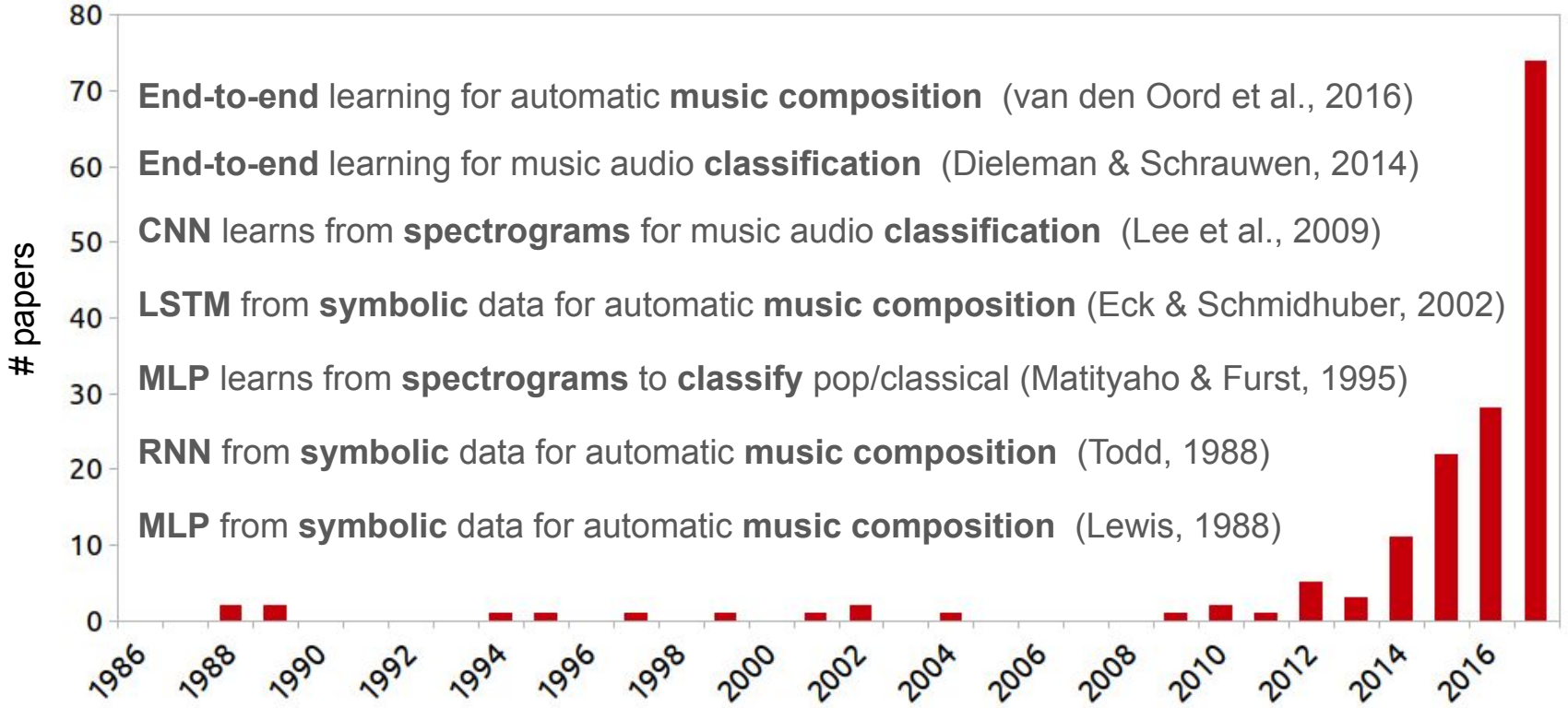
Music source separation with deep learning

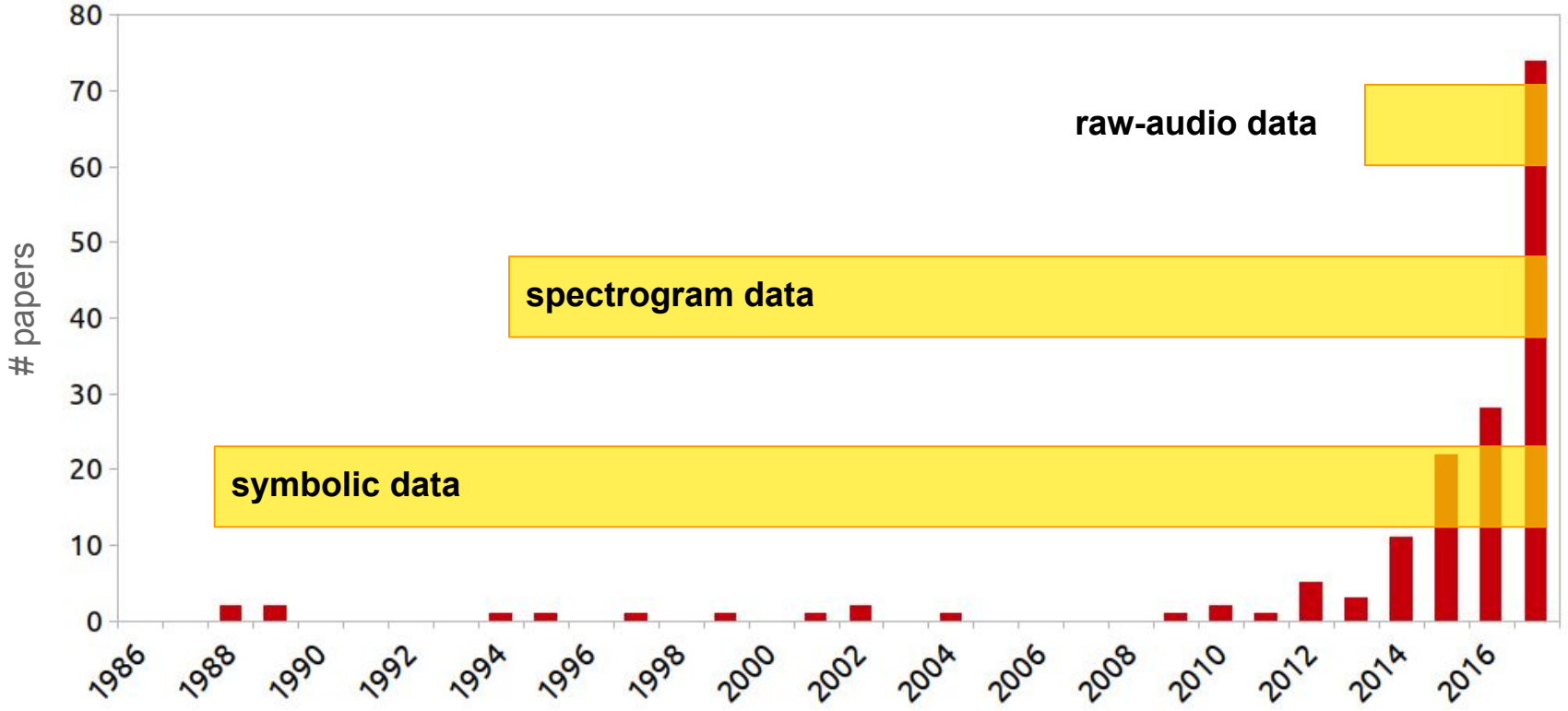
Jordi Pons / www.jordipons.me / @jordiponsdotme


