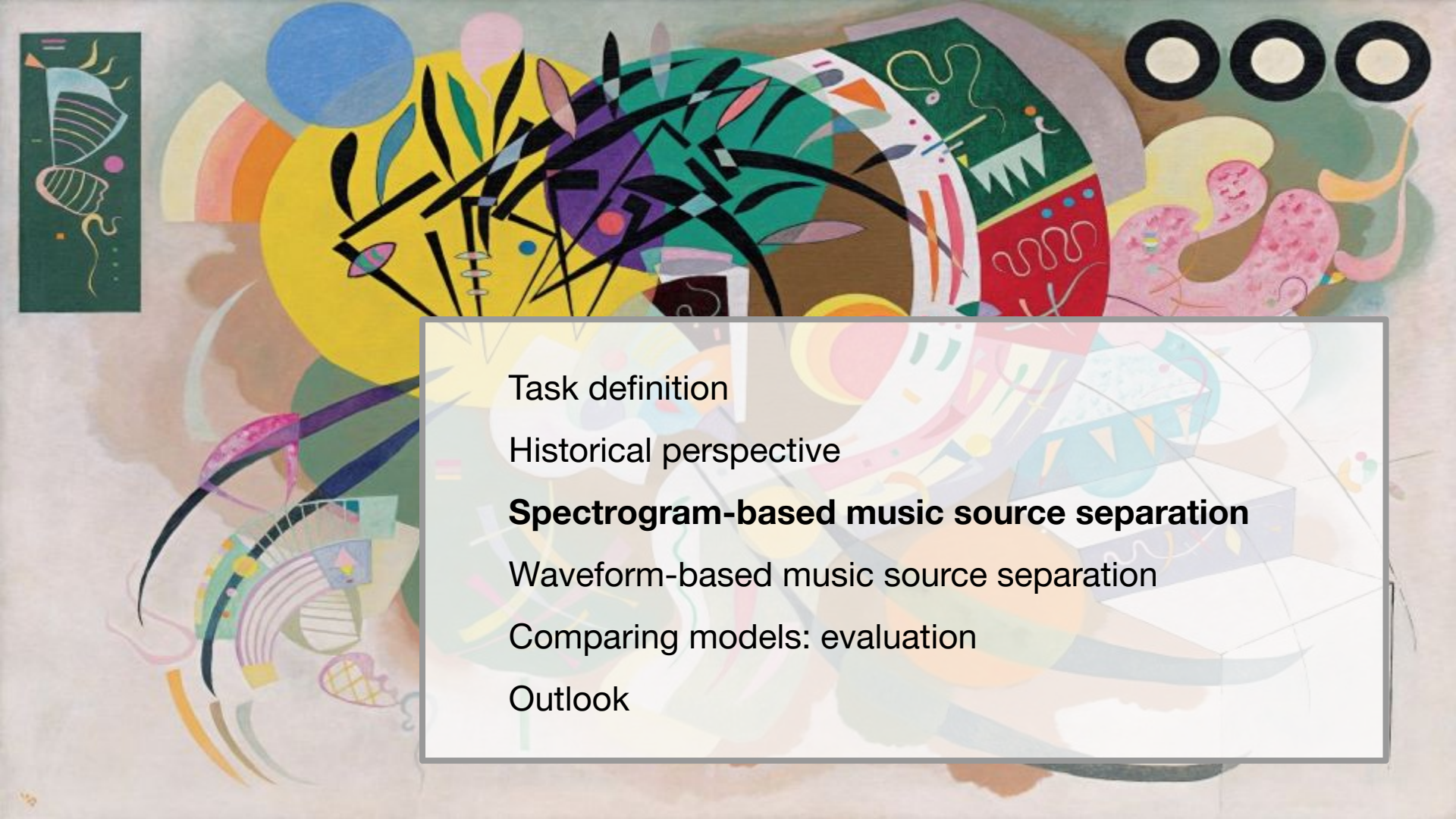




Music source separation with deep learning

Jordi Pons / www.jordipons.me / @jordiponsdotme



Task definition

Historical perspective

Spectrogram-based music source separation

Waveform-based music source separation

Comparing models: evaluation

Outlook



Task definition

Historical perspective

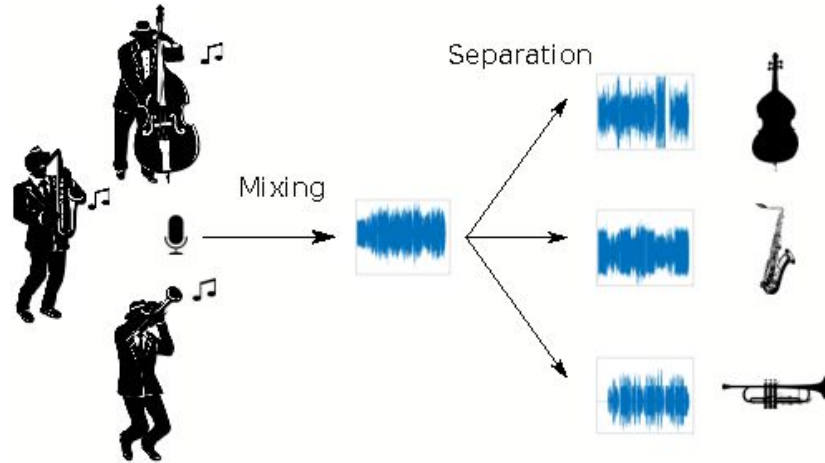
Spectrogram-based music source separation

Waveform-based music source separation


Comparing models: evaluation

Outlook

Task definition: Music Source Separation







Task definition

Historical perspective

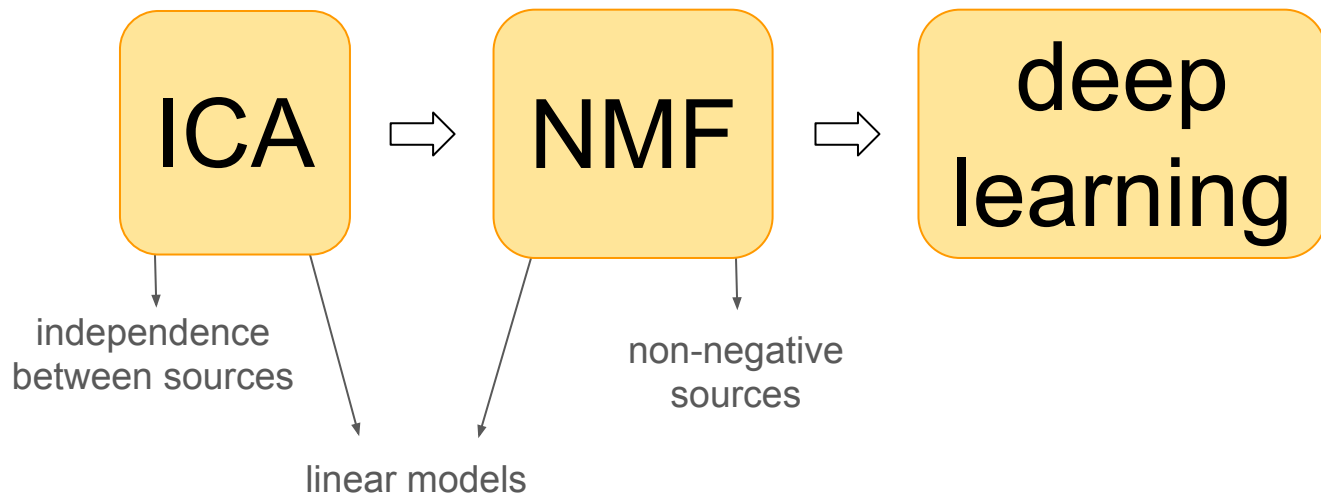
Spectrogram-based music source separation

Waveform-based music source separation

Comparing models: evaluation

Outlook

Historical perspective: unsupervised & linear models



Linear model example

linear approximation activations

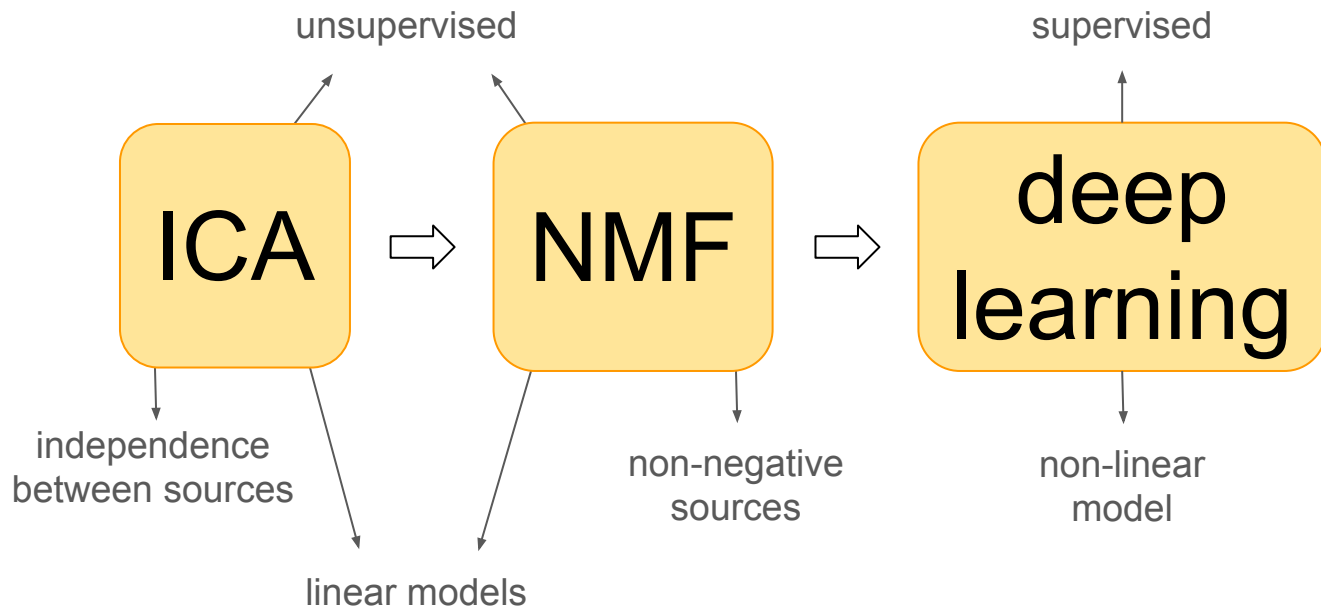
$$X \approx \hat{X} = \sum_k w_k h_k = WH$$

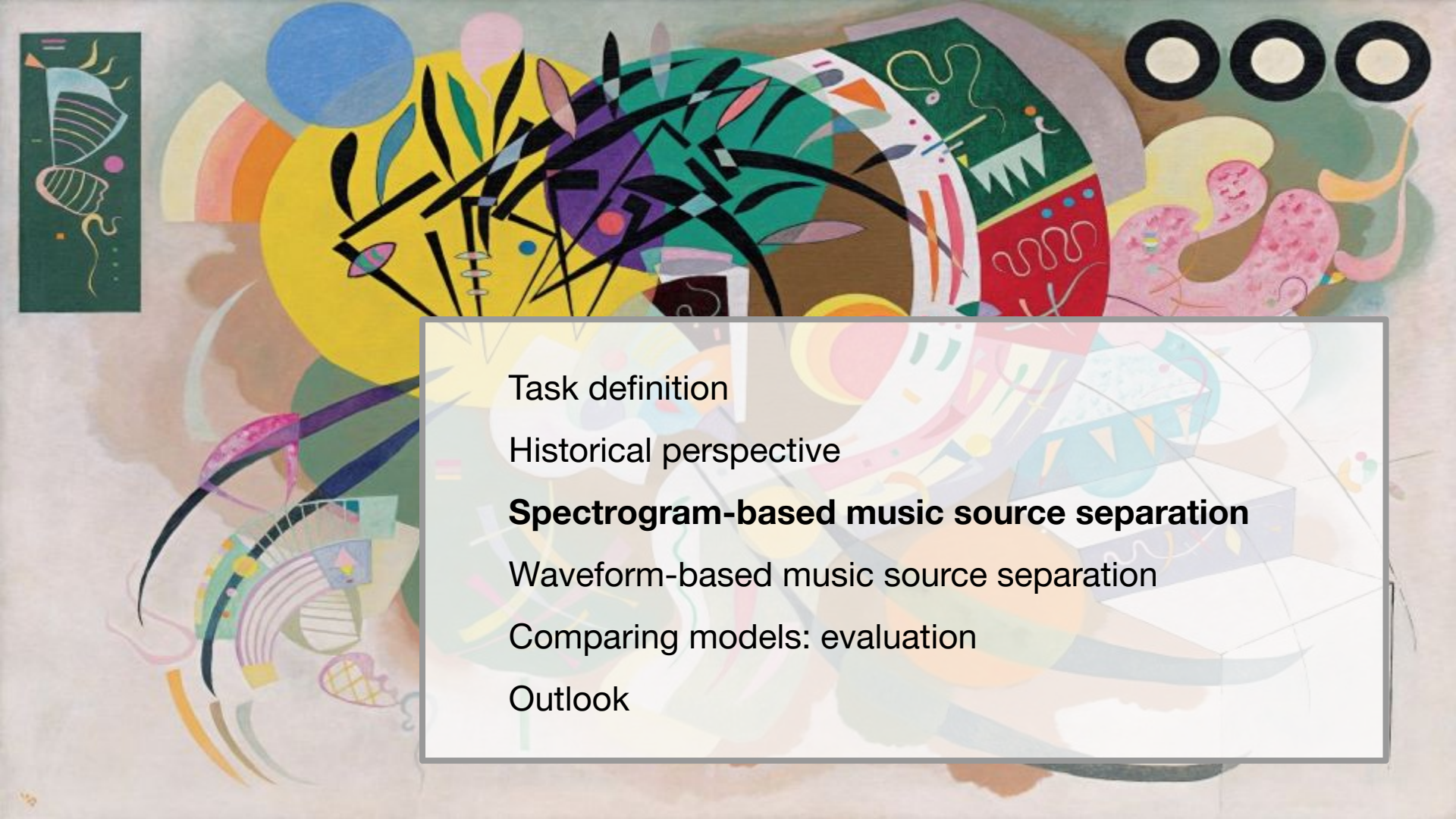
 bases

The diagram illustrates the linear model equation $X \approx \hat{X} = \sum_k w_k h_k = WH$. Annotations include: 'linear approximation' with a downward arrow pointing to \hat{X} ; 'activations' with a diagonal arrow pointing to h_k ; and 'bases' with an upward arrow pointing to w_k .

