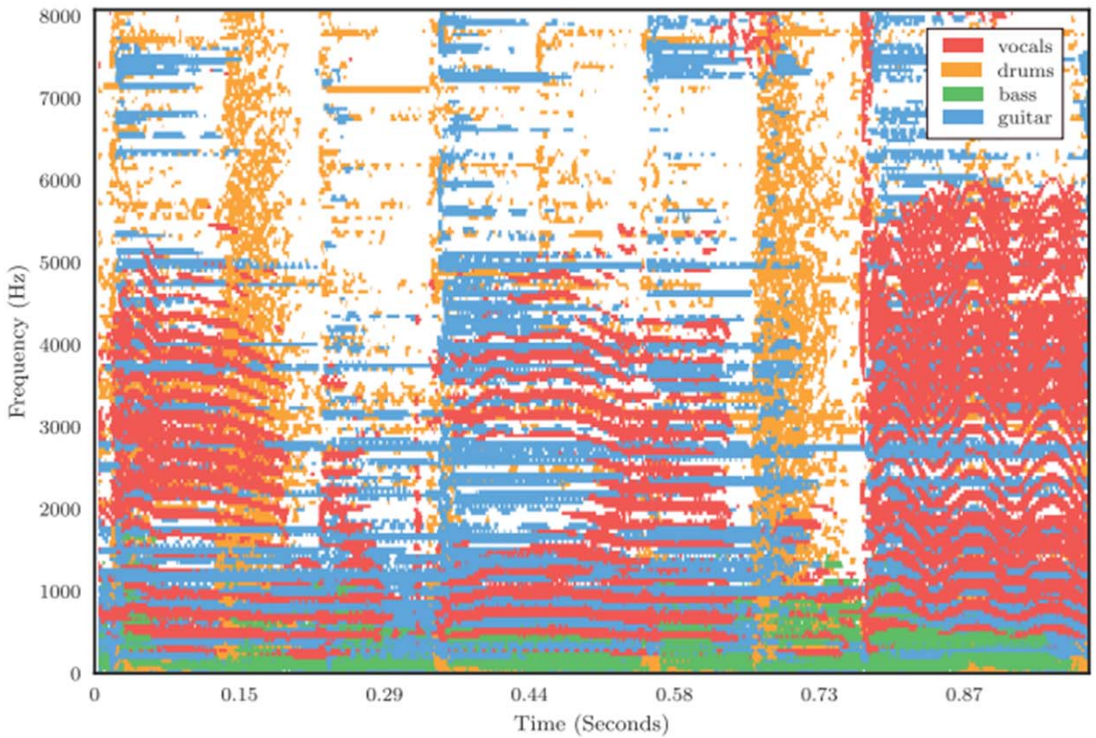
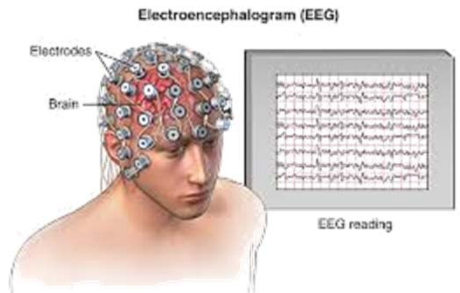


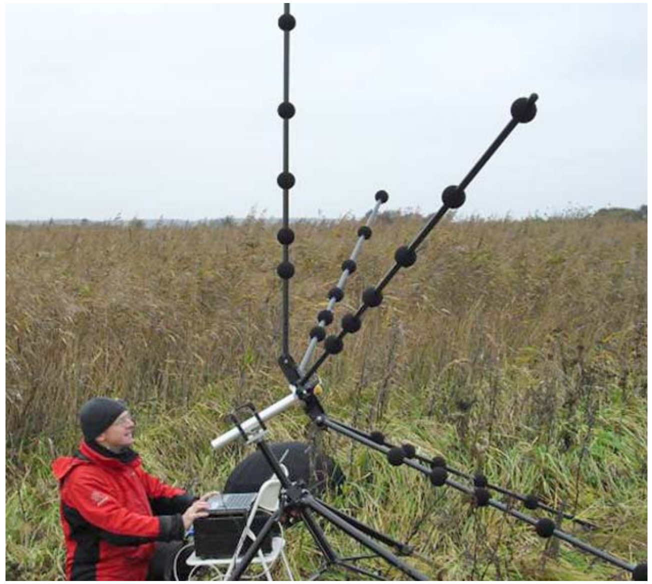
Figure 1: Representation of a music mixture in the time-frequency domain. The dominant musical source in each time-frequency bin is displayed with a different color.