Historical perspective
Context and trends

Papers on “neural networks & music”: milestones



Papers on “neural networks & music”: input data



References

Van den Oord et al., 2016. “Wavenet: a generative model for audio” in arXiv.

Dieleman and Schrauwen, 2014. “End-to-end learning for music audio”
in International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Lee et al., 2009. “Unsupervised feature learning for audio classification using convolutional deep belief networks”
in Advances in Neural Information Processing Systems (NIPS).

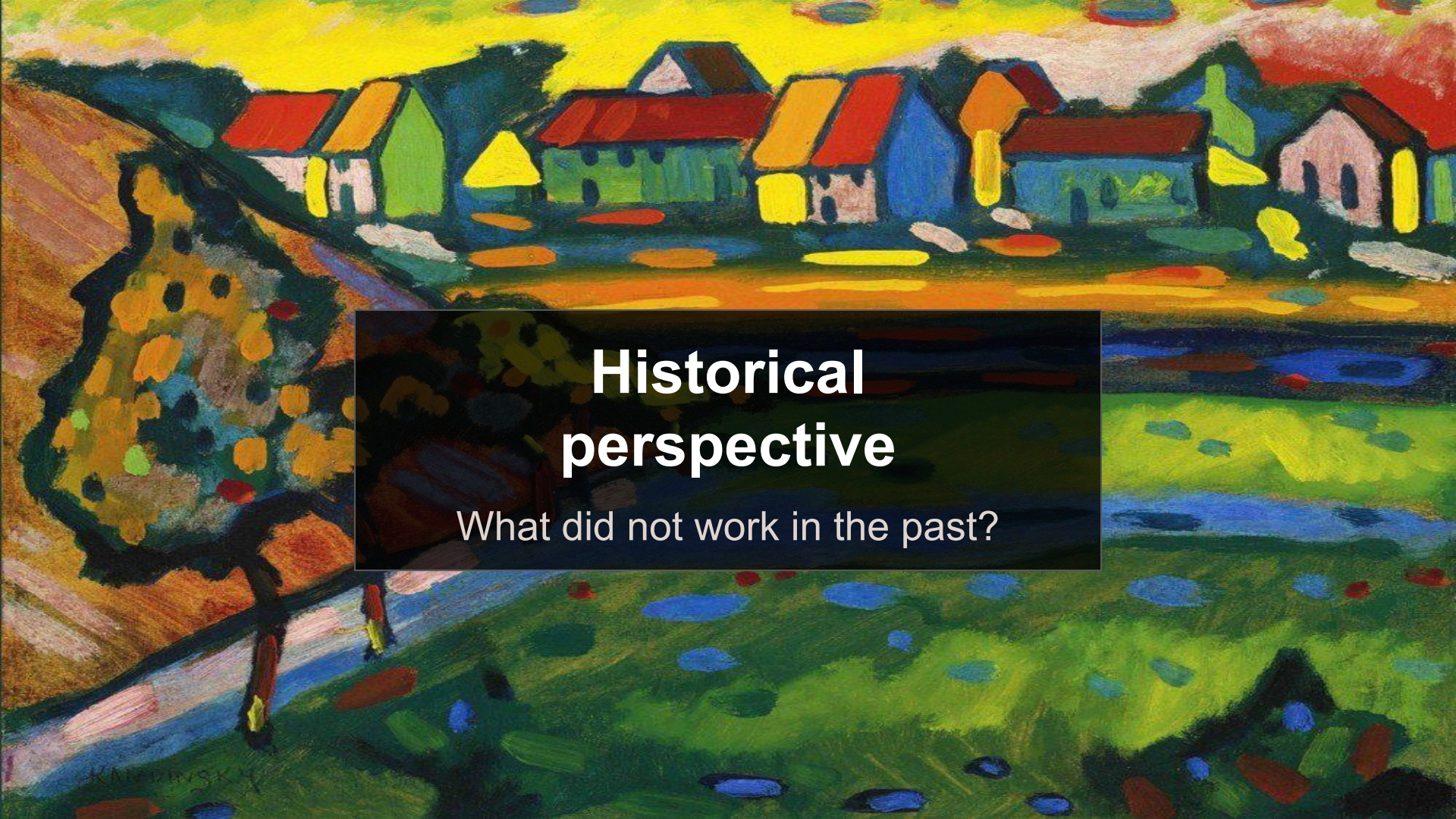
Matityaho and Furst, 1995. “Neural network based model for classification of music type”
in 18th Convention of Electrical and Electronics Engineers in Israel. IEEE, 4–3.

Eck and Schmidhuber, 2002. “Finding temporal structure in music: Blues improvisation with LSTM recurrent networks”
in Proceedings of the Workshop on Neural Networks for Signal Processing.