Unsupervised factorization of the mixture
into **bases** (w) and **activations** (h)

Historical perspective: unsupervised & linear models





Task definition

Historical perspective

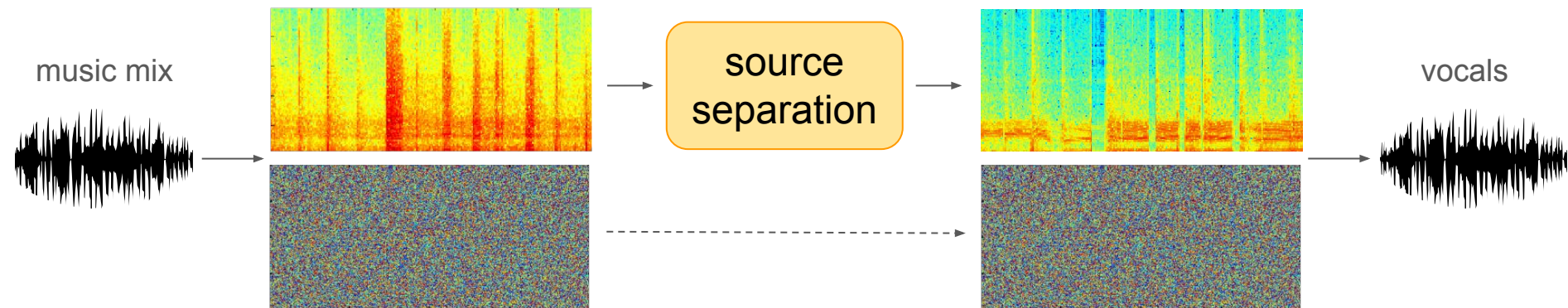
Spectrogram-based music source separation

Waveform-based music source separation

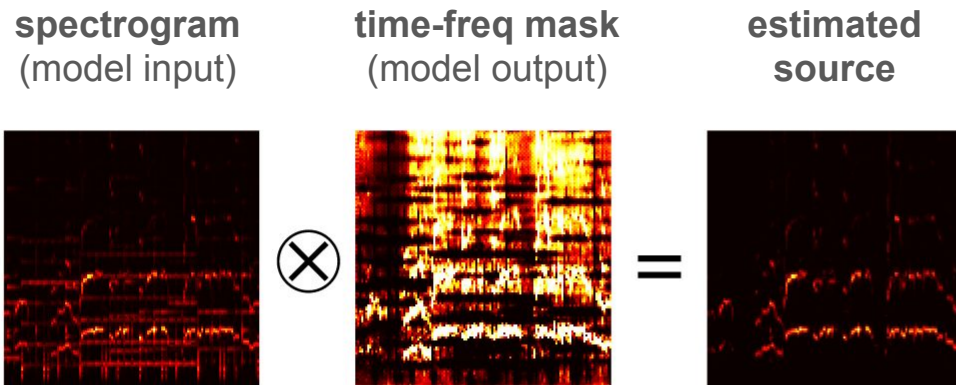
Comparing models: evaluation

Outlook

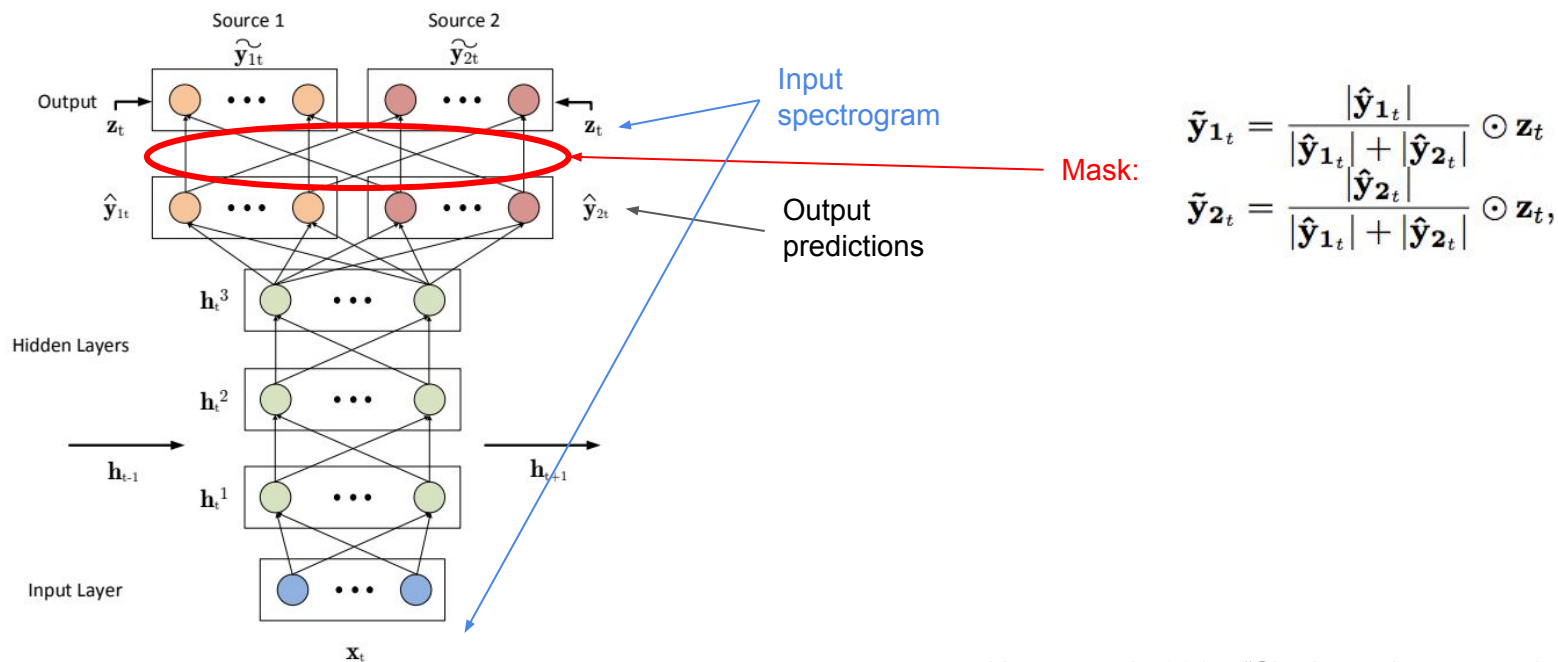
Spectrogram-based music source separation



Filtering spectrograms with masks

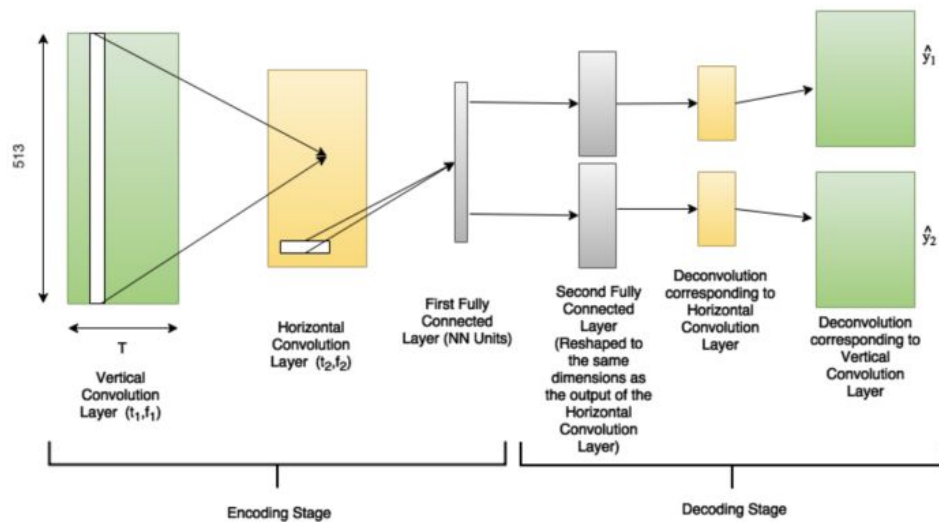


Deep recurrent neural networks

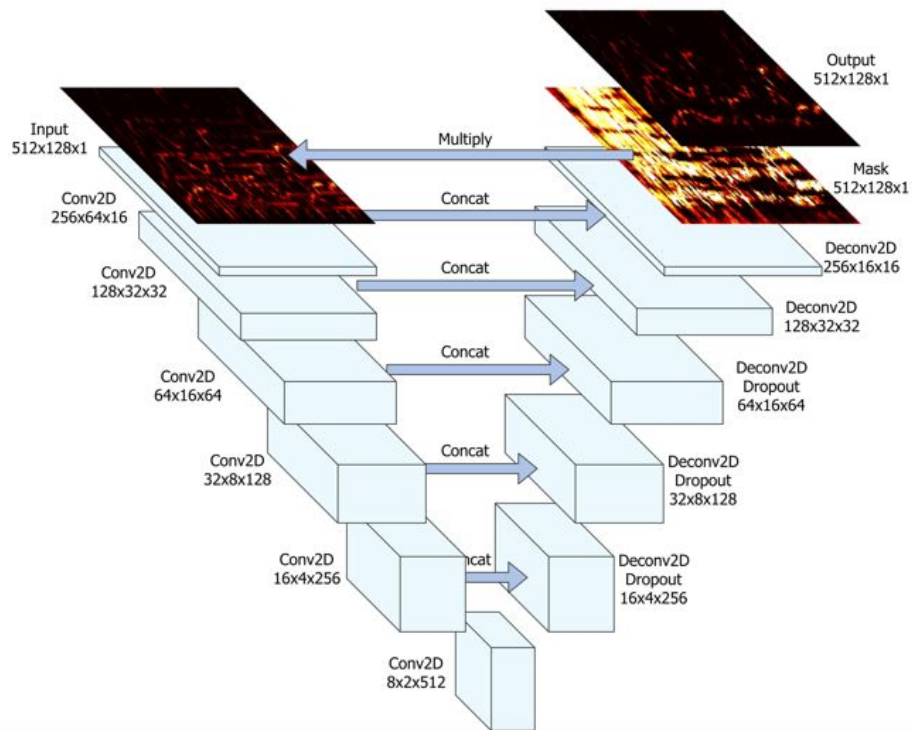


Huang et al., 2014. "Singing-voice separation from monaural recordings using deep recurrent neural networks" in ICASSP.

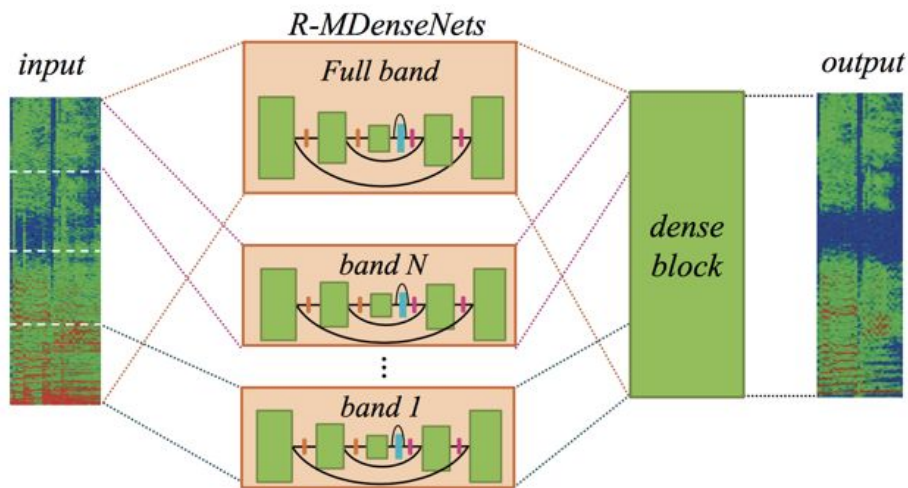
Convolutional auto-encoder



U-net auto-encoder

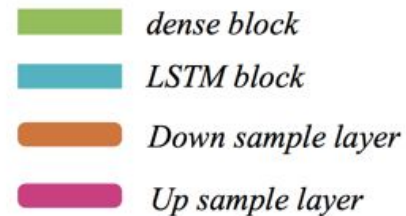


MMDenseLSTM

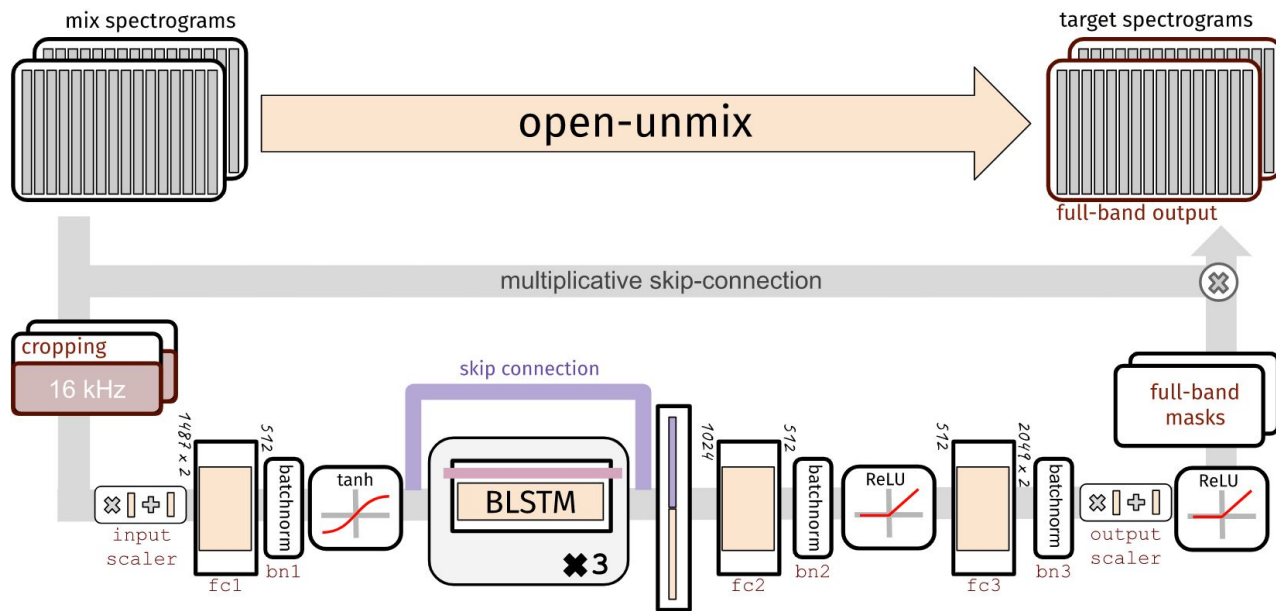


MM

- Multi-scale
- Multi-band

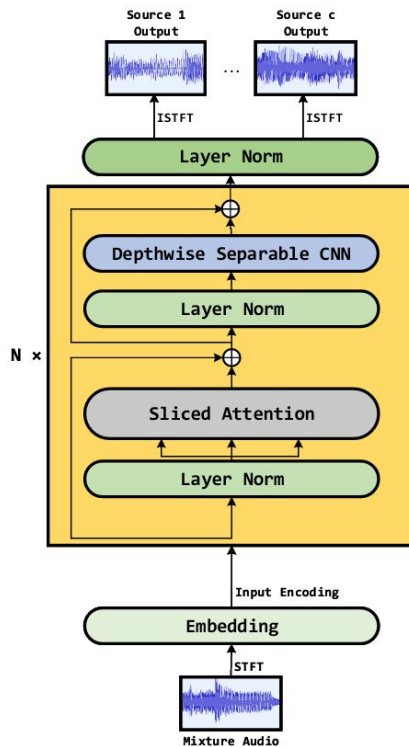


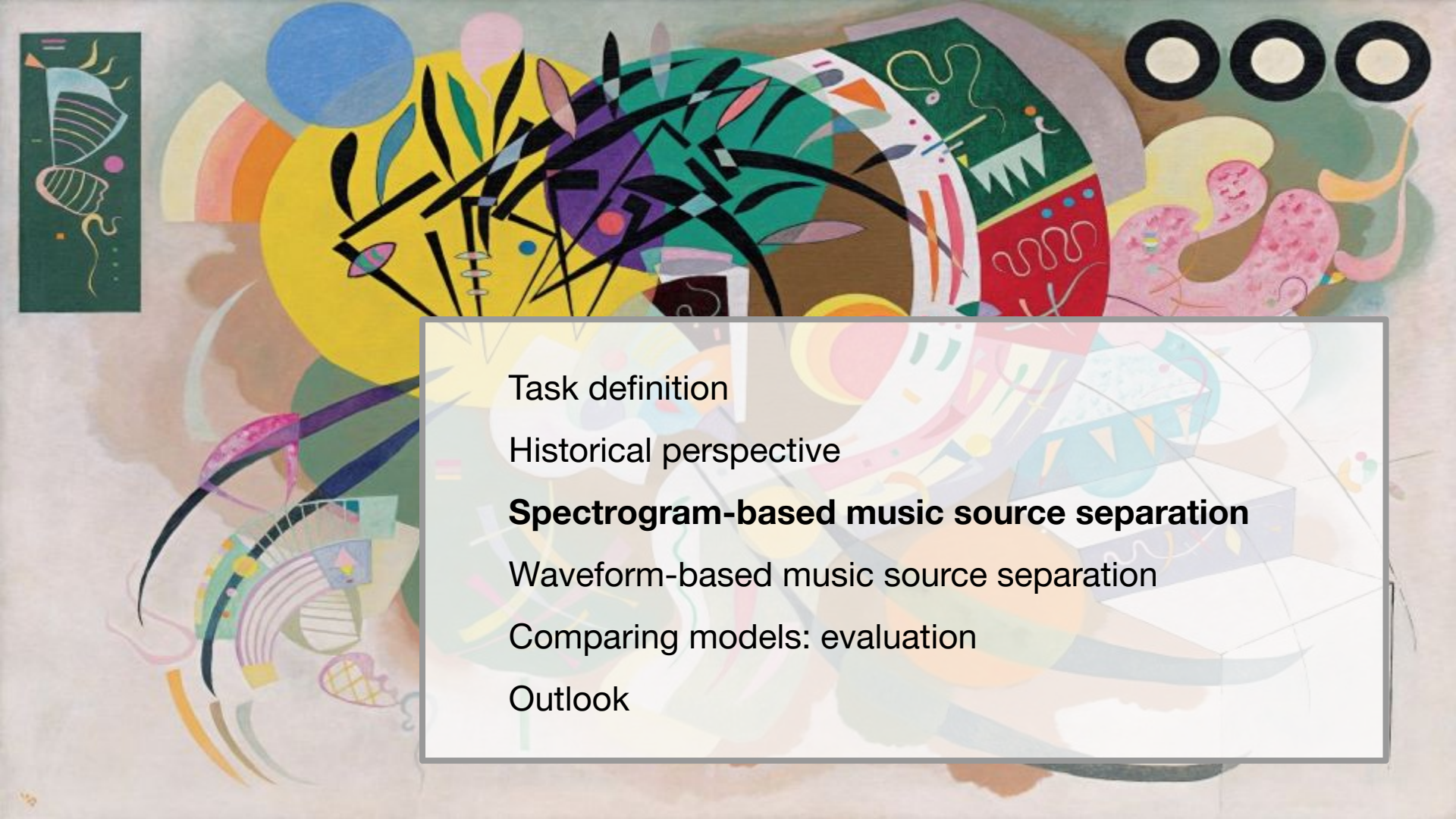
Open-unmix: a state-of-the-art implementation