Ref: Cano et al, “Musical source separation: An introduction,” IEEE Signal Processing Magazine 2019

Types of Separation Problems

- #sources vs. #output channels
  - Overdetermined vs **underdetermined**
- Amount of side information
  - **Blind** source separation vs. **informed** source separation
    - *Score* informed
    - *Lyrics* informed
    - *Melody* informed
  - Most people work on blind & underdetermined source separation
- Online or offline





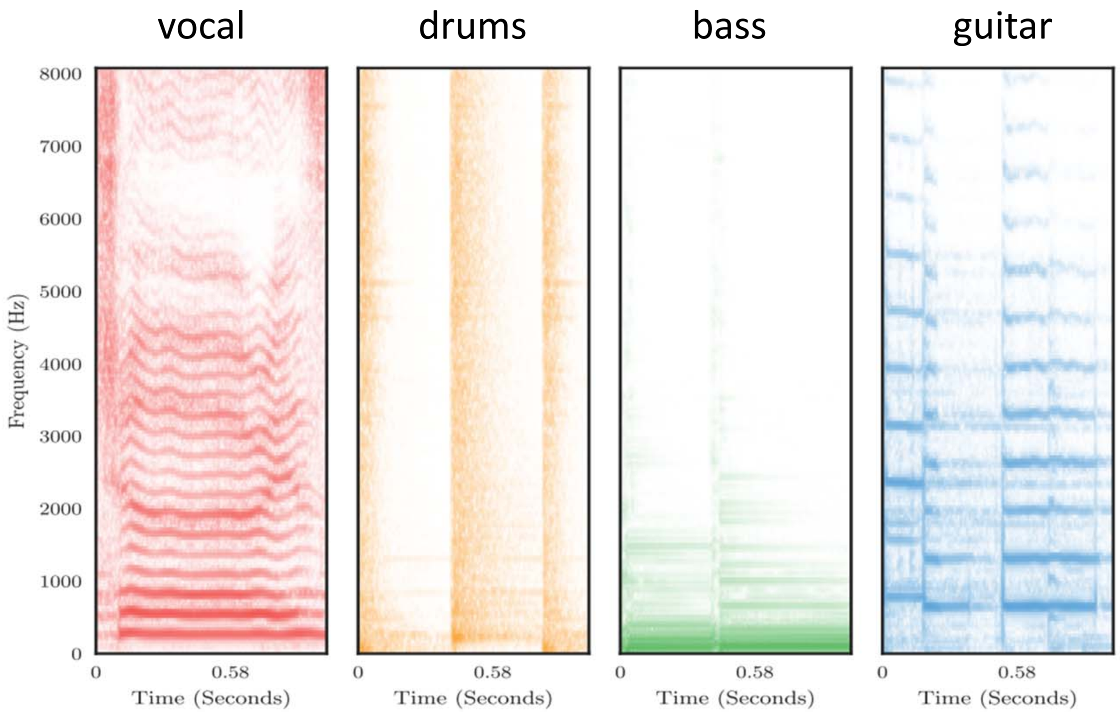
Types of Separation Problems

- What the #output channels are
  - **Two** stems: vocal vs. *non-vocal*
    - or: lead vs. *accompaniments*
  - **Four** stems: vocal, drums, bass, and *others*
  - Beyond four stems
  - Most challenging: Uncertain *number* and *class* of output channels

- Do different output channels correspond to different instruments?
  - Not always
    - Choral music separation (soprano, alto, tenor, and bass)
    - Speaker separation

Clues for Monaural Source Separation

- Different sound sources may have different *time-frequency characteristics* (timbre, pitch range, etc)



Ref: Cano et al, “Musical source separation: An introduction,” IEEE Signal Processing Magazine 2019

Spectrogram-based and Waveform-based

https://paperswithcode.com/sota/music-source-separation-on-musdb18

- STFT = magnitude + phase
- People tend to use the **magnitude STFT**
  - provides rich info as a time-frequency representation
- Phase is hard to model, but **phase is needed** here
- Also hard model waveforms (in early days)