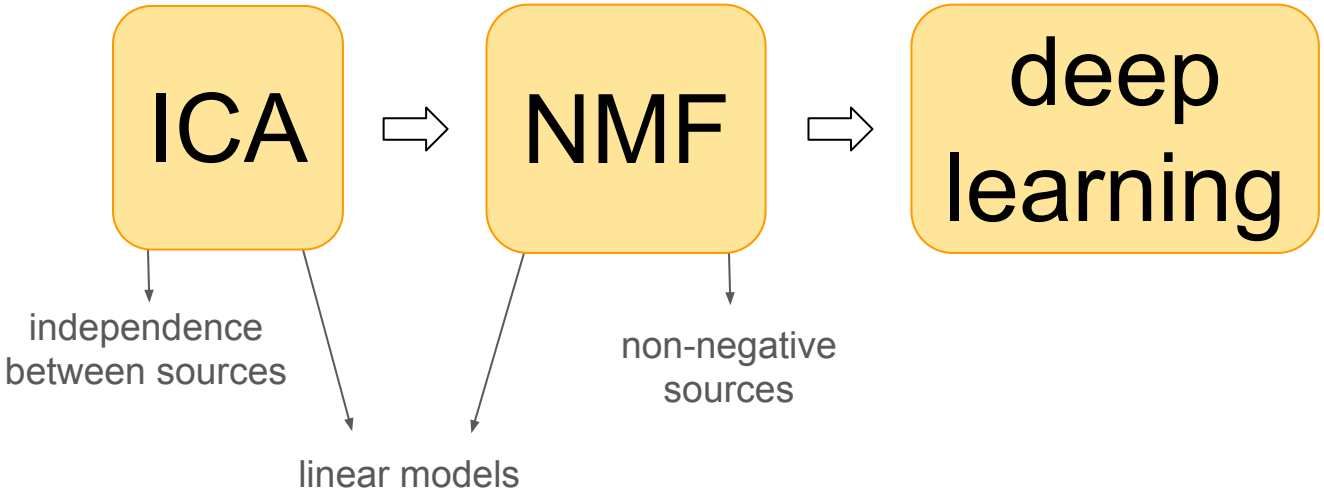
Todd, 1988. “A sequential network design for musical applications”
in Proceedings of the Connectionist Models Summer School.

Lewis, 1988. Creation by Refinement: A creativity paradigm for gradient descent learning networks”
in International Conference on Neural Networks.

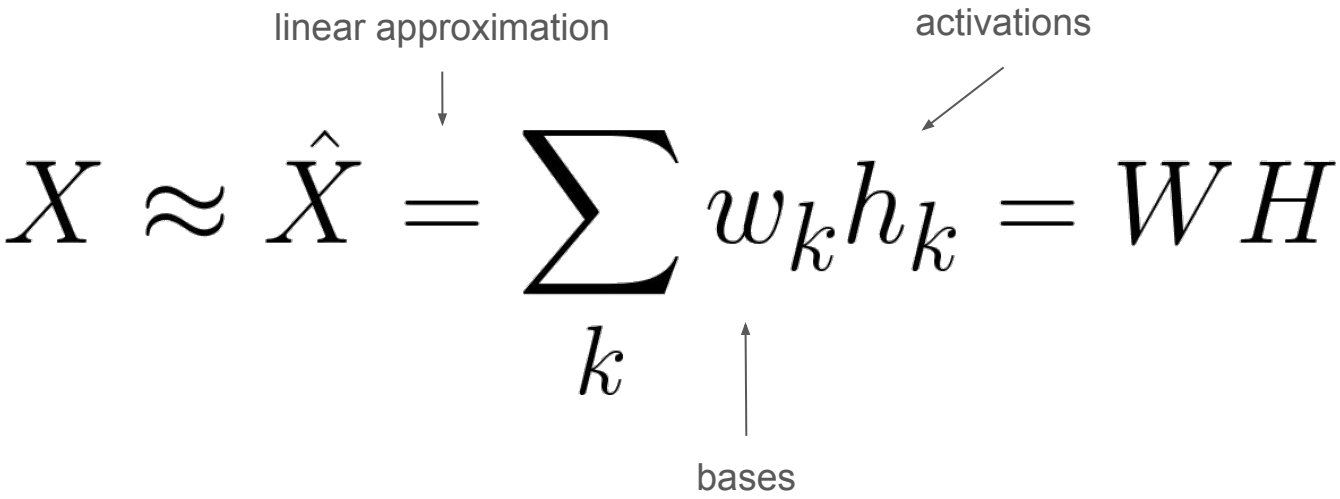
Historical review from: Pons, 2019. “Deep neural networks for music and audio tagging”. PhD Thesis.


Historical perspective
What did not work in the past?

Historical perspective: unsupervised & linear models

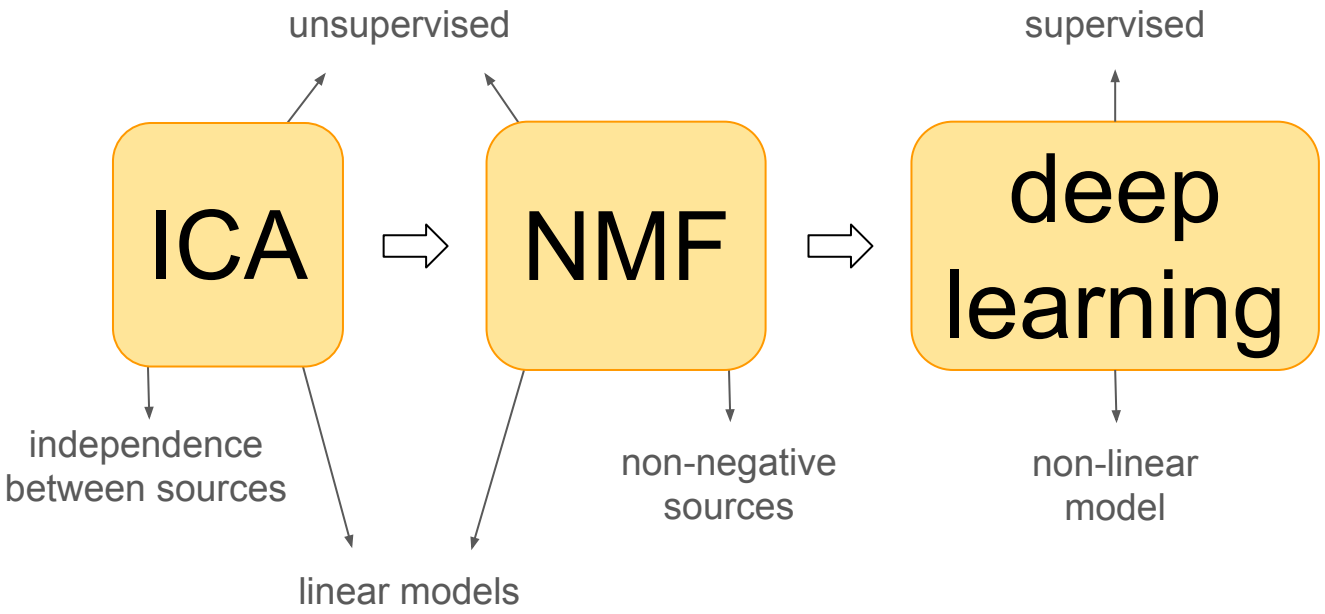


Linear model example



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

Historical perspective: unsupervised & linear models



A widely-used set of tools:

filtering spectrograms

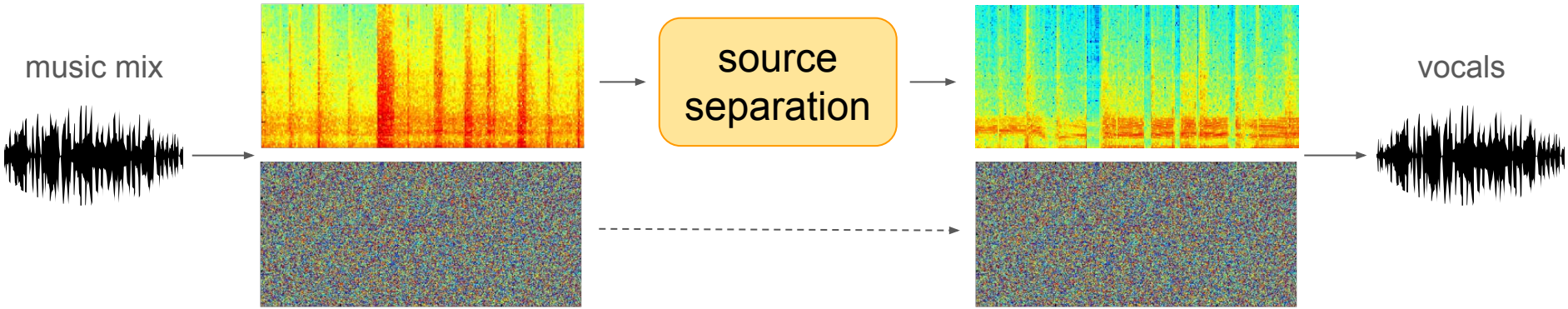
linear models

unsupervised learning

audio domain knowledge


Spectrogram-based
music source separation
Supervised learning & non-linear models

Spectrogram-based music source separation



Muth et al., 2018. “Improving DNN-based Music Source Separation using Phase Features” in ICML Workshop on ML for music.

Filtering spectrograms with masks

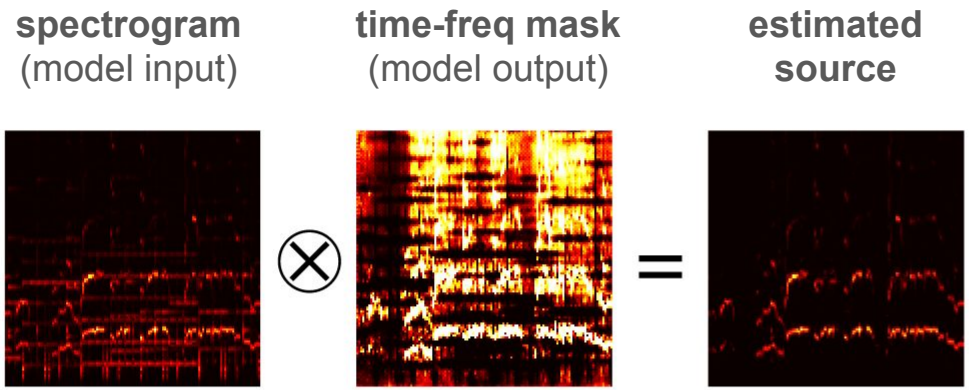
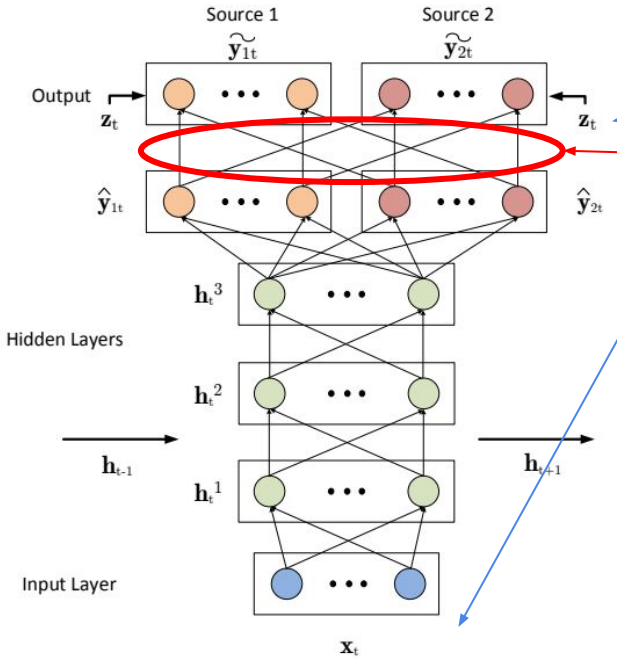


Figure from: Jansson et al., 2017. "Singing voice separation with deep U-Net convolutional networks" in ISMIR.

Deep recurrent neural networks



Input spectrogram

Mask:

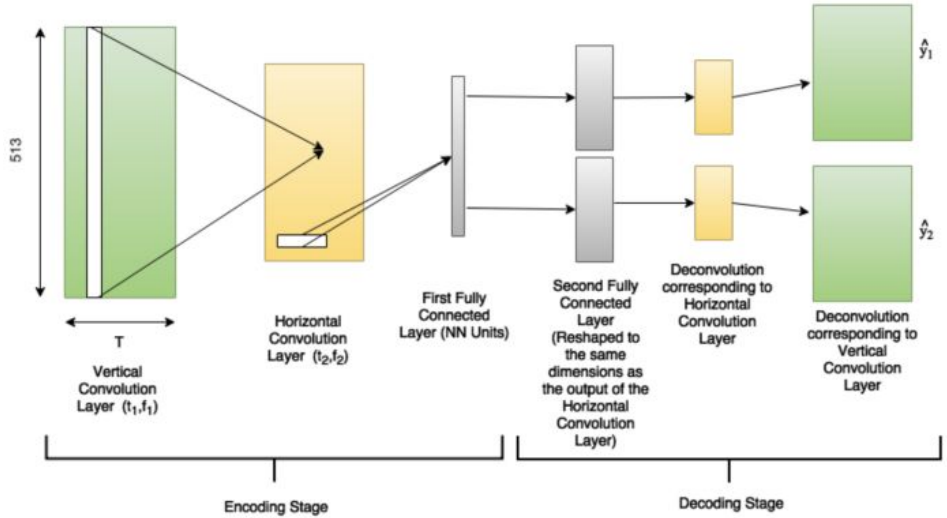
Output predictions

$$\tilde{\mathbf{y}}_{1_t} = \frac{|\hat{\mathbf{y}}_{1_t}|}{|\hat{\mathbf{y}}_{1_t}| + |\hat{\mathbf{y}}_{2_t}|} \odot \mathbf{z}_t$$

$$\tilde{\mathbf{y}}_{2_t} = \frac{|\hat{\mathbf{y}}_{2_t}|}{|\hat{\mathbf{y}}_{1_t}| + |\hat{\mathbf{y}}_{2_t}|} \odot \mathbf{z}_t,$$

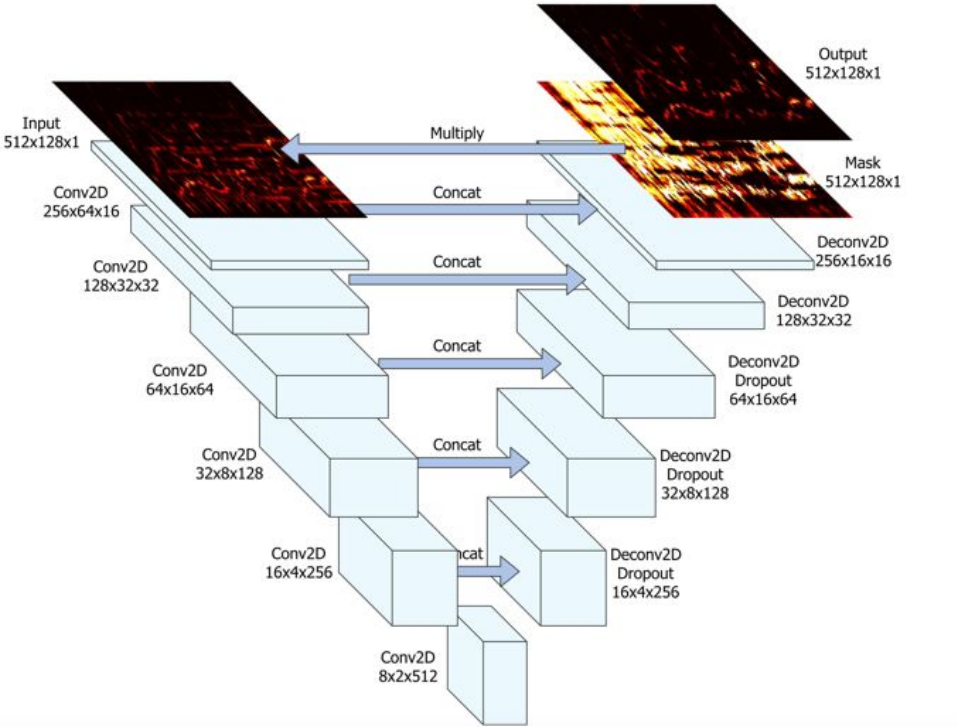
Huang et al., 2014. “Singing-voice separation from monaural recordings using deep recurrent neural networks” in ICASSP.

Convolutional auto-encoder



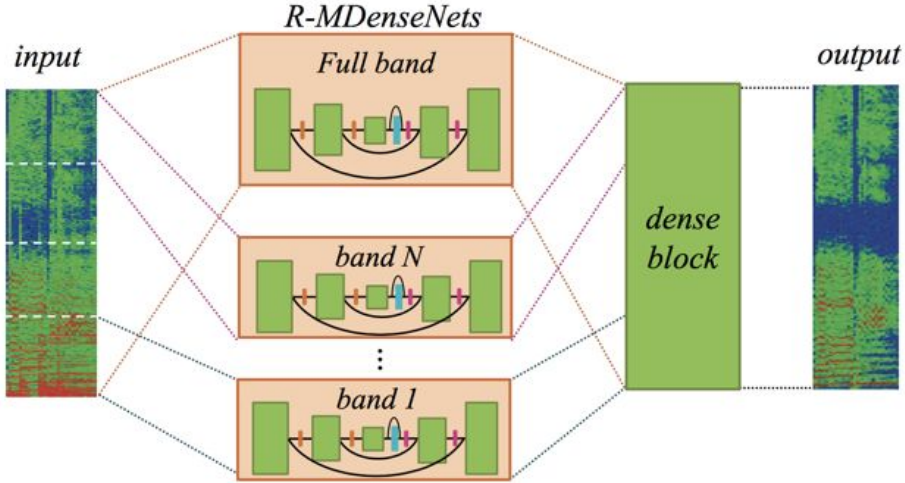
Chandna et al., 2017. “Monaural audio source separation using deep convolutional neural networks” in LVA/ICA.

U-net auto-encoder



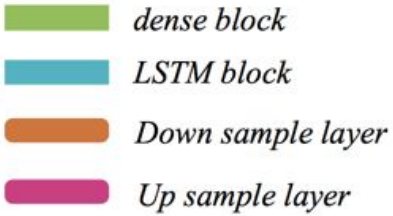
Jansson et al., 2017. “Singing voice separation with deep U-net” in ISMIR.