<https://github.com/sigsep/open-unmix-pytorch>

Sams-Net: attention-based





Task definition

Historical perspective

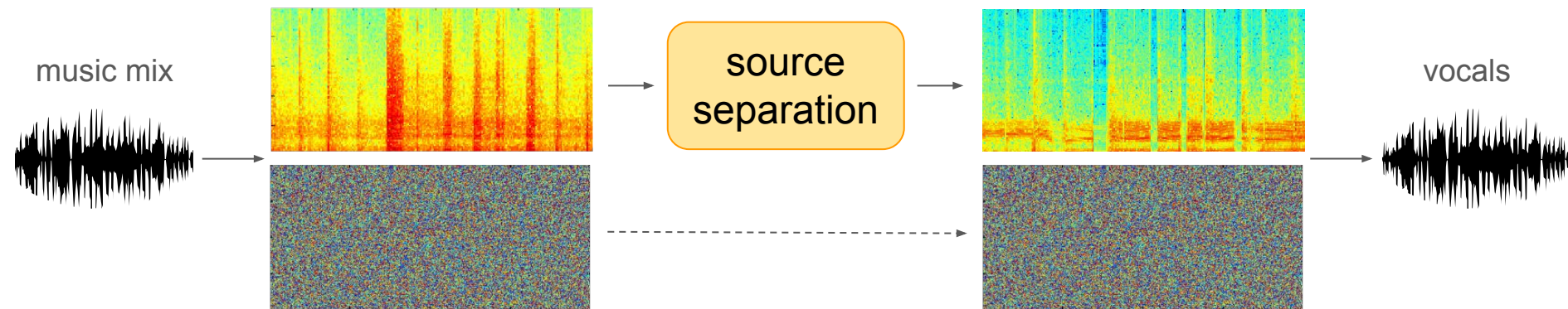
Spectrogram-based music source separation

Waveform-based music source separation

Comparing models: evaluation

Outlook

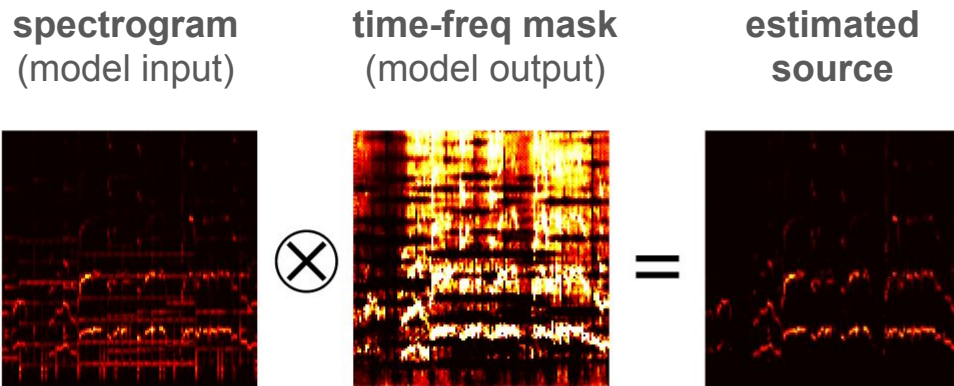
Why end-to-end music source separation?



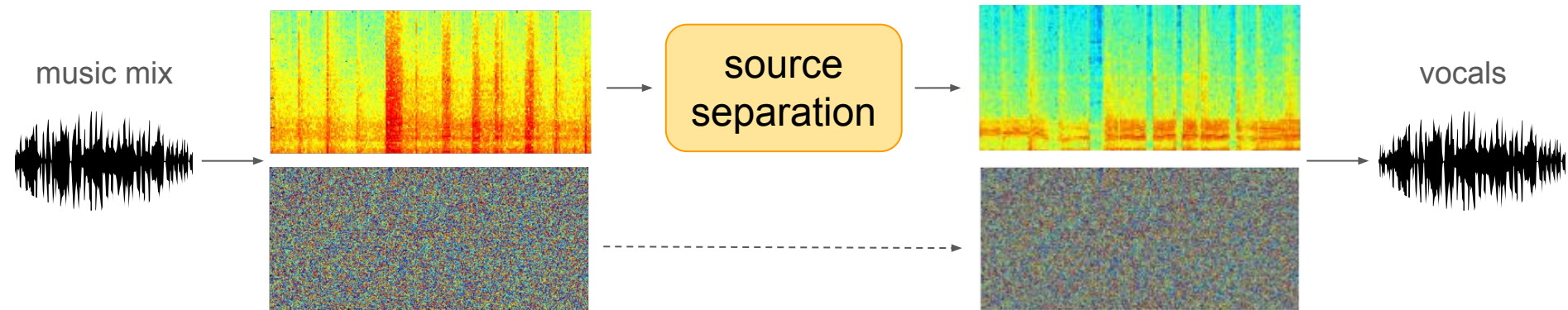
I) Are we missing crucial information when **discarding the phase**?

II) When using the **phase of the mixture at synthesis time**, are we introducing artifacts that are limiting our model's performance?

Why filtering spectrograms with masks?



III) It's **challenging to separate masked signals** (“perceptually” hidden sounds) via **filtering spectrograms**



- I) Are we missing crucial information when **discarding the phase**?
- II) When using the **phase of the mixture at synthesis time**, are we introducing artifacts that are limiting our model's performance?
- III) Is **challenging to separate masked signals** via **filtering spectrograms**

End-to-end music source separation

music mix



deep
learning



vocals



Other (active) research directions:

Use the complex STFT as i/o interface?

Kameoka et al., 2009. “ComplexNMF: A new sparse representation for acoustic signals” in ICASSP.

Dubey et al., 2017. “Does phase matter for monaural source separation?” in arXiv.

Le Roux et al., 2019. “Phasebook and friends: Leveraging discrete representations for source separation” in IEEE Journal of Selected Topics in Signal Processing.

Tan et al., 2019. “Complex Spectral Mapping with a CRNN for Monaural Speech Enhancement” in ICASSP.

Liu et al., 2019. “Supervised Speech Enhancement with Real Spectrum Approximation” in ICASSP.

Other (active) research directions:

Alternative models at synthesis time?

Virtanen and Klapuri, 2000. “Separation of harmonic sound sources using **sinusoidal modeling**,” in ICASSP.

Chandna et al., 2019. “A **vocoder** based method for singing voice extraction” in ICASSP.

End-to-end music source separation

music mix



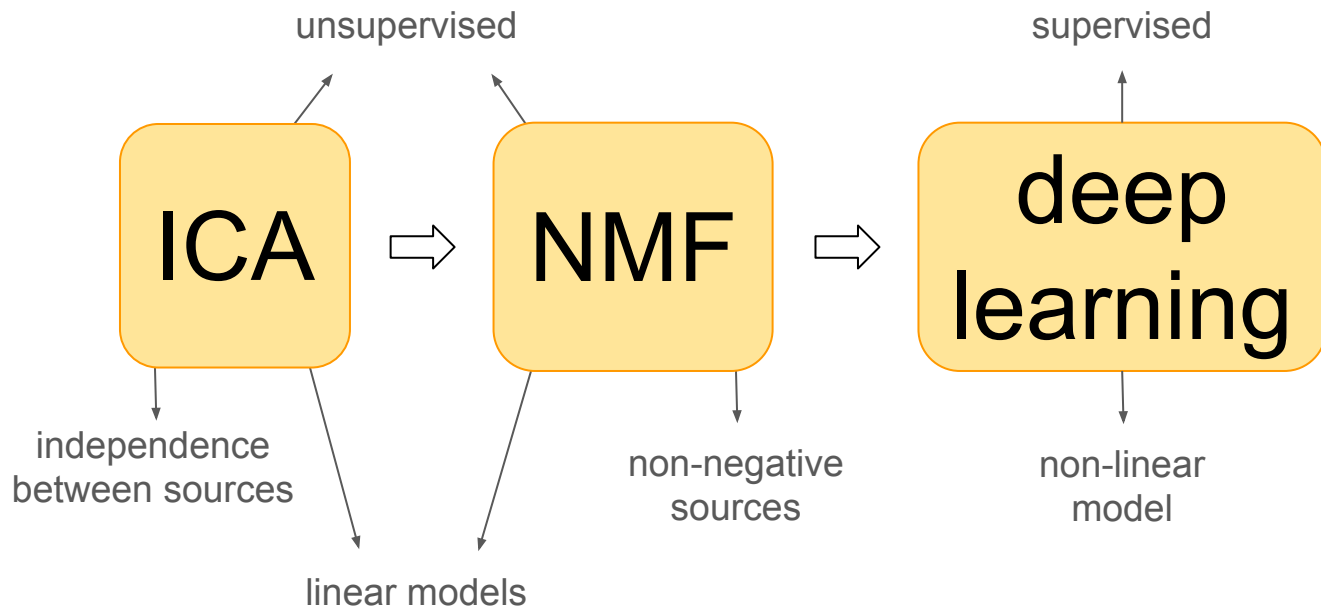
deep
learning



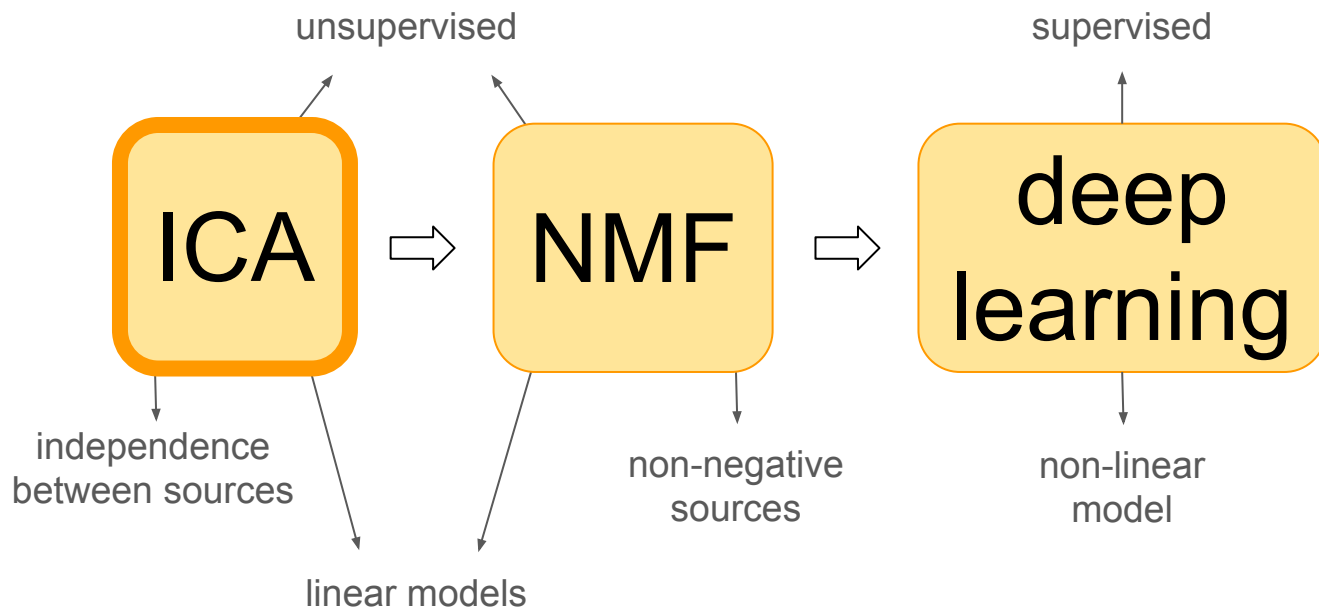
vocals



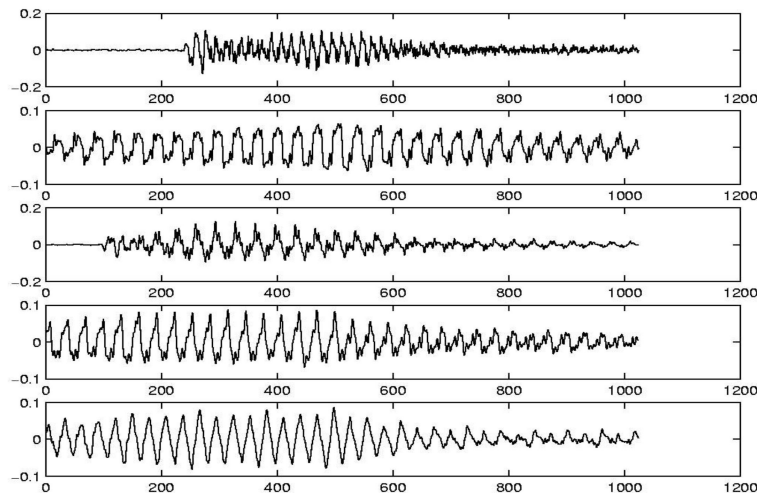
Historical perspective: waveform-based models?



Historical perspective: waveform-based models?



waveform-based ICA

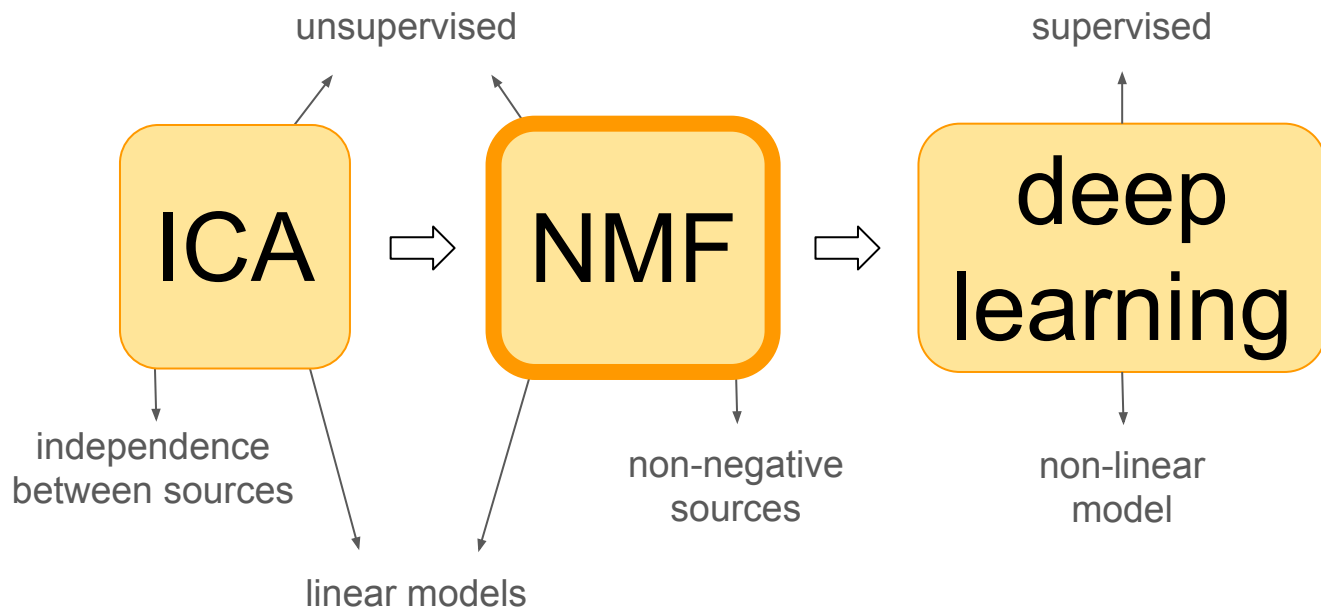


$$X \approx \hat{X} = \sum_k w_k h_k = WH$$

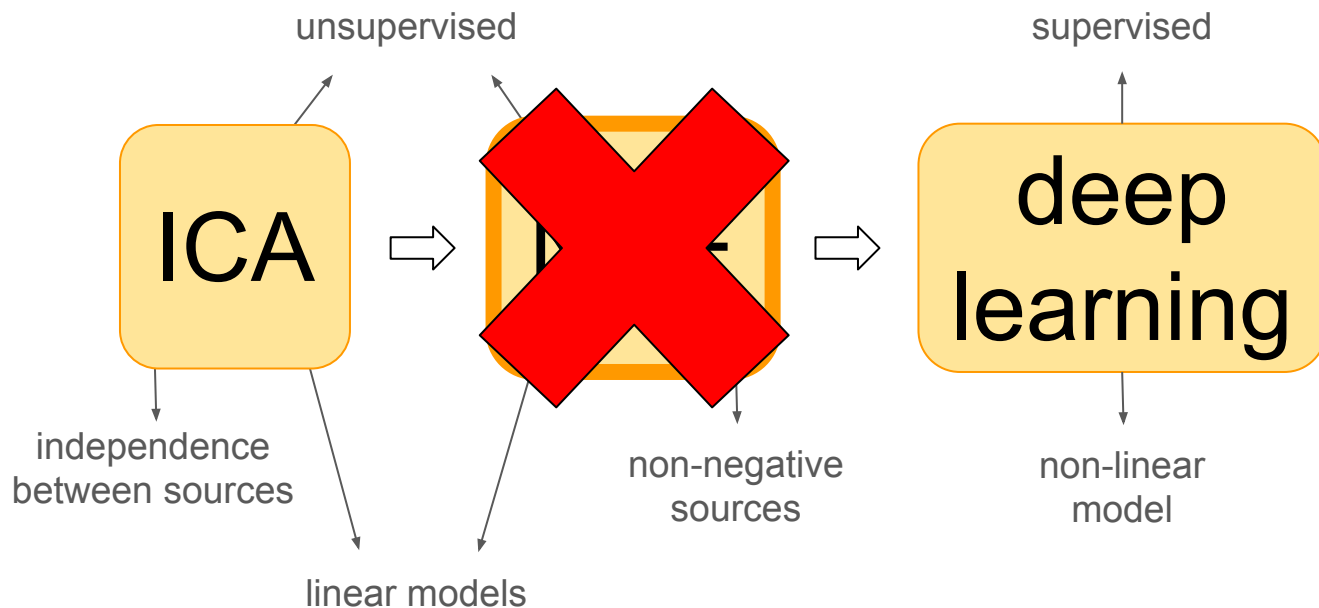
bases
activations

Problem 1: phase sensitive basis
Problem 2: simplicity of the linear model

Historical perspective: waveform-based models?