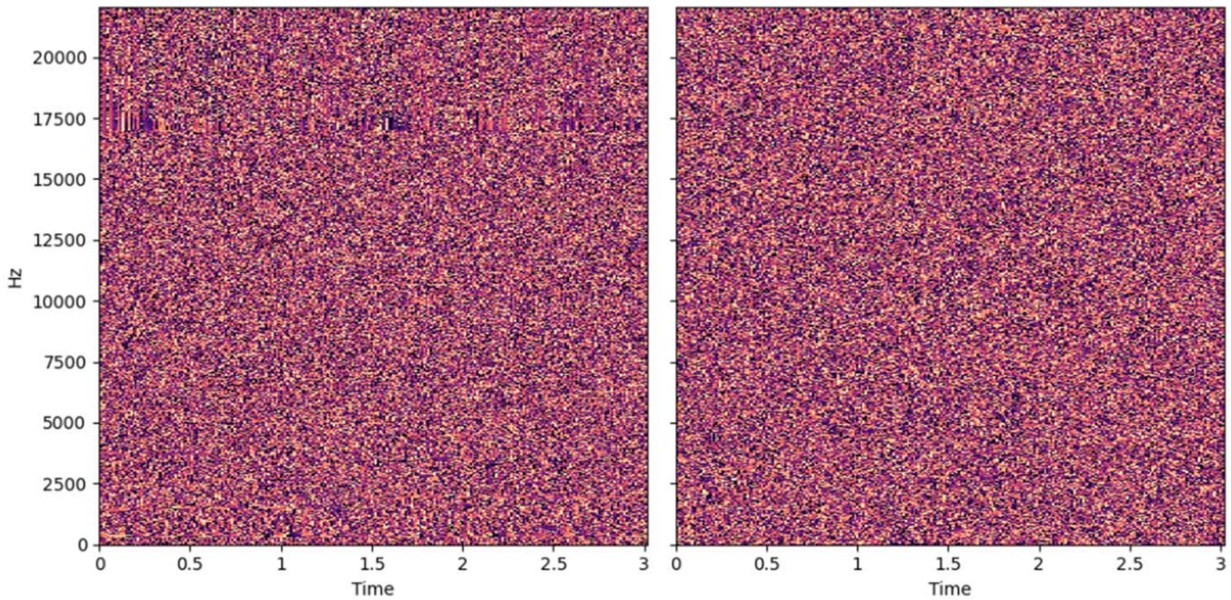


Fig. 17 The structure of phase within an STFT makes it hard to model. One of these two images shows the phase component of an STFT and another shows random noise. Can you guess which is which?

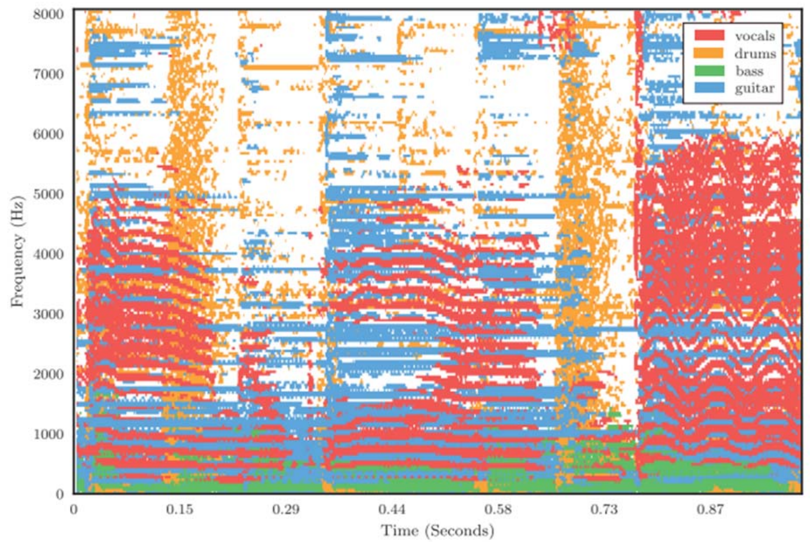
Approach

- **Traditional** methods
  - *Unsupervised*: rule-based, model-based
  - Faster, light-weight, but limited performance
  - Usually work on spectrograms

- **Deep learning** based methods
  - *Supervised*: learn from “clean sources”
  - Mixture in, clean sources out
  - Better result
  - Work on spectrograms, waveforms, or both

ML/DL Viewpoint: Time-frequency Classification

- **Per song:** genre classification
- **Per short-time chunk:** instrument activity detection
- **Per time-frequency point:**
  f0 estimation, multi-pitch estimation,
  source separation



- Input and output are of the same shape
- But, how about **phase?**

Deal with Phase: Approaches

https://source-separation.github.io/tutorial/basics/phase.html

- **Copy the phase from the mixture**
- **Given the magnitude, estimate the phase** (this is called a *"vocoder"*)
- Work on **complex-valued spectrograms**
- Work on **audio waveforms**, not magnitude spectrograms