MMDenseLSTM



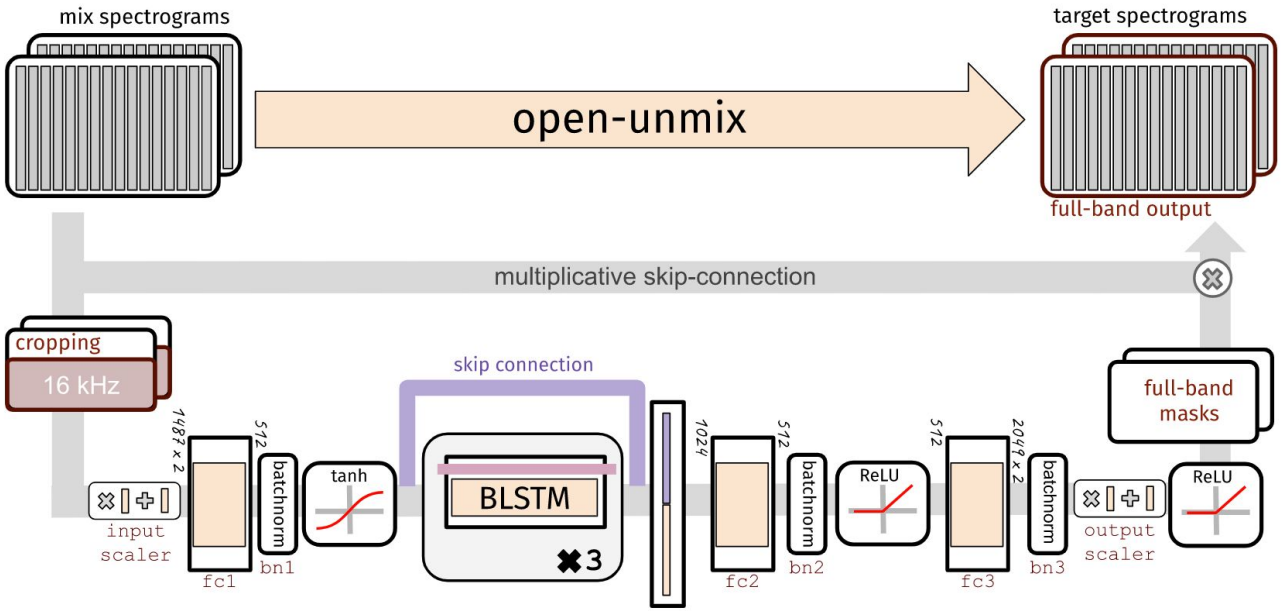
MM

- Multi-scale
- Multi-band



Takahashi et al., 2018. “MMDenseLSTM: an efficient combination of convolutional and recurrent neural networks for audio source separation” in IWAENC.

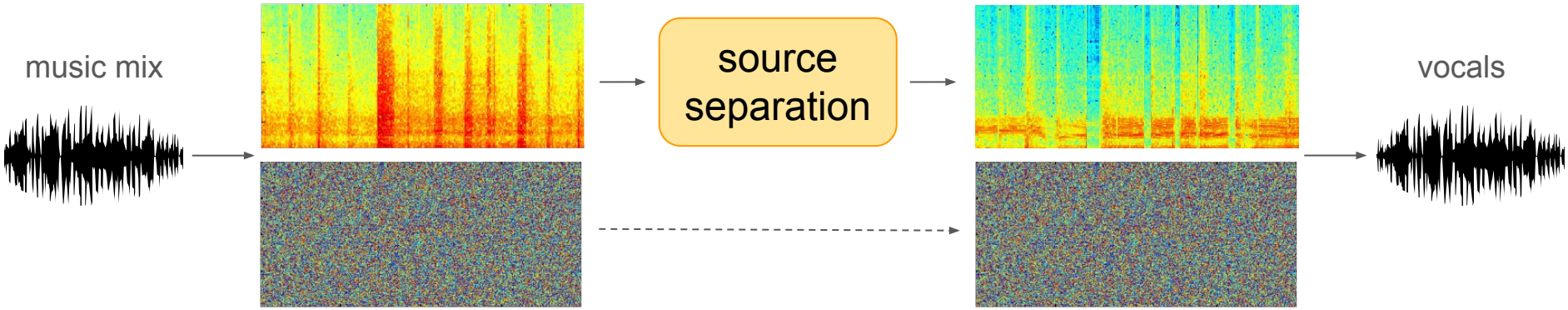
Open-unmix: a state-of-the-art implementation



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


Waveform-based
music source separation
End-to-end learning

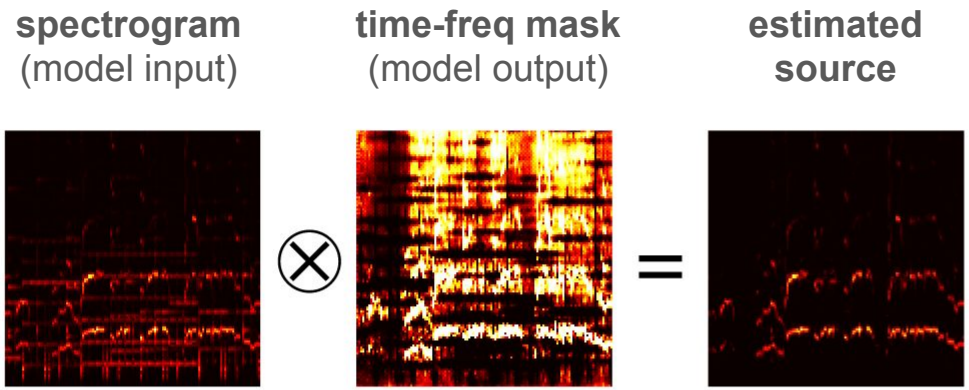
Why end-to-end music source separation?



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

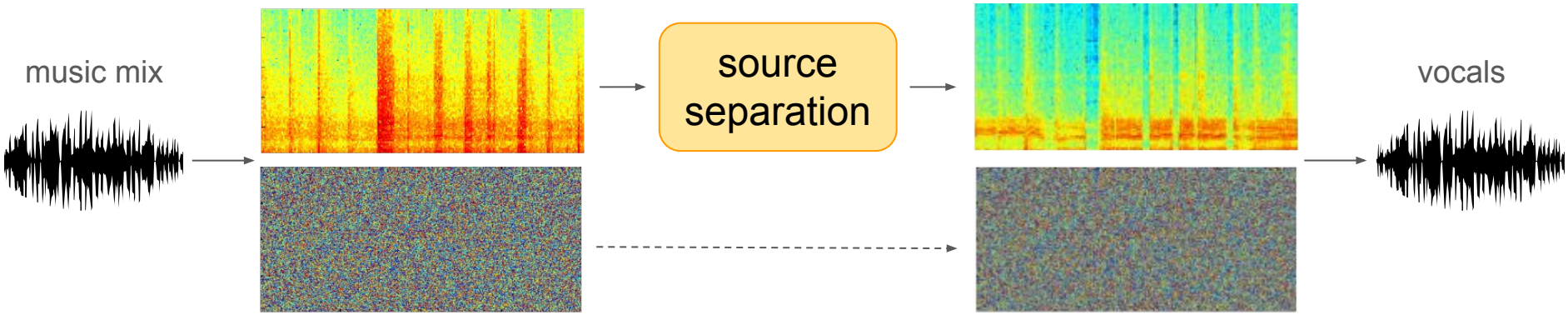
II) When using the **phase of the mixture at synthesis time**,are we introducing artifacts that are limiting our model's performance?