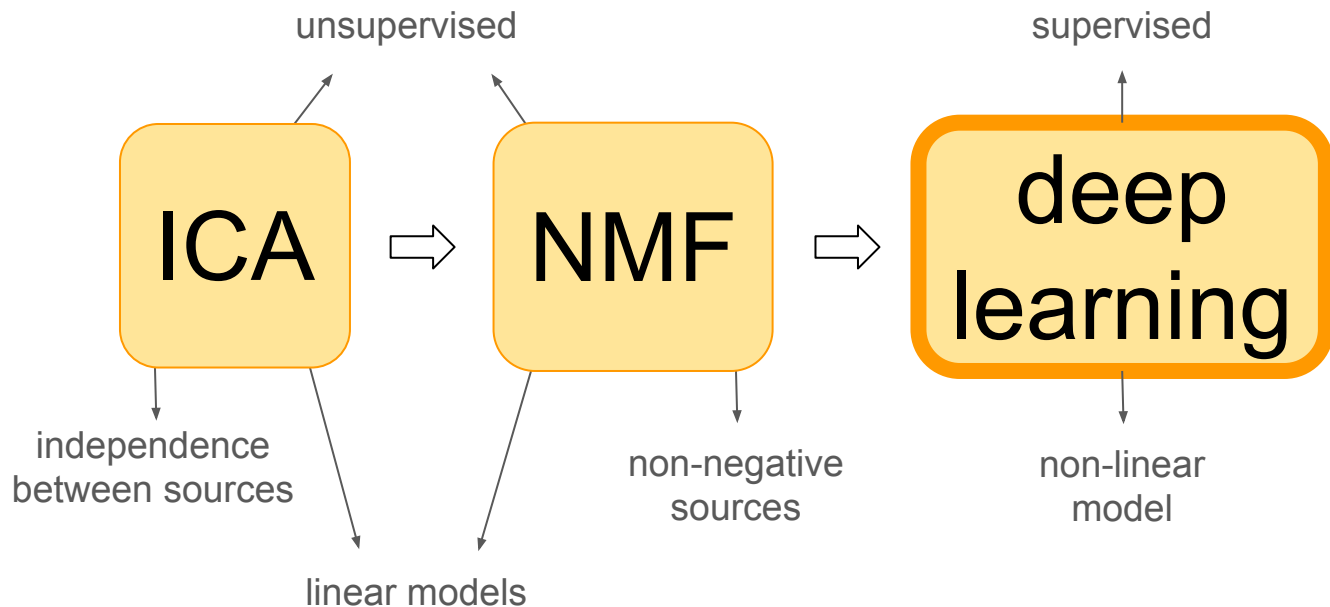


Historical perspective: waveform-based models?



NMF cannot be used with waveforms due to its non-negative constraint! (waveforms range from -1 to 1)

Historical perspective: waveform-based models?



A widely-used set of tools:

filtering spectrograms

linear models

unsupervised learning

audio domain knowledge

..maybe we could try another toolset?

~~filtering~~ → synthesis?

~~linear models~~ → non-linear models?

~~unsupervised learning~~ → supervised learning?

~~audio domain knowledge~~ → data driven?

End-to-end music source separation: 12 publications

Stoller et al., 2018. "Wave-u-net: A multi-scale neural network for end-to-end audio source separation" in arXiv.

Grais et al., 2018. "Raw Multi-Channel Audio Source Separation using Multi-Resolution Convolutional Auto-Encoders" in EUSIPCO.

Lluis, et al., 2018. "End-to-end music source separation: is it possible in the waveform domain?" in arXiv.

Slizovskaia et al., 2018. "End-to-end Sound Source Separation Conditioned on Instrument Labels" in arXiv.

Cohen-Hadria et al., 2019. "Improving singing voice separation using Deep U-Net and Wave-U-Net with data augmentation" in arXiv.

Kaspersen, 2019. "HydraNet: A Network For Singing Voice Separation". Master Thesis.

Akhmetov et al., 2019. "Time Domain Source Separation with Spectral Penalties". Technical Report.

Défossez et al., 2019. "Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed" in arXiv.

Narayanaswamy et al., 2019. "Audio Source Separation via Multi-Scale Learning with Dilated Dense U-Nets" in arXiv.

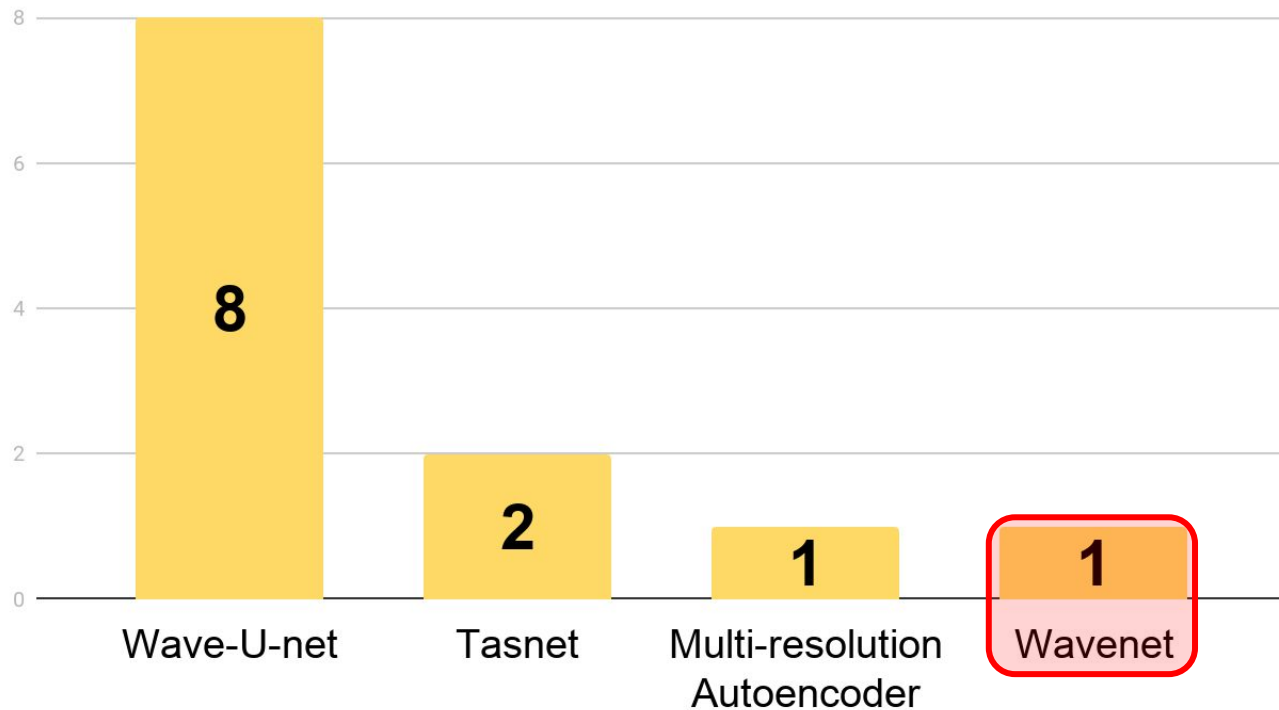
Défossez et al., 2019. "Music Source Separation in the Waveform Domain" in arXiv.

Samuel et al., 2019. "Meta-learning Extractors for Music Source Separation" in ICASSP.

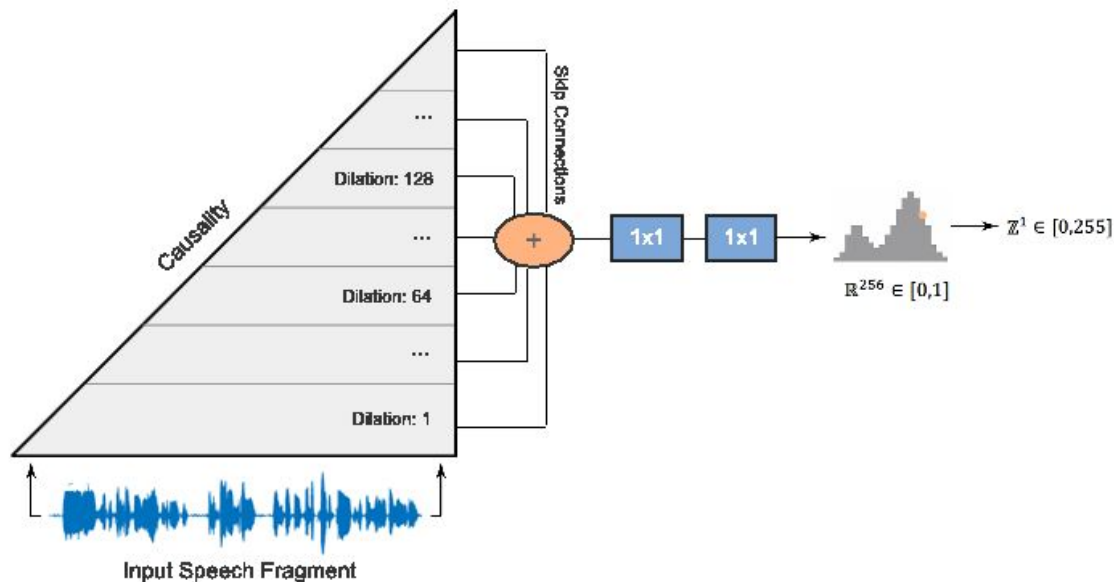
Nakamura et al., 2020. "Time-domain audio source separation based on wave-u-net combined with discrete wavelet transform" in ICASSP.

ALL THE PUBLICATIONS (WE ARE AWARE OF) IN CHRONOLOGICAL ORDER AS OF FEBRUARY 2020

End-to-end music source separation: architectures



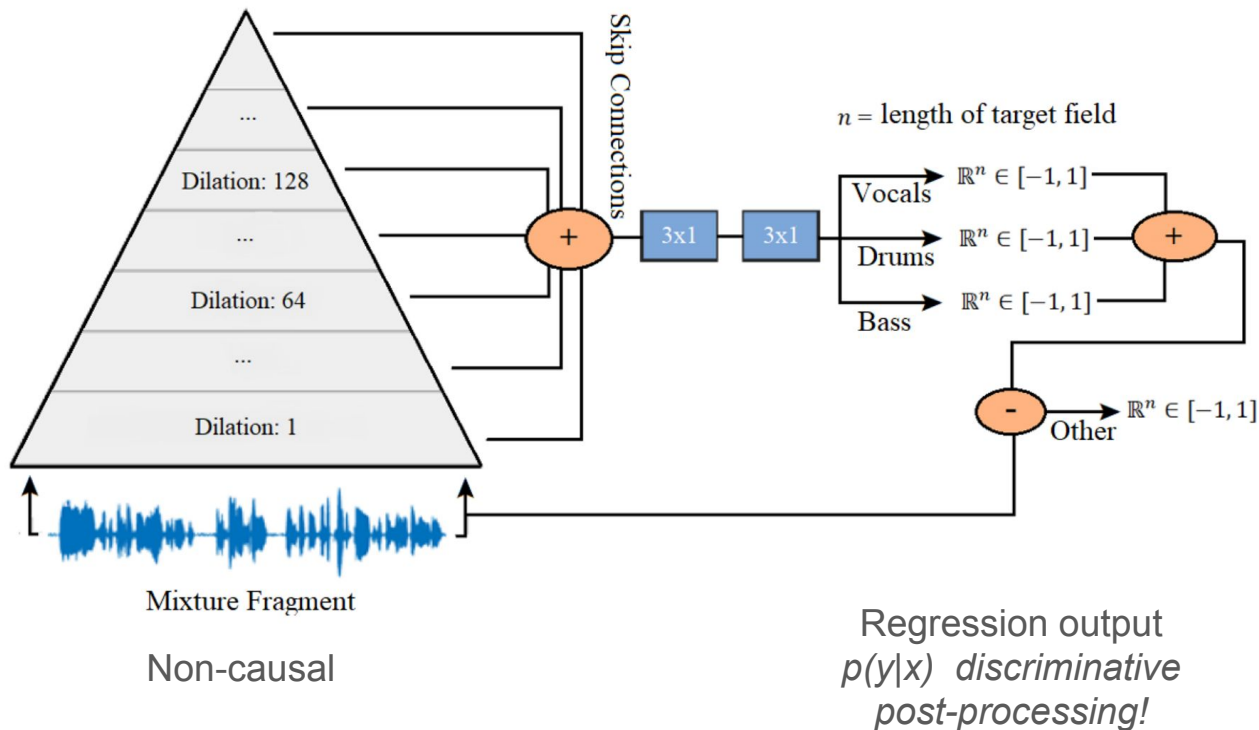
Introduction: the “generative” Wavenet



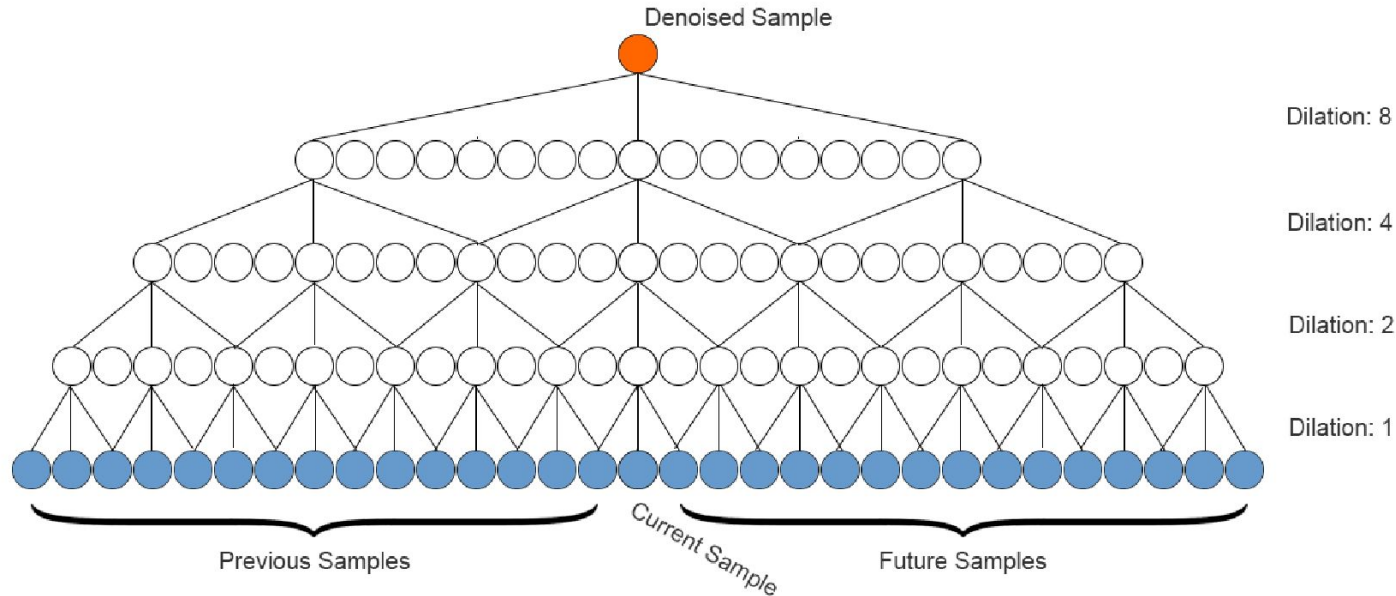
Causal

Softmax-output
distribution
modeling $p(x)$

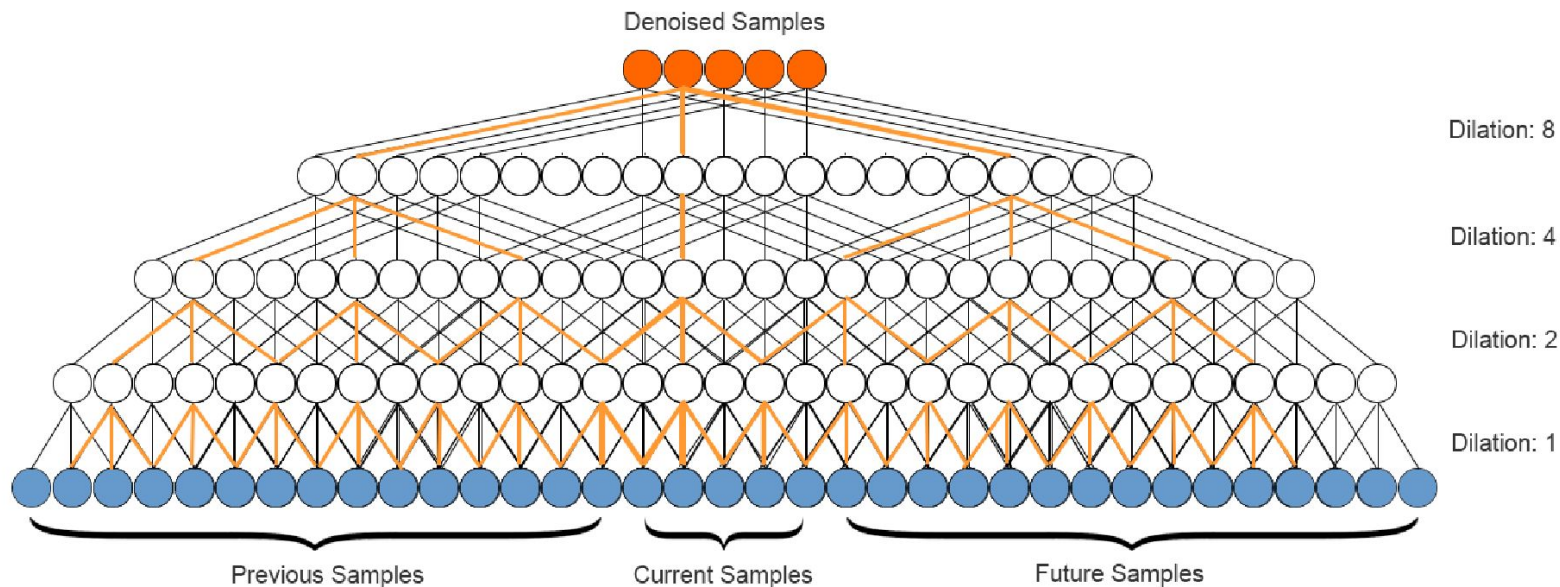
A “regression” Wavenet for music source separation