Supervised Approach

- Learn from **paired data** of {mixture, stems}

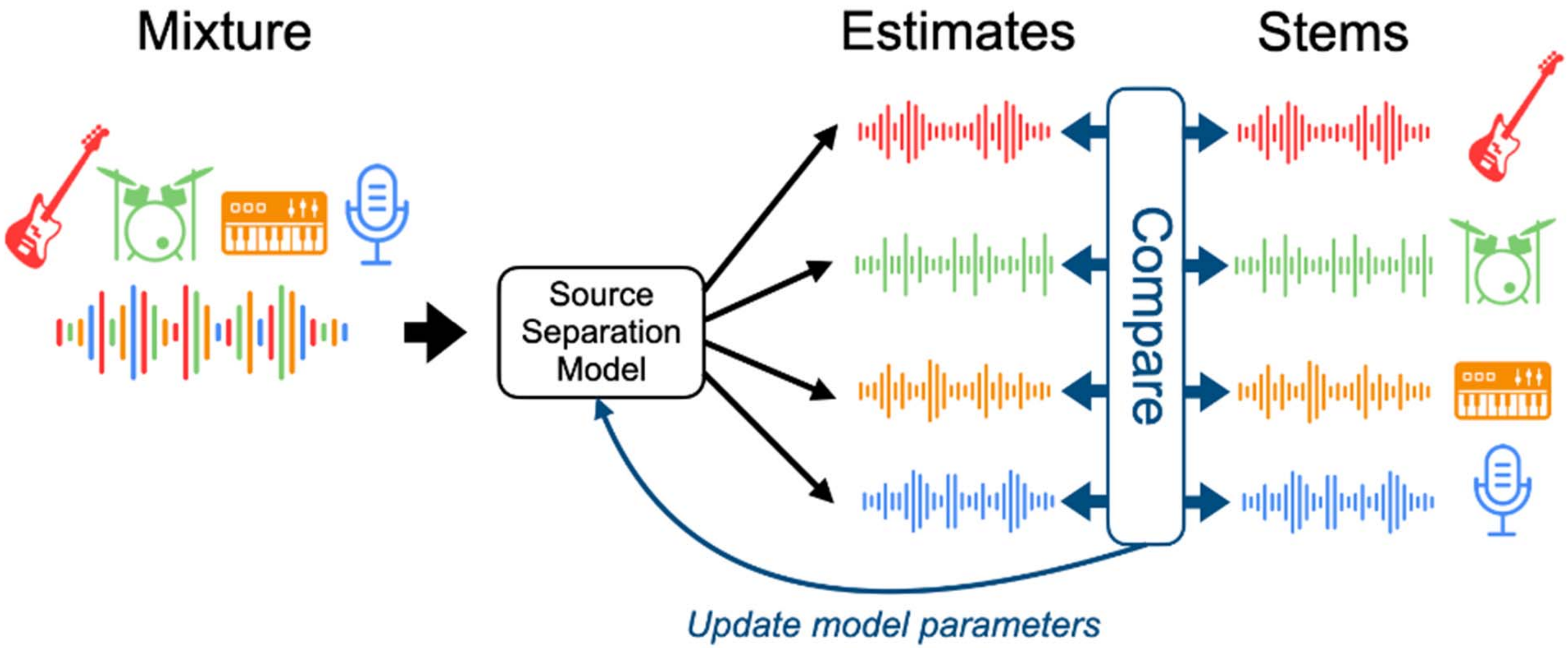
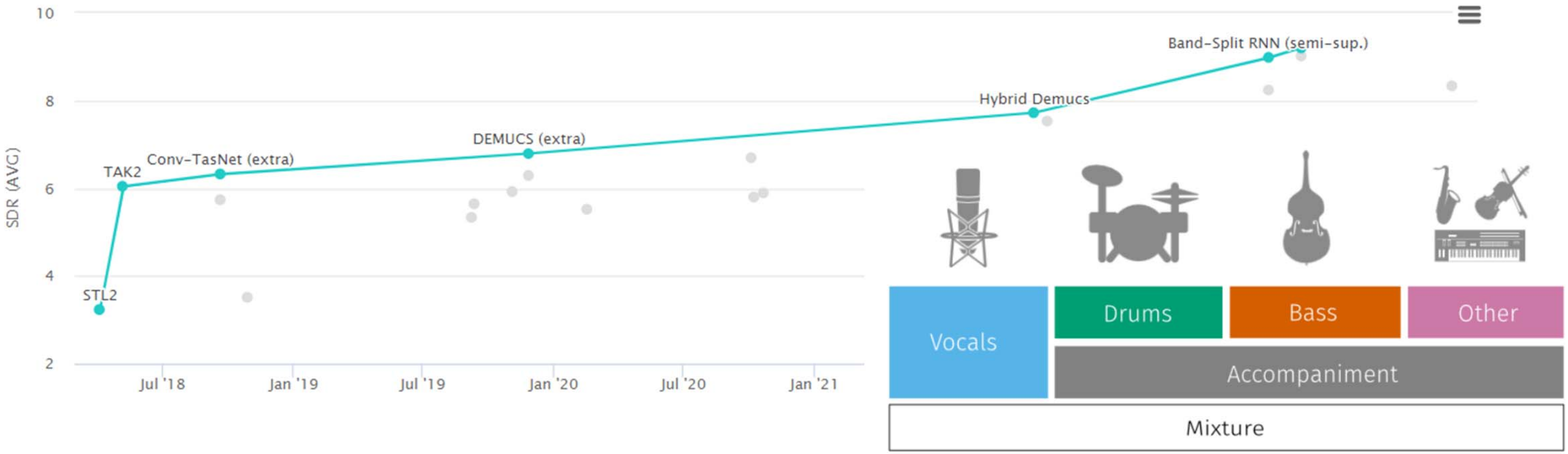