Why filtering spectrograms with masks?



III) **Filtering** spectrograms does **not allow recovering masked (perceptually hidden sound) signals**

Figure from: Jansson et al., 2017. "Singing voice separation with deep U-Net convolutional networks" in ISMIR.

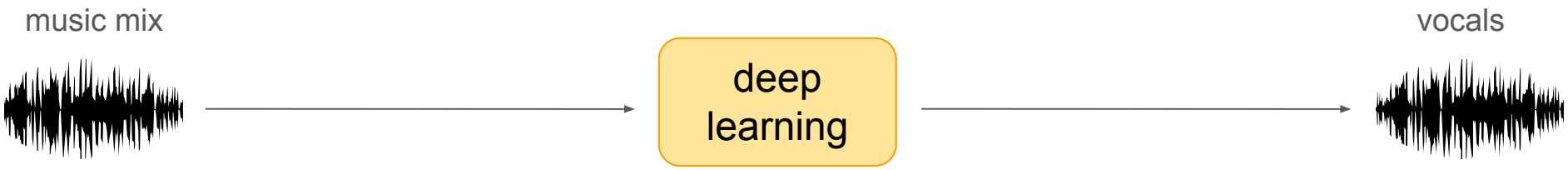


I) Are we missing crucial information when **discarding the phase**?

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

III) **Filtering** spectrograms does **not allow recovering masked signals**

End-to-end music source separation



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 Spetrum Approximation” in ICASSP.

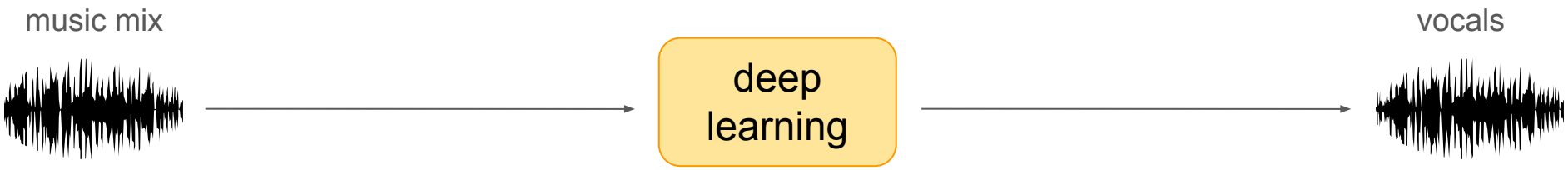
Other (active) research directions:

Alternative models at synthesis time?

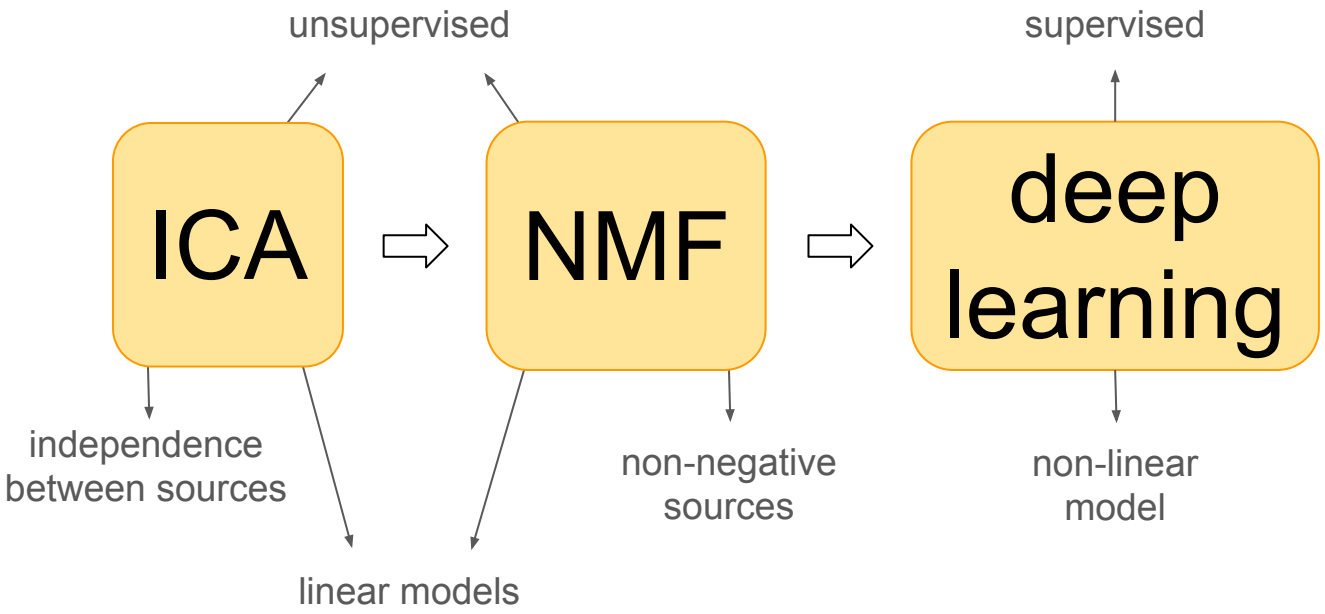
Virtanen and Klapuri, 2000. “Separation of harmonic sound sources using **sinusoidal modeling**,” in ICASSP.