Fully convolutional & deterministic



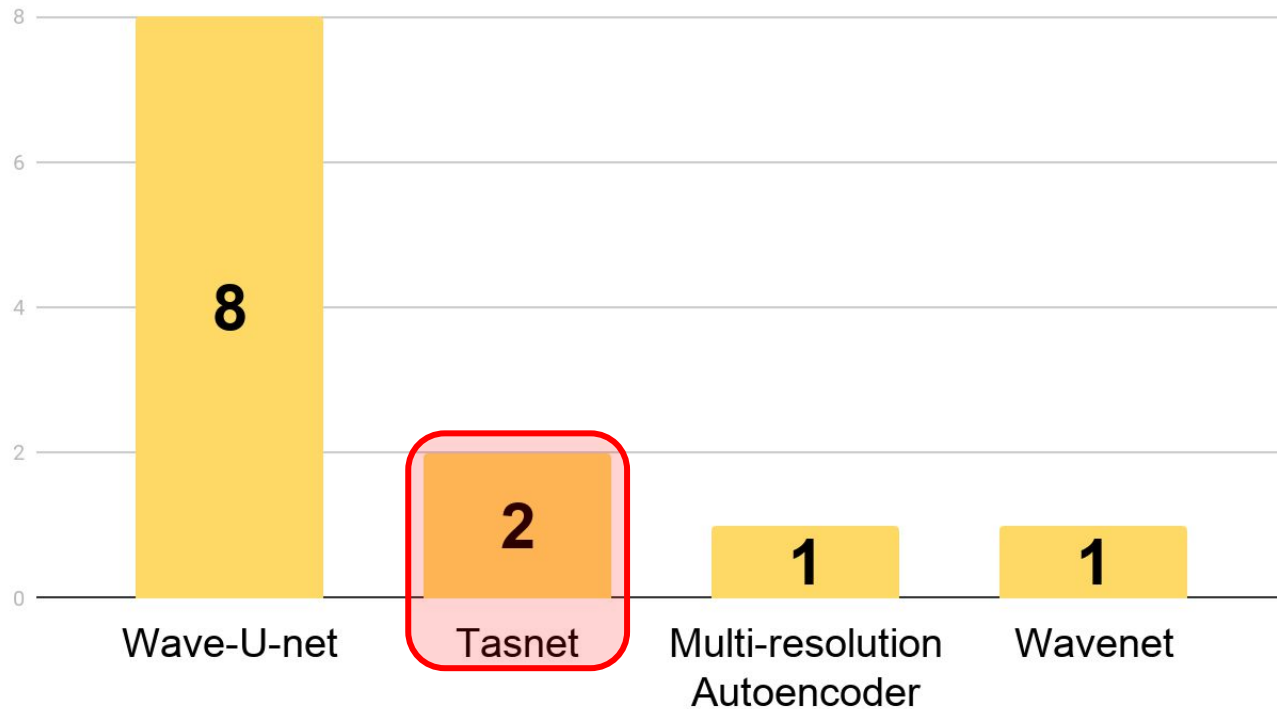
Fully convolutional & deterministic



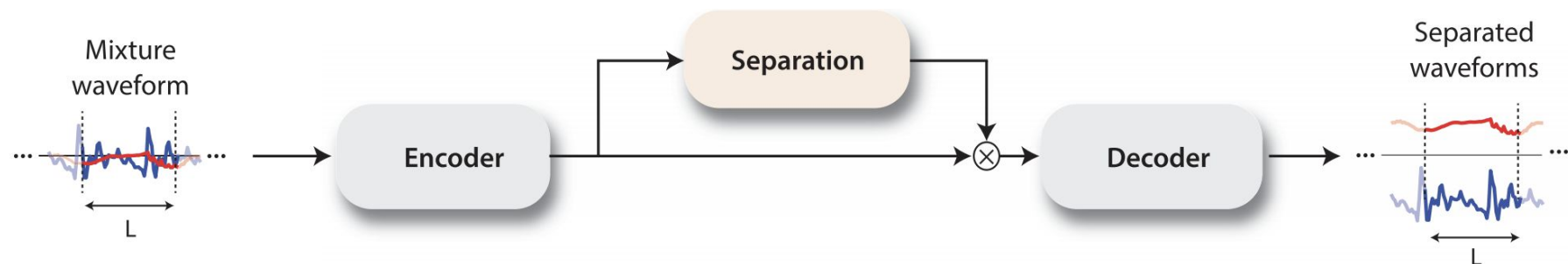
Real time inference!

1601 samples input $\rightarrow \approx 0.56$ sec per second of music on GPU!

End-to-end music source separation: architectures

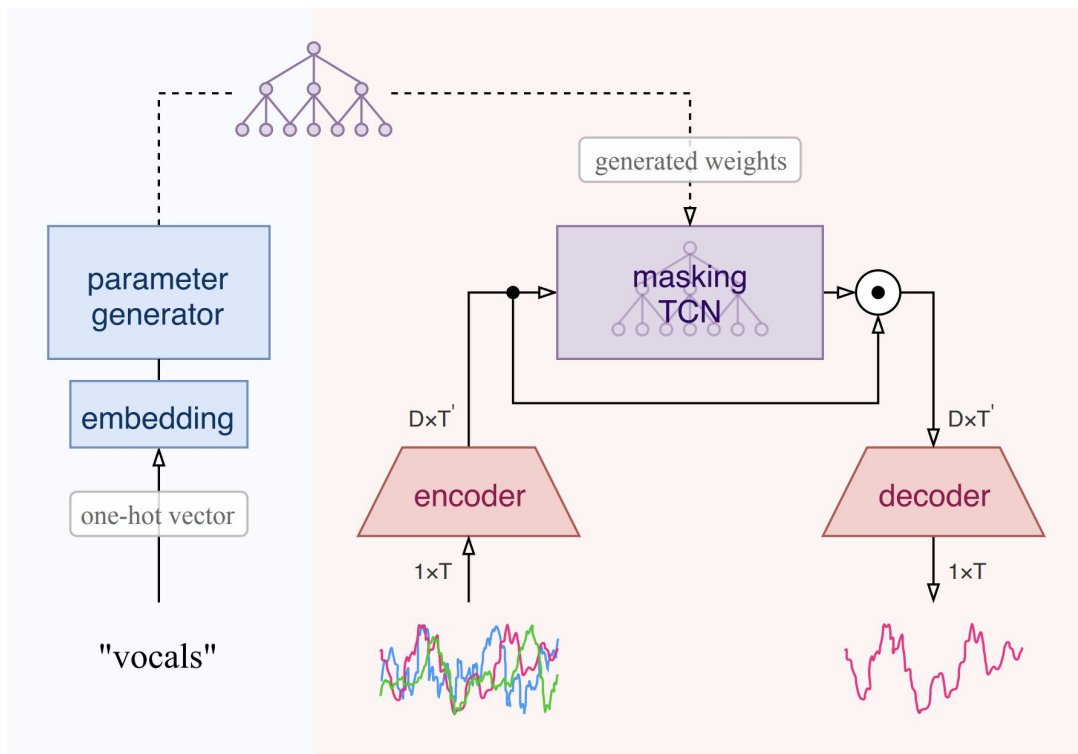


TasNet: encoder + separator + decoder

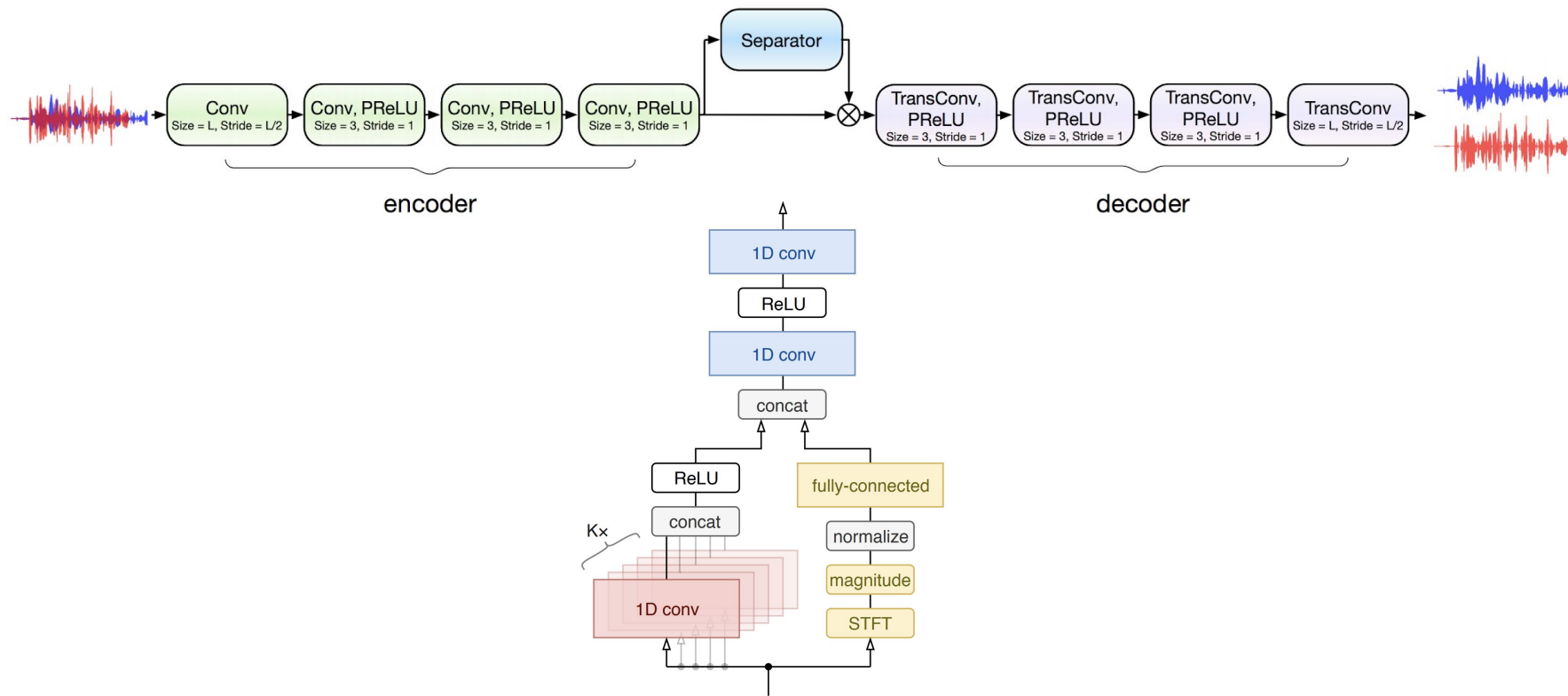


Défossez, et al., 2019. "Music source separation in the waveform domain" in arxiv.
Luo, et al. 2018. "Tasnet: time-domain audio separation network for real-time, single-channel speech separation" in ICASSP.

Separator: meta-learning with TasNet

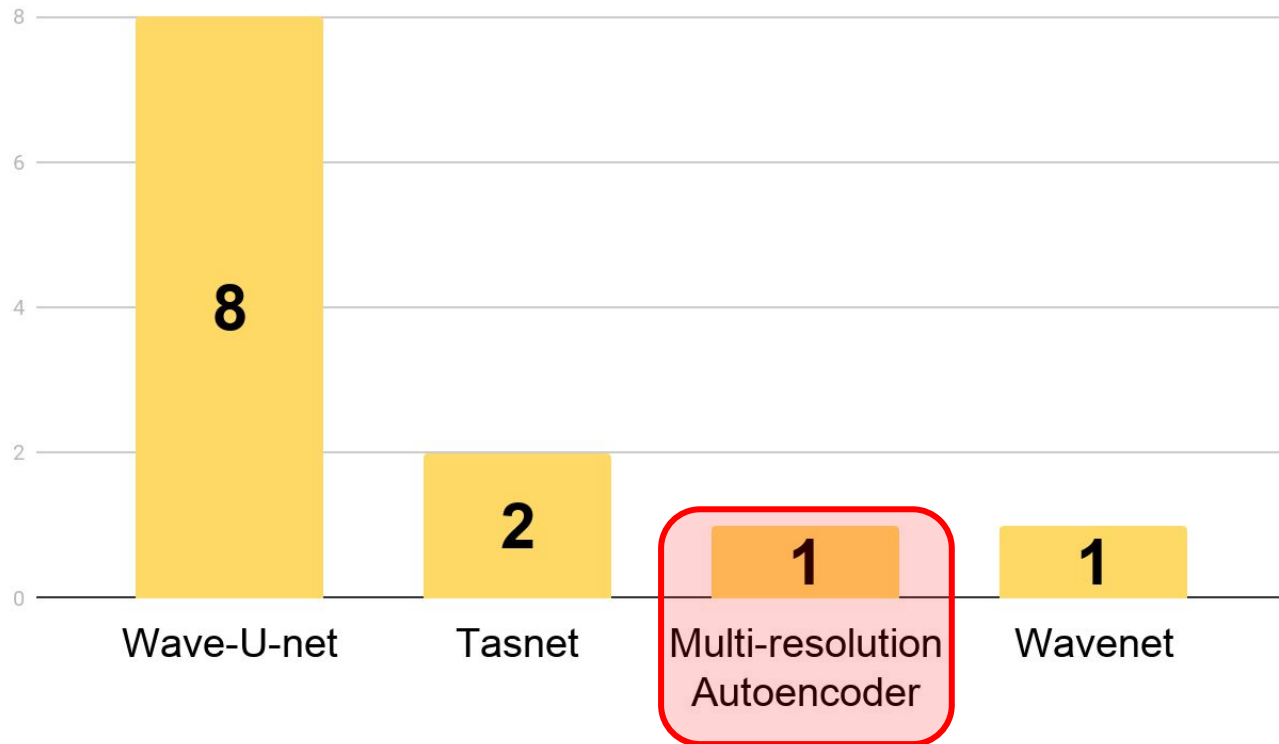


Encoders and Decoders

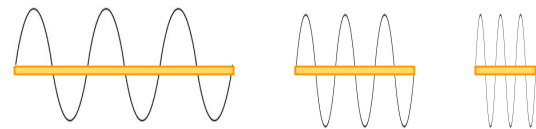
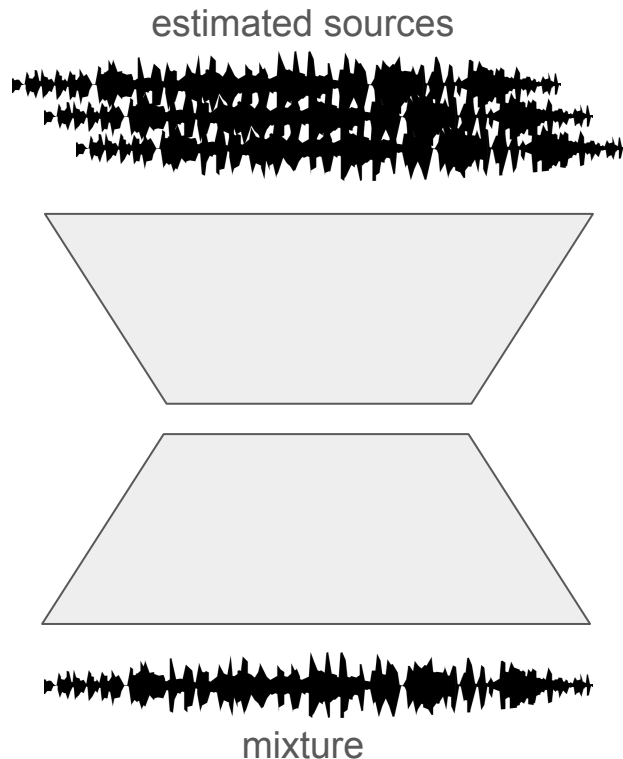


Samuel et al., 2020. “Meta-learning Extractors for Music Source Separation” in ICASSP.
Kadioglu et al., 2020. “An empirical study of Conv-TasNet” in ICASSP.