Figure source: https://source-separation.github.io/tutorial/approaches/deep/introduction.html

Benchmark

https://paperswithcode.com/sota/music-source-separation-on-musdb18



Rank	Model	SDR (avg)	SDR (vocals)	SDR (drums)	SDR (bass)	SDR (other)	Extra Training Data	Paper	Code	Result	Year	Tags
1	**Sparse HT Demucs** (fine tuned)	9.20	9.37	10.83	10.47	6.41	✓	Hybrid Transformers for Music Source Separation			2022	

Evaluation Metrics

https://source-separation.github.io/tutorial/basics/evaluation.html

- **Computed in the time-domain**
- **Source-to-Distortion Ratio** (**SDR**)
- Source-to-Interference Ratio (SIR)
- Source-to-Artifact Ratio (SAR)
  - true sources: **a**, **b**
  - estimated sources: **ae**, **be**
  - SDR(a): how **ae** is similar to **a**
  - SIR(a): how **ae** is similar to **b**
  - SAR(a): how **ae** is not similar to either **a or b**

Evaluation Metrics

https://docs.google.com/presentation/d/1XLC7SyGMRfOj3WwJaiyaYFOwCI69w4aXWLYl7UEsvXQ/

Source Separation Metrics: What are they really measuring?

Keynote presentation at the 2021 Music Demixing Workshop

$$\hat{s} = s_{target} + e_{interf} + e_{artif}$$

$$s_{target} = P_s \hat{s} = \frac{< \hat{s}, s >}{< s, s >} s$$

A rescaled s, which is as close as possible to s-hat

$$e_{interf} = P_n \hat{s} = \frac{< \hat{s}, n >}{< n, n >} n$$

A rescaled n, which is as close as possible to s-hat

$$e_{artif} = \hat{s} - s_{target} - e_{interf} = \hat{s} - \frac{< \hat{s}, s >}{< s, s >} s - \frac{< \hat{s}, n >}{< n, n >} n$$

What remains of s-hat

Outline

- Basics
- **Tools**

Library: Demucs

https://github.com/facebookresearch/demucs

Model	Domain	Extra data?	Overall SDR
Wave-U-Net	waveform	no	3.2
Open-Unmix	spectrogram	no	5.3
Demucs (v2)	waveform	no	6.3
Band-Spit RNN	spectrogram	no	**8.2**
Hybrid Demucs (v3)	hybrid	no	7.7
MMDenseLSTM	spectrogram	804 songs	6.0
Spleeter	spectrogram	25k songs	5.9
HT Demucs f.t. (v4)	hybrid	800 songs	**9.0**

Audiostrip is providing free online separation with Demucs on their website https://audiostrip.co.uk/.

Neutone provides a realtime Demucs model in their free VST/AU plugin that can be used in your favorite DAW.

Other pre-trained models can be selected with the `-n` flag. The list of pre-trained models is:

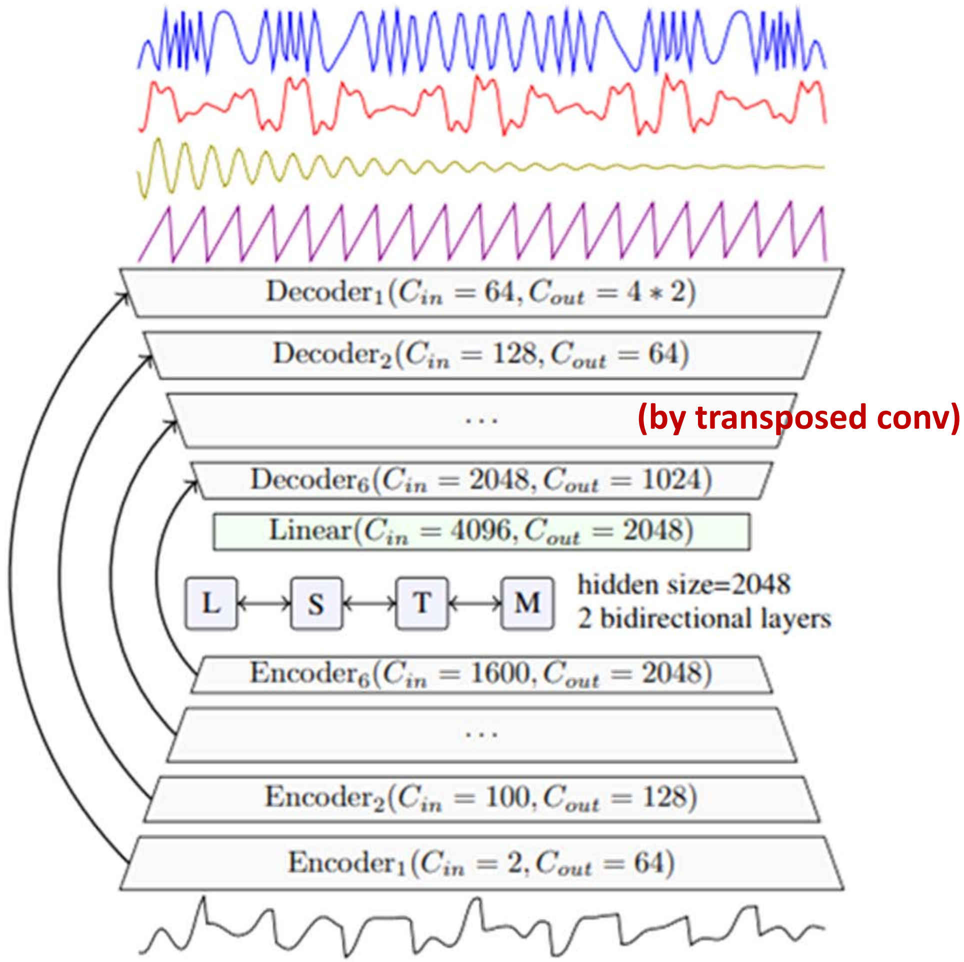
- `htdemucs_6s`: 6 sources version of `htdemucs`, with `piano` and `guitar` being added as sources. Note that the `piano` source is not working great at the moment.

Demucs (v2)

(waveform-domain loss)

- Several novel elements inspired by latest work on audio synthesis at that time

Difference	Valid set L1 loss	Test set SDR
no BiLSTM	0.171	5.40
no pitch/tempo aug.	0.156	5.80
no initial weight rescaling	0.156	5.88
no 1x1 convolutions in encoder	0.154	5.89
ReLU instead of GLU	0.157	5.92
no BiLSTM, depth=7	0.160	5.94
MSE loss	N/A	5.99
no resampling	0.153	6.03
no shift trick	N/A	6.05
kernel size of 1 in decoder convolutions	0.153	6.11
Reference	0.150	6.28