End-to-end music source separation: architectures



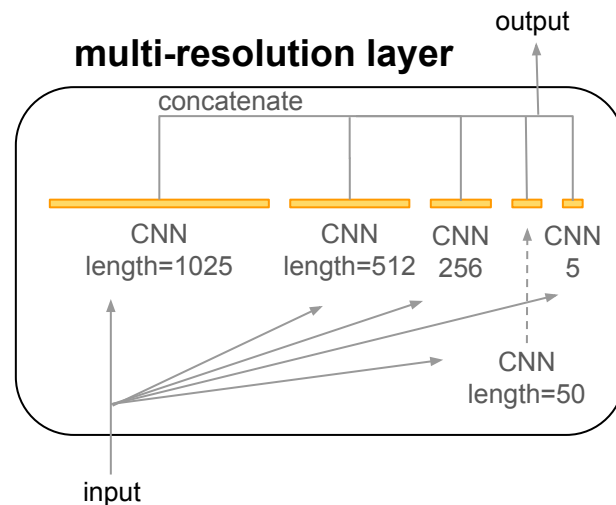
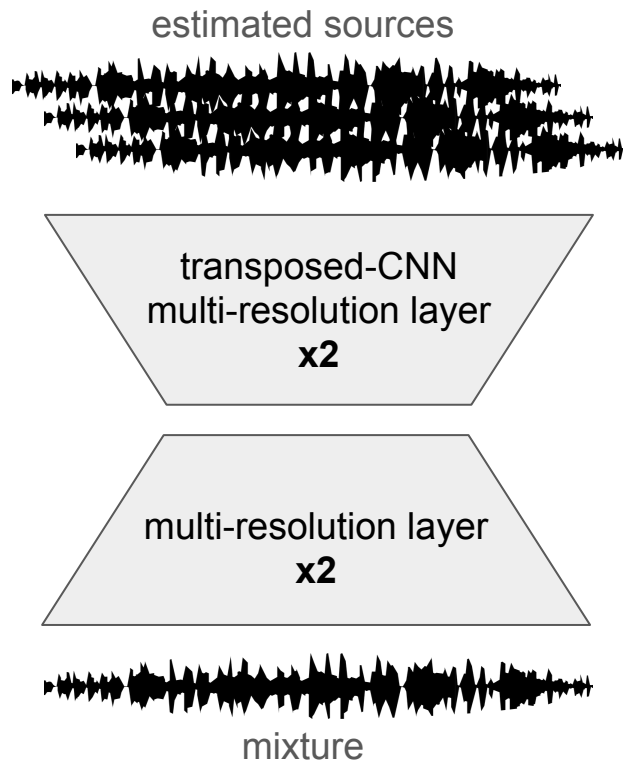
Multi-resolution & Convolutional autoencoder



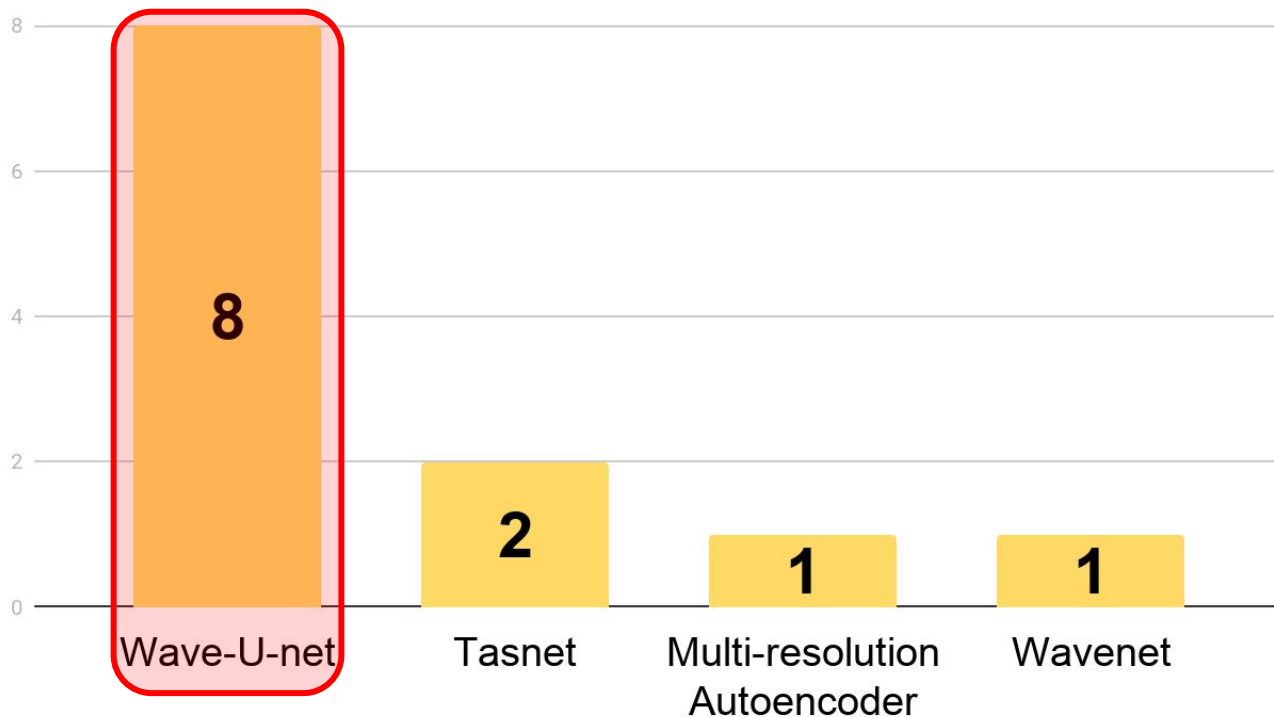
**Multi-resolution CNN: efficient way
to represent 3 periods!**

Multi-resolution CNN = Inception CNN
(different filter shapes in
the same CNN layer)

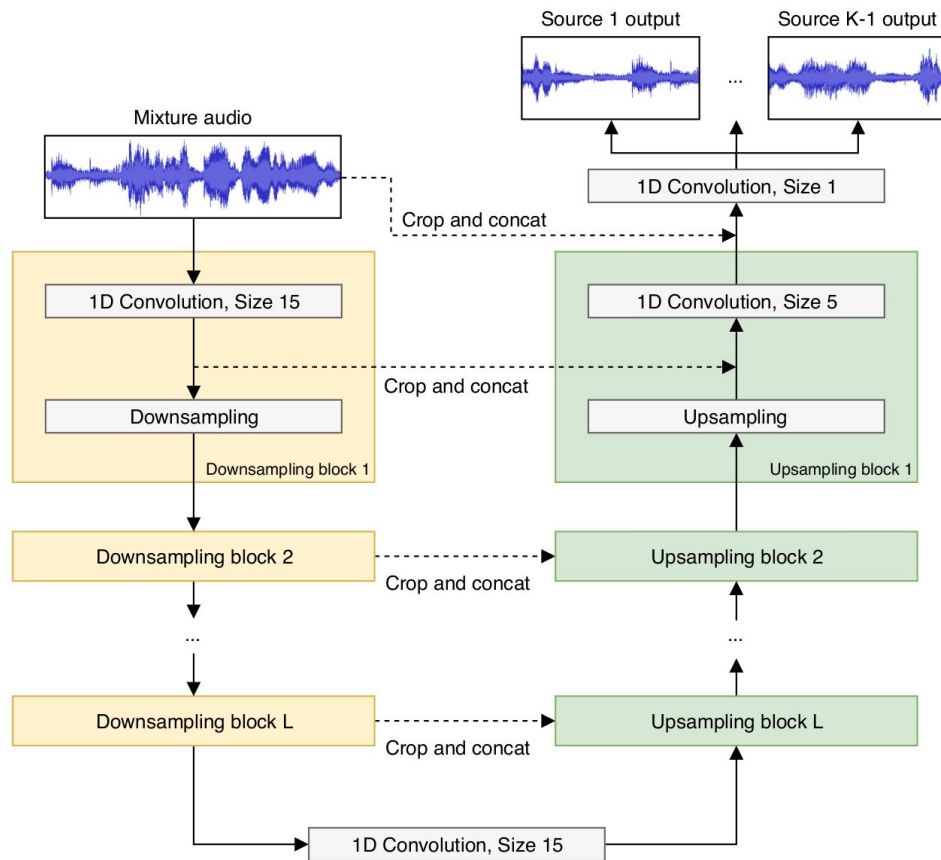
Multi-resolution & Convolutional autoencoder



End-to-end music source separation: architectures



Wave-U-net



Wave-u-net extensions

Wave-u-net extensions

- **Multiplicative conditioning using instrument labels at the bottleneck.**

Slizovskaia et al., 2019. “End-to-end Sound Source Separation Conditioned on Instrument Labels” in ICASSP.

- **Data augmentation.**

Cohen-Hadria et al., 2019. “Improving singing voice separation using Deep U-Net and Wave-U-Net with data augmentation” in arXiv.

- **Loss function in the spectral domain.**

Akhmetov et al., 2019. “Time Domain Source Separation with Spectral Penalties”. Technical Report.

- **Architectural changes:**

- **Add BiLSTMs at the bottleneck.**

Kaspersen, 2019. “HydraNet: A Network For Singing Voice Separation”. Master Thesis.

- **Use dilated convolutions and dense CNNs.**

Narayanaswamy et al., 2019. “Audio Source Separation via Multi-Scale Learning with Dilated Dense U-Nets” in arXiv.

- **Downsampling & upsampling with discrete wavelet transform (w/ DWT).**

Nakamura et al., 2020. “Time-domain audio source separation based on wave-u-net combined w/ DWT” in ICASSP.

- **Achieve comparable results to a spectrogram-based model: Demucs.**

w/ BiLSTMs at the bottleneck, data augmentation, and some additional architectural changes.

Défossez et al., 2019. “Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed” in arXiv.

Wave-u-net extensions

- Multiplicative conditioning using instrument labels at the bottleneck.

Slizovskaia et al., 2019. “End-to-end Sound Source Separation Conditioned on Instrument Labels” in ICASSP.

- **Data augmentation.**

Cohen-Hadria et al., 2019. “Improving singing voice separation using Deep U-Net and Wave-U-Net with data augmentation” in arXiv.

- Loss function in the spectral domain.

Akhmetov et al., 2019. “Time Domain Source Separation with Spectral Penalties”. Technical Report.

- Architectural changes:

- Add BiLSTMs at the bottleneck.

Kaspersen, 2019. “HydraNet: A Network For Singing Voice Separation”. Master Thesis.

- Use dilated convolutions and dense CNNs.

Narayanaswamy et al., 2019. “Audio Source Separation via Multi-Scale Learning with Dilated Dense U-Nets” in arXiv.

- Downsampling & upsampling with discrete wavelet transform (w/ DWT).

Nakamura et al., 2020. “Time-domain audio source separation based on wave-u-net combined w/ DWT” in ICASSP.

- Achieve comparable results to a spectrogram-based model: Demucs.

w/ BiLSTMs at the bottleneck, data augmentation, and some additional architectural changes.

Défossiez et al., 2019. “Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed” in arXiv.

Data augmentation strategies

It is used to **artificially expand the size of a training dataset** by creating modified versions of it.

- Random swapping left/right channel for each source
- Random scaling sources
- Random mixing of sources from different songs
- Pitch-shifting
- Time-stretching

Uhlich et al, 2017. “Improving music source separation based on deep neural networks through data augmentation and network blending” in ICASSP.

Cohen-Hadria et al., 2019. “Improving singing voice separation using Deep U-Net and Wave-U-Net with data augmentation” in arXiv.

Wave-u-net extensions

- **Multiplicative conditioning using instrument labels at the bottleneck.**

Slizovskaia et al., 2019. “End-to-end Sound Source Separation Conditioned on Instrument Labels” in ICASSP.

- **Data augmentation.**

Cohen-Hadria et al., 2019. “Improving singing voice separation using Deep U-Net and Wave-U-Net with data augmentation” in arXiv.

- **Loss function in the spectral domain.**

Akhmetov et al., 2019. “Time Domain Source Separation with Spectral Penalties”. Technical Report.

- **Architectural changes:**

- **Add BiLSTMs at the bottleneck.**

Kaspersen, 2019. “HydraNet: A Network For Singing Voice Separation”. Master Thesis.

- **Use dilated convolutions and dense CNNs.**

Narayanaswamy et al., 2019. “Audio Source Separation via Multi-Scale Learning with Dilated Dense U-Nets” in arXiv.

- **Downsampling & upsampling with discrete wavelet transform (w/ DWT).**

Nakamura et al., 2020. “Time-domain audio source separation based on wave-u-net combined w/ DWT” in ICASSP.

- **Achieve comparable results to a spectrogram-based model: Demucs.**

w/ BiLSTMs at the bottleneck, data augmentation, and some additional architectural changes.

Défossez et al., 2019. “Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed” in arXiv.

Wave-u-net extensions

- **Multiplicative conditioning** using instrument labels at the bottleneck.

Slizovskaia et al., 2019. “End-to-end Sound Source Separation Conditioned on Instrument Labels” in ICASSP.

- **Data augmentation.**

Cohen-Hadria et al., 2019. “Improving singing voice separation using Deep U-Net and Wave-U-Net with data augmentation” in arXiv.

- **Loss function** in the spectral domain.

Akhmetov et al., 2019. “Time Domain Source Separation with Spectral Penalties”. Technical Report.

- **Architectural changes:**

- **Add BiLSTMs at the bottleneck.**

Kaspersen, 2019. “HydraNet: A Network For Singing Voice Separation”. Master Thesis.

- **Use dilated convolutions and dense CNNs.**

Narayanaswamy et al., 2019. “Audio Source Separation via Multi-Scale Learning with Dilated Dense U-Nets” in arXiv.

- **Downsampling & upsampling with discrete wavelet transform (w/ DWT).**

Nakamura et al., 2020. “Time-domain audio source separation based on wave-u-net combined w/ DWT” in ICASSP.

- **Achieve comparable results to a spectrogram-based model: Demucs.**

w/ BiLSTMs at the bottleneck, data augmentation, and some additional architectural changes.

Défossez et al., 2019. “Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed” in arXiv.

Wave-u-net extensions

- Multiplicative conditioning using instrument labels at the bottleneck.

Slizovskaia et al., 2019. “End-to-end Sound Source Separation Conditioned on Instrument Labels” in ICASSP.

- Data augmentation.

Cohen-Hadria et al., 2019. “Improving singing voice separation using Deep U-Net and Wave-U-Net with data augmentation” in arXiv.

- Loss function in the spectral domain.

Akhmetov et al., 2019. “Time Domain Source Separation with Spectral Penalties”. Technical Report.

- Architectural changes:

- Add BiLSTMs at the bottleneck.

Kaspersen, 2019. “HydraNet: A Network For Singing Voice Separation”. Master Thesis.

- Use dilated convolutions and dense CNNs.

Narayanaswamy et al., 2019. “Audio Source Separation via Multi-Scale Learning with Dilated Dense U-Nets” in arXiv.

- Downsampling & upsampling with discrete wavelet transform (w/ DWT).

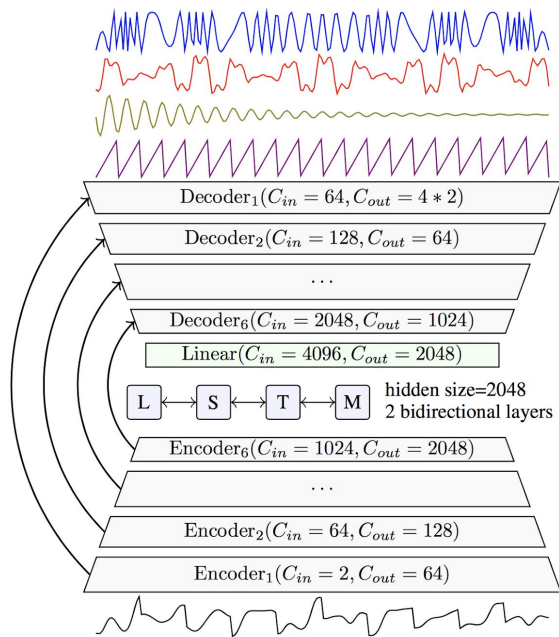
Nakamura et al., 2020. “Time-domain audio source separation based on wave-u-net combined w/ DWT” in ICASSP.

- **Achieve comparable results to a spectrogram-based model: Demucs.**

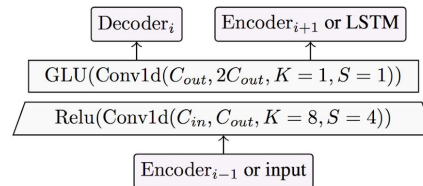
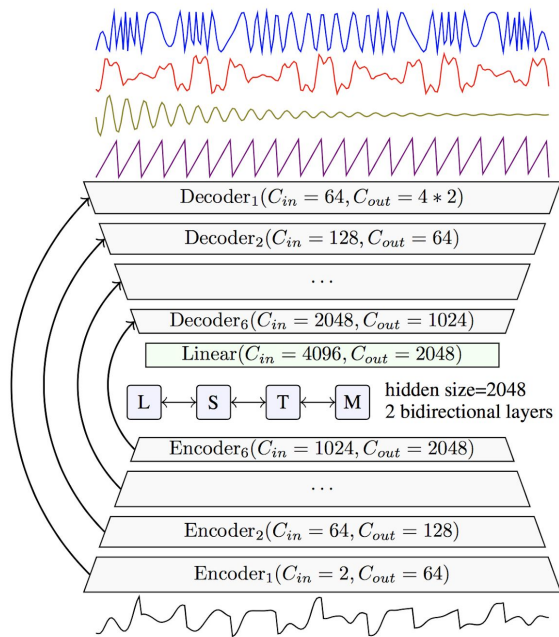
w/ BiLSTMs at the bottleneck, data augmentation, and some additional architectural changes.

Défossiez et al., 2019. “Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed” in arXiv.

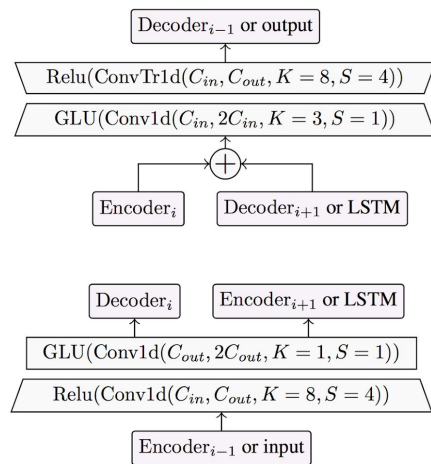
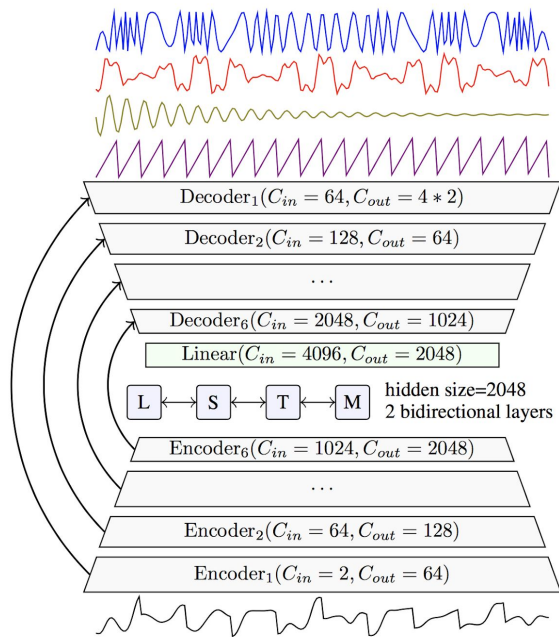
Wave-u-net extensions: Demucs



Wave-u-net extensions: Demucs



Wave-u-net extensions: Demucs

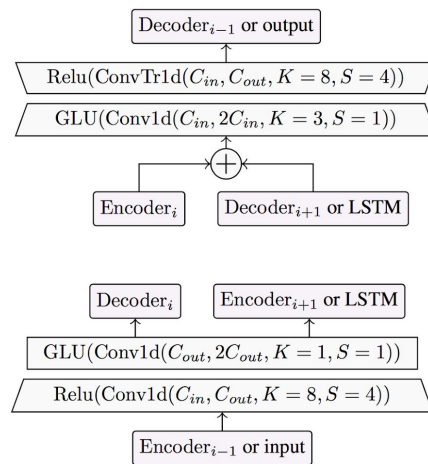


Wave-u-net extensions: Wave-U-net vs. Demucs

Block	Operation	Shape
	Input	(16384, 1)
DS, repeated for $i = 1, \dots, L$	Conv1D($F_c \cdot i, f_d$) Decimate	(4, 288)
	Conv1D($F_c \cdot (L + 1), f_d$)	(4, 312)
US, repeated for $i = L, \dots, 1$	Upsample Concat(DS block i) Conv1D($F_c \cdot i, f_u$)	(16834, 24)
	Concat(Input)	(16834, 25)
	Conv1D($K, 1$)	(16834, 2)

Wave-U-net: building blocks

ENCODER



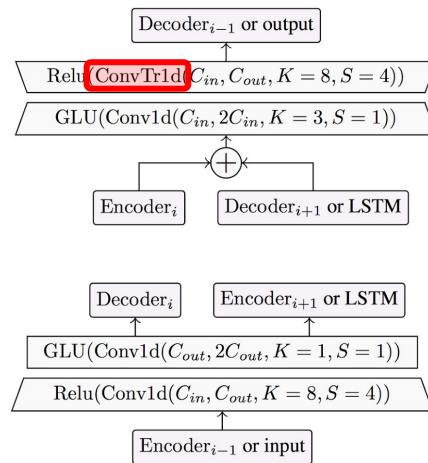
Demucs: building blocks

Wave-u-net extensions: Wave-U-net vs. Demucs

Block	Operation	Shape
	Input	(16384, 1)
DS, repeated for $i = 1, \dots, L$	Conv1D($F_c \cdot i, f_d$) Decimate	(4, 288)
	Conv1D($F_c \cdot (L + 1), f_d$)	(4, 312)
US, repeated for $i = L, \dots, 1$	Upsample Concat(DS block i) Conv1D($F_c \cdot i, f_u$)	(16834, 24)
	Concat(Input)	(16834, 25)
	Conv1D($K, 1$)	(16834, 2)

Wave-U-net: building blocks

DECODER

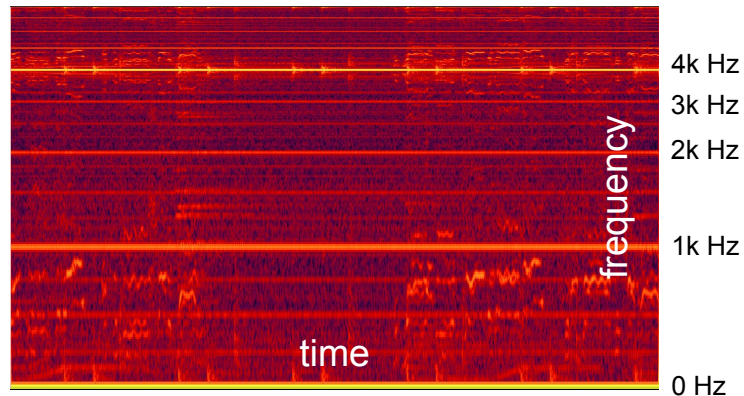


Demucs: building blocks

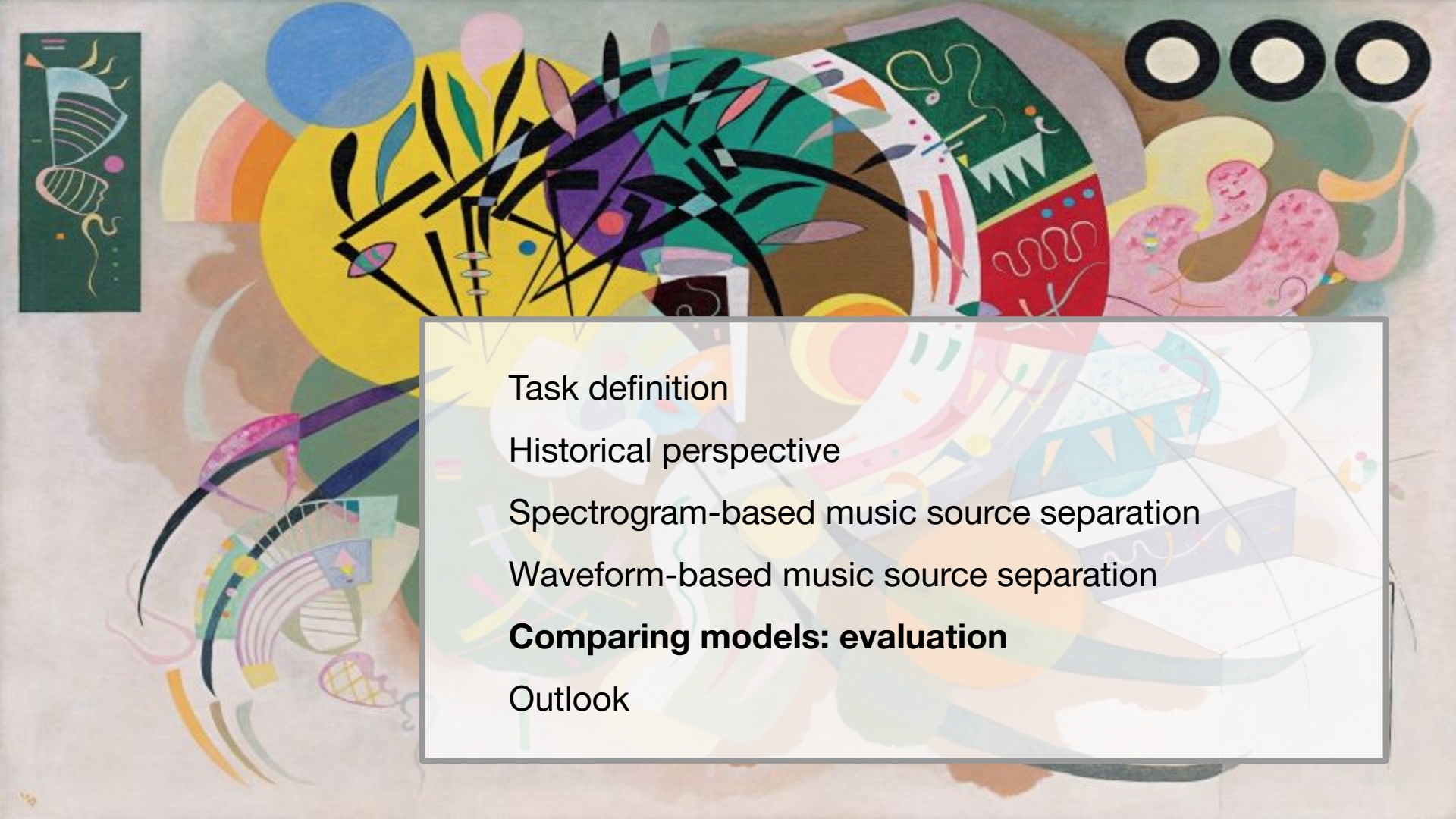
Deconvolutions and high-frequency artifacts



Checkerboard artifacts in images



High-frequency buzzing noise in audio



Task definition

Historical perspective

Spectrogram-based music source separation

Waveform-based music source separation

Comparing models: evaluation

Outlook

Evaluation metrics: SDR, SIR, SAR

$$\text{SDR} := 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}\|^2}$$

“overall performance”

$$\text{SIR} := 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2}$$

“interference from other sources”

$$\text{SAR} := 10 \log_{10} \frac{\|s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}}\|^2}{\|e_{\text{artif}}\|^2}$$

“algorithmic artifacts”

http://craffel.github.io/mir_eval/

<https://github.com/sigsep/sigsep-mus-eval/>

Subjective evaluation

Session 1	Block 1	Trial 1
<p>Attending ONLY to the BACKGROUND, select the category which best describes the sample you just heard.</p>		
<p>the BACKGROUND in this sample was</p>		
<p>5 - NOT NOTI</p> <p>4 - SLIGHTLY</p> <p>3 - NOTICEAE</p> <p>2 - SOMEWHAT</p> <p>1 - VERY INTRU</p>	<p>Select the category which best describes the sample you just heard for purposes of everyday speech communication.</p> <p>the OVERALL SPEECH SAMPLE was</p> <p>5 - EXCELLENT</p> <p>4 - GOOD</p> <p>3 - FAIR</p> <p>2 - POOR</p> <p>1 - BAD</p>	
<p>Attending ONLY to the SPEECH SIGNAL, select the category which best describes the sample you just heard.</p>		
<p>the SPEECH SIGNAL in this sample was</p>		
<p>5 - DISTORTED</p> <p>4 - DISTORTED</p> <p>3 - DISTORTED</p> <p>2 - DISTORTED</p> <p>1 - DISTORTED</p>		

ITU-T Recommendation P.835

Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm

Which architectures seem to work the best?

Model	Domain	# Param	Test SDR (dB)				
			Vocals	Drums	Bass	Other	Average
IRM oracle	N/A	N/A	9.43	8.45	7.12	7.85	8.21
DeepConvSep [29]	Spectrogram	0.32M	2.37	3.14	0.17	-2.13	0.89
WaveNet [30]	Waveform	3.30M	3.35	4.13	2.49	2.60	2.60
Wave-U-Net [13]	Waveform	10.20M	3.25	4.22	3.21	2.25	3.23
Spect U-Net [31]	Spectrogram	9.84M	5.74	4.66	3.67	3.40	4.37
Open-Unmix [11]	Spectrogram	8.90M	6.32	5.73	5.23	4.02	5.36
Demucs [14]	Waveform	66.42M	6.29	6.08	5.83	4.12	5.58
Meta-TasNet [32]	Waveform	12.00M	6.40	5.91	5.58	4.19	5.52
MMDenseLSTM [16]	Spectrogram	4.88M	6.60	6.41	5.16	4.15	5.58
Sams-Net	Spectrogram	3.70M	6.61	6.63	5.25	4.09	5.65

Which architectures seem to work the best?

Model	Domain	# Param	Test SDR (dB)				
			Vocals	Drums	Bass	Other	Average
IRM oracle	N/A	N/A	9.43	8.45	7.12	7.85	8.21
DeepConvSep [29]	Spectrogram	0.32M	2.37	3.14	0.17	-2.13	0.89
WaveNet [30]	Waveform	3.30M	3.35	4.13	2.49	2.60	2.60
Wave-U-Net [13]	Waveform	10.20M	3.25	4.22	3.21	2.25	3.23
Spect U-Net [31]	Spectrogram	9.84M	5.74	4.66	3.67	3.40	4.37
Open-Unmix [11]	Spectrogram	8.90M	6.32	5.73	5.23	4.02	5.36
Demucs [14]	Waveform	66.42M	6.29	6.08	5.83	4.12	5.58
Meta-TasNet [32]	Waveform	12.00M	6.40	5.91	5.58	4.19	5.52
MMDenseLSTM [16]	Spectrogram	4.88M	6.60	6.41	5.16	4.15	5.58
Sams-Net	Spectrogram	3.70M	6.61	6.63	5.25	4.09	5.65

Which architectures seem to work the best?

Model	Domain	# Param	Test SDR (dB)				Average
			Vocals	Drums	Bass	Other	
IRM oracle	N/A	N/A	9.43	8.45	7.12	7.85	8.21
DeepConvSep [29]	Spectrogram	0.32M	2.37	3.14	0.17	-2.13	0.89
WaveNet [30]	Waveform	3.30M	3.35	4.13	2.49	2.60	2.60
Wave-U-Net [13]	Waveform	10.20M	3.25	4.22	3.21	2.25	3.23
Spect U-Net [31]	Spectrogram	9.84M	5.74	4.66	3.67	3.40	4.37
Open-Unmix [11]	Spectrogram	8.90M	6.32	5.73	5.23	4.02	5.36
Demucs [14]	Waveform	66.42M	6.29	6.08	5.83	4.12	5.58
Meta-TasNet [32]	Waveform	12.00M	6.40	5.91	5.58	4.19	5.52
MMDenseLSTM [16]	Spectrogram	4.88M	6.60	6.41	5.16	4.15	5.58
Sams-Net	Spectrogram	3.70M	6.61	6.63	5.25	4.09	5.65

Which architectures seem to work the best?