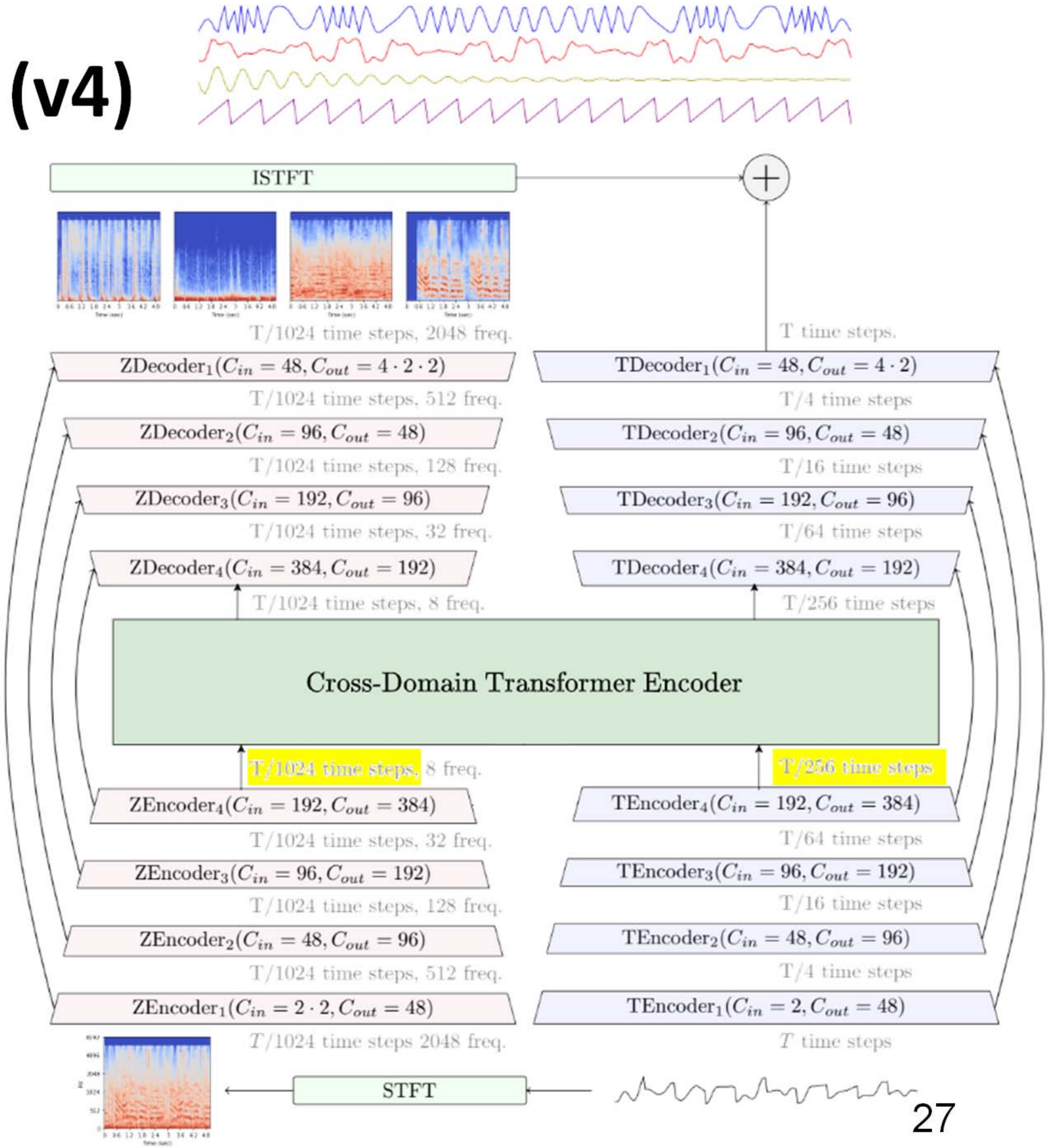


Ref: Défossez et al, "Music source separation in the waveform domain," arXiv 2019

Demucs (v4)

- Use **Transformer** to learn long range contextual information
  - "Innermost layers are replaced by a cross-domain Transformer Encoder, using self-attention within one domain, and cross-attention across domains."
- Need larger data to benefit from the Transformer
  - "Highly benefited from using extra training data"

Ref: Rouard et al, "Hybrid Transformers for music source separation," ICASSP 2023