Model	Domain	# Param	Test SDR (dB)				
			Vocals	Drums	Bass	Other	Average
IRM oracle	N/A	N/A	9.43	8.45	7.12	7.85	8.21
DeepConvSep [29]	Spectrogram	0.32M	2.37	3.14	0.17	-2.13	0.89
WaveNet [30]	Waveform	3.30M	3.35	4.13	2.49	2.60	2.60
Wave-U-Net [13]	Waveform	10.20M	3.25	4.22	3.21	2.25	3.23
Spect U-Net [31]	Spectrogram	9.84M	5.74	4.66	3.67	3.40	4.37
Open-Unmix [11]	Spectrogram	8.90M	6.32	5.73	5.23	4.02	5.36
Demucs [14]	Waveform	66.42M	6.29	6.08	5.83	4.12	5.58
Meta-TasNet [32]	Waveform	12.00M	6.40	5.91	5.58	4.19	5.52
MMDenseLSTM [16]	Spectrogram	4.88M	6.60	6.41	5.16	4.15	5.58
Sams-Net	Spectrogram	3.70M	6.61	6.63	5.25	4.09	5.65

Which architectures seem to work the best?

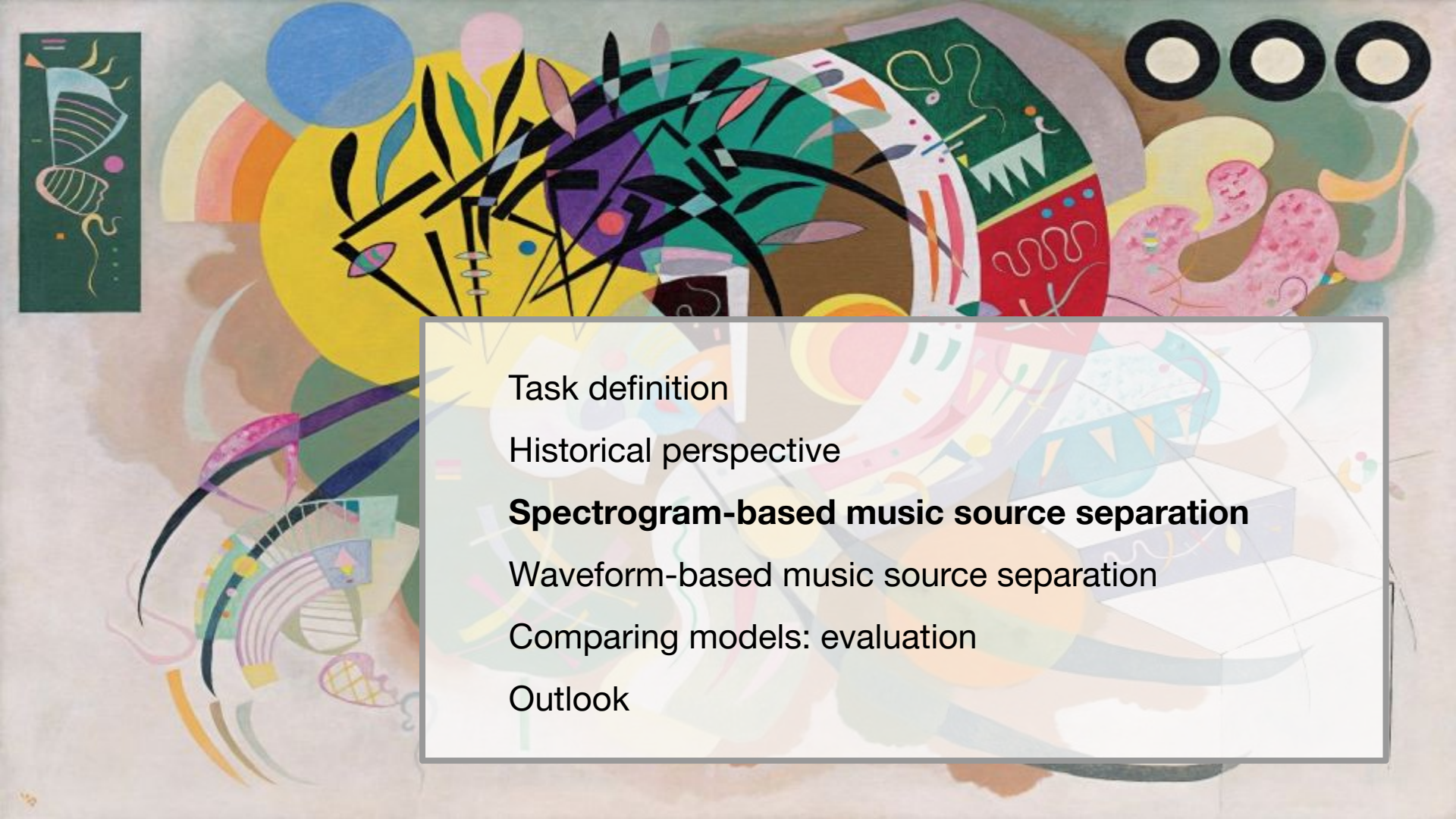
Model	Domain	# Param	Test SDR (dB)				
			Vocals	Drums	Bass	Other	Average
IRM oracle	N/A	N/A	9.43	8.45	7.12	7.85	8.21
DeepConvSep [29]	Spectrogram	0.32M	2.37	3.14	0.17	-2.13	0.89
WaveNet [30]	Waveform	3.30M	3.35	4.13	2.49	2.60	2.60
Wave-U-Net [13]	Waveform	10.20M	3.25	4.22	3.21	2.25	3.23
Spect U-Net [31]	Spectrogram	9.84M	5.74	4.66	3.67	3.40	4.37
Open-Unmix [11]	Spectrogram	8.90M	6.32	5.73	5.23	4.02	5.36
Demucs [14]	Waveform	66.42M	6.29	6.08	5.83	4.12	5.58
Meta-TasNet [32]	Waveform	12.00M	6.40	5.91	5.58	4.19	5.52
MMDenseLSTM [16]	Spectrogram	4.88M	6.60	6.41	5.16	4.15	5.58
Sams-Net	Spectrogram	3.70M	6.61	6.63	5.25	4.09	5.65

Which architectures seem to work the best?

Model	Domain	# Param	Test SDR (dB)				
			Vocals	Drums	Bass	Other	Average
IRM oracle	N/A	N/A	9.43	8.45	7.12	7.85	8.21
DeepConvSep [29]	Spectrogram	0.32M	2.37	3.14	0.17	-2.13	0.89
WaveNet [30]	Waveform	3.30M	3.35	4.13	2.49	2.60	2.60
Wave-U-Net [13]	Waveform	10.20M	3.25	4.22	3.21	2.25	3.23
Spect U-Net [31]	Spectrogram	9.84M	5.74	4.66	3.67	3.40	4.37
Open-Unmix [11]	Spectrogram	8.90M	6.32	5.73	5.23	4.02	5.36
Demucs [14]	Waveform	648M	6.29	6.08	5.83	4.12	5.58
Meta-TasNet [32]	Waveform	52M	6.40	5.91	5.58	4.19	5.52
MMDenseLSTM [16]	Spectrogram	4.88M	6.60	6.41	5.16	4.15	5.58
Sams-Net	Spectrogram	3.70M	6.61	6.63	5.25	4.09	5.65

Which architectures seem to work the best?

Model	Domain	MOS Quality	MOS Contamination
Open-Unmix	spectrogram	3.0 / 5	3.3 / 5
Demucs	waveform	3.2 / 5	3.3 / 5
Conv-Tasnet	waveform	2.9 / 5	3.4 / 5



Task definition

Historical perspective

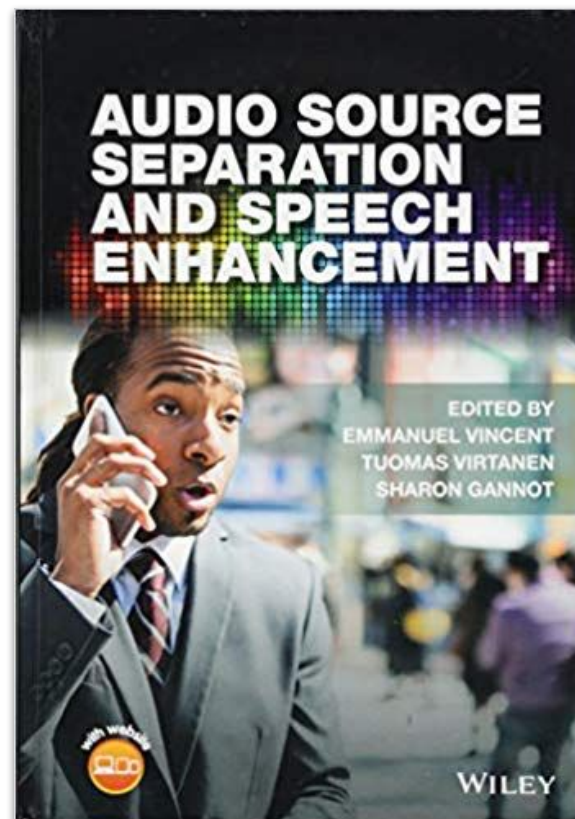
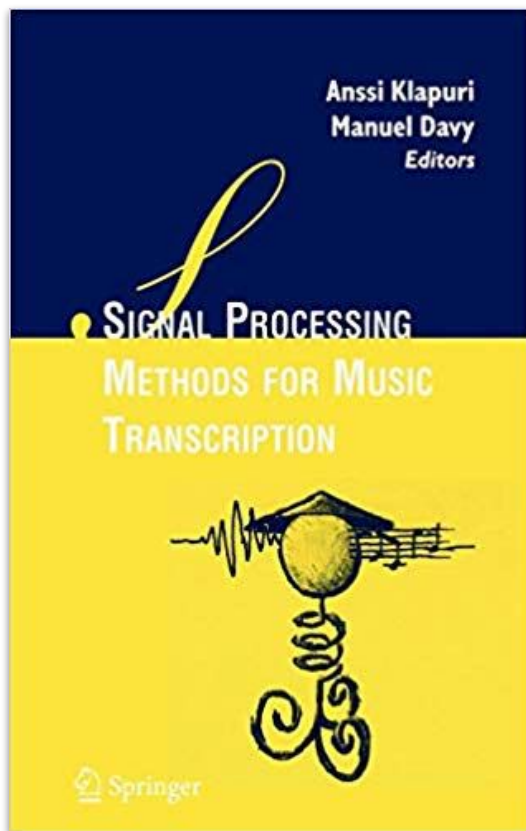
Spectrogram-based music source separation

Waveform-based music source separation

Comparing models: evaluation

Outlook

Additional references



Einfonia Cano, Derry FitzGerald, Antoine Liukas,
Mark D. Rumbley, and Fabian-Robert Stier

Musical Source Separation

An introduction



Many people listen to recorded music as part of their everyday lives, e.g., from radio or TV programs, compact discs, downloads, or, increasingly, online streaming services. Sometimes we might want to remix the balance within the music, perhaps to make the vocals louder or to suppress an unwanted sound, or we might want to upmix a two-channel stereo recording to a 5.1-channel surround sound system. We might also want to change the spatial location of a musical instrument within the mix. All of these applications are relatively straightforward, provided we have access to separate sound channels (stems) for each musical audio object.

However, if we only have access to the final recording mix, which is usually the case, this is much more challenging. To estimate the original musical sources, which would allow us to remix, suppress, or upmix the sources, we need to perform musical source separation (MSS).

In the general source separation problem, we are given one or more mixture signals that contain different combinations of some original source signals. This is illustrated in Figure 1, where four sources, i.e., vocals, drums, bass, and guitar, are all present in the mixture. The task is to recover one or more of the source signals given the mixtures. In some cases, this is relatively straightforward, e.g., if there are at least as many mixtures as there are sources and if the mixing process is fixed, with no delays, filters, or nonlinear mastering [1].

However, MSS is normally more challenging. Typically, there may be many musical instruments and voices in a two-channel recording, and the sources have often been processed with the addition of filters and reverbification (sometimes nonlinear) in the recording and mixing process. In some cases, the sources may move or the production parameters may change, meaning that the mixture is time varying.

Nevertheless, musical sound sources have particular properties and structures that can help us. For example, musical source signals often have a regular harmonic structure of frequencies at regular intervals and can have frequency contours characteristic of each musical instrument. They may also repeat, in particular, temporal patterns based on the musical structure.

Digital Music Analysis 16 (2004) 2018–2019
doi:10.1016/j.dma.2004.06.001

1053-088X/19/020101

IEEE SIGNAL PROCESSING MAGAZINE | January 2019 |

31

End-to-end music source separation: is it possible in the waveform domain?

Francesc Lluis* Jordi Pons* Xavier Serra

Music Technology Group, Universitat Pompeu Fabra, Barcelona.

name.surname@upf.edu

Abstract

Most of the currently successful source separation techniques use the magnitude spectrum as input, and are therefore by default omitting useful information, we study the viability of using end-to-end models for music source separation — which take into account all the information available in the raw audio signal, including the phase. Although during the last decades end-to-end music source separation has been considered almost unattainable, our results confirm that waveform-based models can perform similarly (if not better) than a spectrum-based deep learning model. Namely, a WavNet-based model we propose and Wave-U-Net can outperform DeepConvSep, a recent spectrum-based deep learning model.

Index Terms: source separation, end-to-end learning.

1. Introduction

When two or more sounds co-exist, they interfere with each other resulting in a novel mixture signal where sounds are superposed (and, sometimes, masked). The source separation task tackles the inverse problem of recovering each individual sound source contribution from an observed mixture signal.

With the recent advances in deep learning, source separation techniques have improved substantially [1]. Interestingly, though, nearly all successful deep learning algorithms use the magnitude spectrum as input [1, 2, 3] — and are therefore, by default, omitting part of the signal: the phase. Omitting the potentially useful information of the phase entails the risk of finding a sub-optimal solution. In this work, we aim to take full advantage of the acoustic modeling capabilities of deep learning to investigate whether it is possible to approach the problem of music source separation directly in an end-to-end learning fashion. Consequently, our investigation is centered on studying how to separate music sources (e.g., singing voice, bass or drums) directly from the raw waveform music mixture.

During the last two decades, matrix decomposition methods have dominated the field of audio source separation. Several algorithms have been proposed throughout the years, with independent component analysis (ICA) [4], sparse coding [5], or non-negative matrix factorization (NMF) [6] being the most used ones. Given that magnitude or power spectrum representations are always non-negative, imposing a non-negative constraint (like in NMF) is particularly useful when analyzing these spectrograms — but less appropriate for processing waveforms, which range from -1 to 1 . For that reason, methods like ICA and sparse coding have historically been used to process waveforms [7, 8, 9]. Waveform representations preserve all the information available in the raw signal. However, given the unpredictable behavior of the phase in real-life

sounds, it is rare to find identical waveforms produced by the same sound source. As a result of this variability, a single basis' cannot represent a sound source and therefore, one requires *i)* a large amount of bases, or *ii)* shift-invariant bases to obtain accurate decompositions [8, 10]. Although several matrix decomposition methods have been used for decomposing waveform-based mixtures [7, 8, 9], these have never worked as well as the spectrogram-based ones.

Due to the above mentioned difficulties, the phase of complex time-frequency representations is commonly discarded, assuming that magnitude spectrograms already carry meaningful information about the sound sources to be separated. Phase-related problems disappear when sounds are just represented as magnitude or power spectrograms, since different realizations of the same sound are almost identical in this time-frequency plane. This allows to easily overcome the variability problem found when operating with waveforms.

Most matrix decomposition methods rely on a signal model assuming that sources add linearly in the time domain [10]. However, the addition of signals in the time and frequency domains is not equivalent if phases are discarded. Only in expectation: $E\{|X(k)|^2\} = |Y_1(k)|^2 + |Y_2(k)|^2$, where $X(k) = DFT\{x(t)\}$. This means that we can approximate the time-domain summation in the power spectral domain. For that reason, many approaches utilize power spectrograms as inputs. Although magnitude spectrograms work well in practice [11], there is no similar theoretical justification for such an inconsistency with the signal model when the phases are discarded.

Finally, note that these methods operating on top of spectrograms still need to deliver a waveform signal. To this end, the main practice is to filter the original magnitude or power spectrogram with (predicted) time-frequency masks. Accordingly, the original noisy phase of the mixture is used when synthesizing the waveforms of the estimated sources — which might introduce an additional source of error [10]. Notably, many modern spectrogram-based deep learning models are also relying on this same (potentially problematic) approach [2, 12]. To overcome this issue, some tried to consider the phase when separating the sources [13, 14, 15], or some others relied on a sinusoidal signal model at synthesis time [16]. However, in our work, we do not want to rely on any time-frequency transform or any signal model. Instead, we aim to directly approach the problem in the waveform domain.

As seen, many issues still exist around the idea of discarding the phase: are we missing crucial information when discarding it? When using the phase of the mixture at synthesis time, are we introducing artifacts that are limiting our model's performance? Or, since magnitude spectrograms (differently from

ICA, sparse coding or NMF model) the mixture signal as a weighted sum of bases, which represent a source or components of a source.

Using the full complex STFT number, instead of utilizing phaseless representations (either as the input or when applying the masks).

*Contributed equally.

<https://sigsep.github.io/tutorials/>



Music source separation with deep learning

Jordi Pons / www.jordipons.me / @jordiponsdotme