Model Size vs Separation Quality

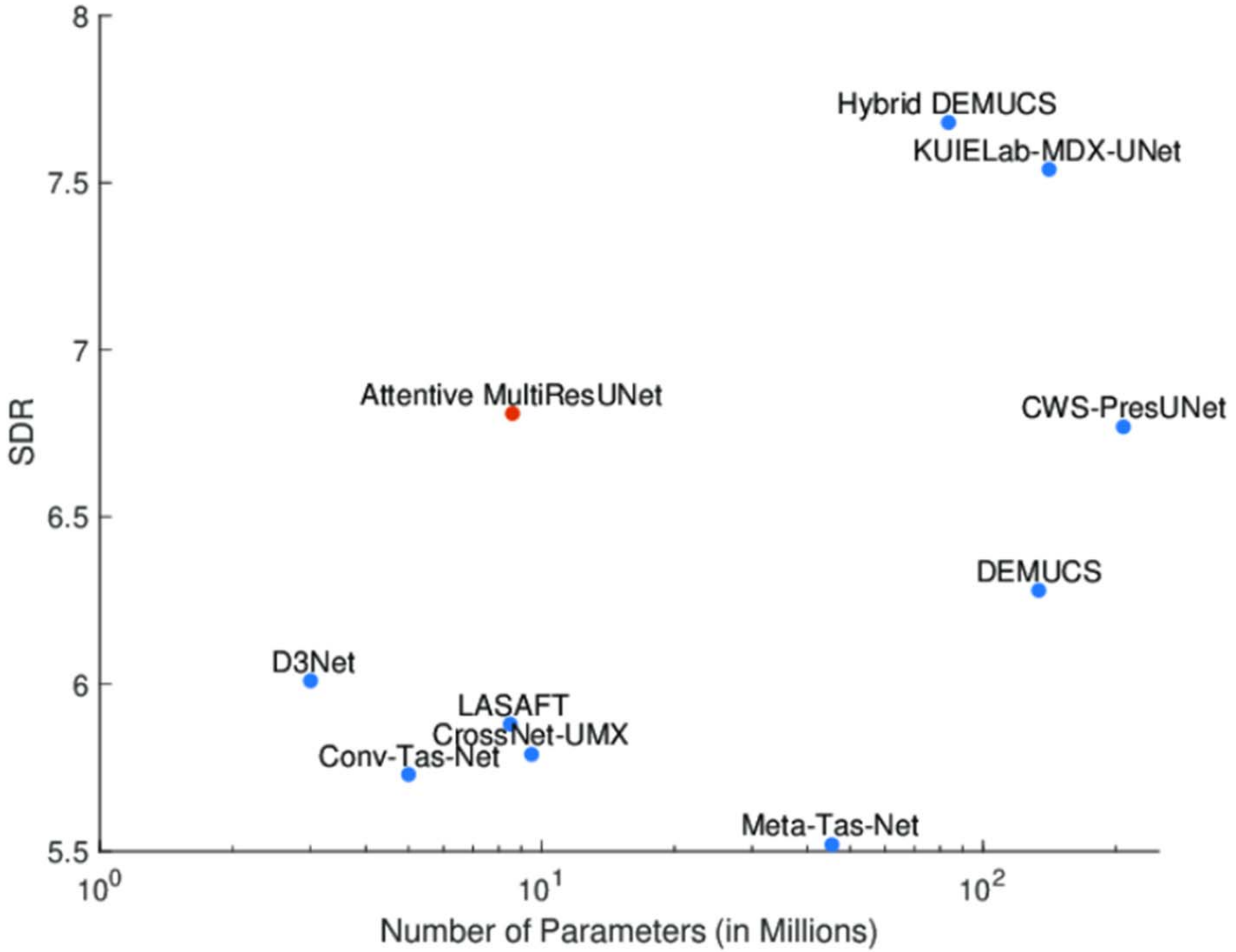


Figure from:
https://www.researchgate.net/figure/SDR-Performance-vs-Model-Parameters_fig2_365391217

Extra (Private) Datasets

Rank	Model	SDR (avg)	SDR (vocals)	SDR (drums)	SDR (bass)	SDR (other)	Extra Training Data	Paper	Code	Result	Year
1	**Sparse HT Demucs** (fine tuned)	9.20	9.37	10.83	10.47	6.41	✓	Hybrid Transformers for Music Source Separation			2022
2	**Hybrid Transformer Demucs** (f.t.)	9.00	9.20	10.08	9.78	6.42	✓	Hybrid Transformers for Music Source Separation			2022
3	**Band-Split RNN** (semi-sup.)	8.97	10.47	10.15	8.16	7.08	✓	Music Source Separation with Band-split RNN			2022
4	**TFC-TDF-UNet** (v3)	8.34	9.59	8.44	8.45	6.86	×	Sound Demixing Challenge 2023 Music Demixing Track Technical Report: TFC-TDF-UNet v3			2023
5	**Band-Split RNN**	8.23	10.21	8.58	7.51	6.62	×	Music Source Separation with Band-split RNN			2022
6	**Hybrid Demucs**	7.72	8.04	8.58	8.67	5.59	×	Hybrid Spectrogram and Waveform Source Separation			2021
7	**KUIELab-MDX-Net**	7.54	9.00	7.33	7.86	5.95	×	KUIELab-MDX-Net: A Two-Stream Neural Network for Music Demixing			2021

Model Checkpoints Collected by ZFTurbo

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

Available models for training:

- MDX23C based on KUIELab TFC TDF v3 architecture. Key: `mdx23c` .
- Demucs4HT [Paper]. Key: `htdemucs` .
- VitLarge23 based on Segmentation Models Pytorch. Key: `segm_models` .
- TorchSeg based on TorchSeg module. Key: `torchseg` .
- Band Split RoFormer [Paper, Repository] . Key: `bs_roformer` .
- Mel-Band RoFormer [Paper, Repository]. Key: `mel_band_roformer` .
- Swin Upernet [Paper] Key: `swin_upernet` .
- BandIt Plus [Paper, Repository] Key: `bandit` .
- SCNet [Paper, Official Repository, Unofficial Repository] Key: `scnet` .
- BandIt v2 [Paper, Repository] Key: `bandit_v2` .
- Apollo [Paper, Repository] Key: `apollo` .
- TS BSMamba2 [Paper, Repository] Key: `bs_mamba2` .
- Conformer [Paper, Repository] Key: `conformer` .
- SCNet Tran Key: `scnet_tran` .
- SCNet Masked Key: `scnet_masked` .

Inference example

```
python inference.py \
    --model_type mdx23c \
    --config_path configs/config_mdx23c_musdb18.yaml \
    --start_check_point results/last_mdx23c.ckpt \
    --input_folder input/wavs/ \
    --store_dir separation_results/
```

Ultimate Vocal Remover

https://github.com/Anjok07/ultimatevocalremovergui

- GUI & batch processing
- Can separate lead vocal from backing vocal harmonies