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

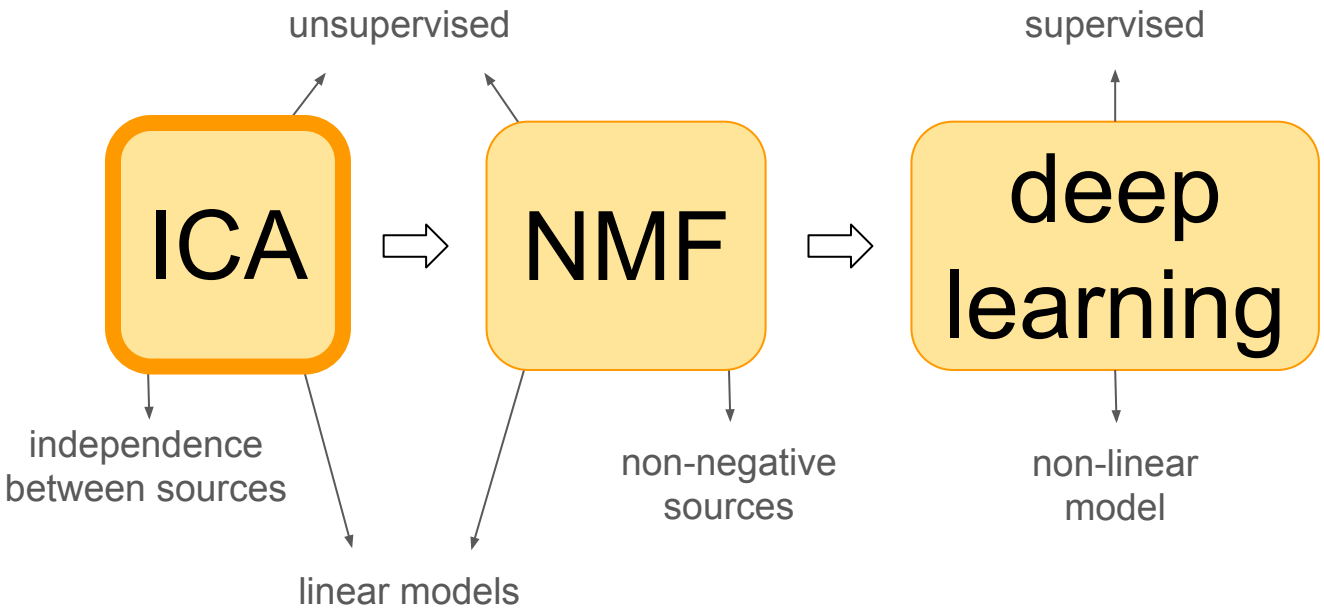
End-to-end music source separation



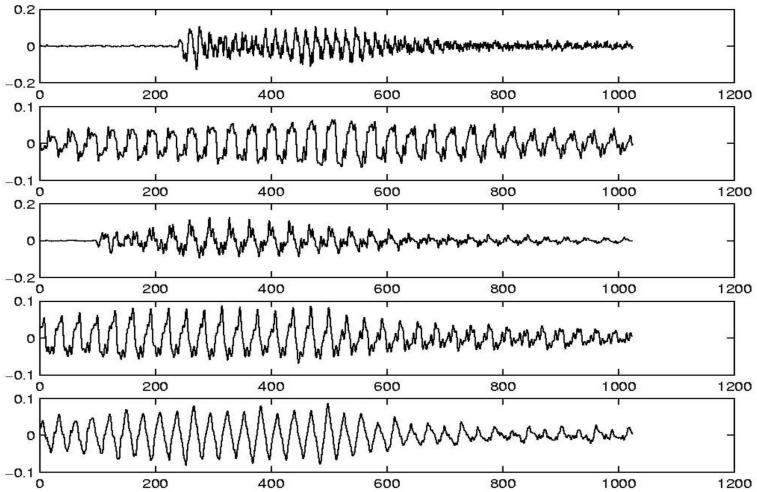
Historical perspective: waveform-based models?



Historical perspective: waveform-based models?



waveform-based ICA



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

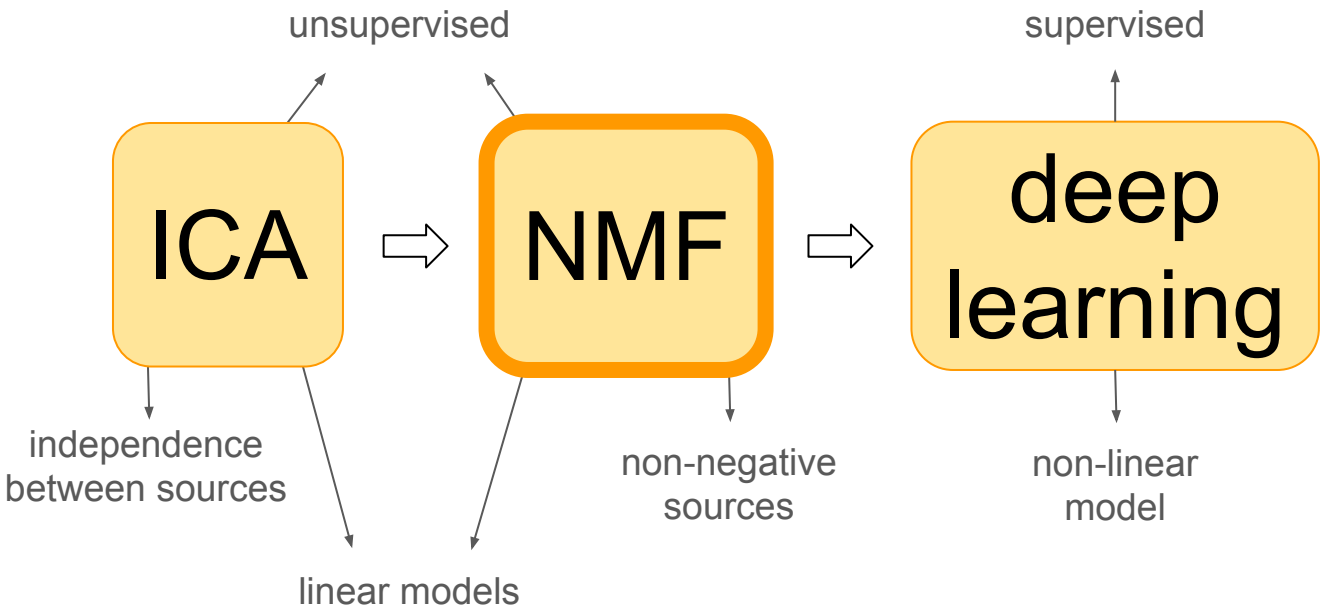
bases

activations

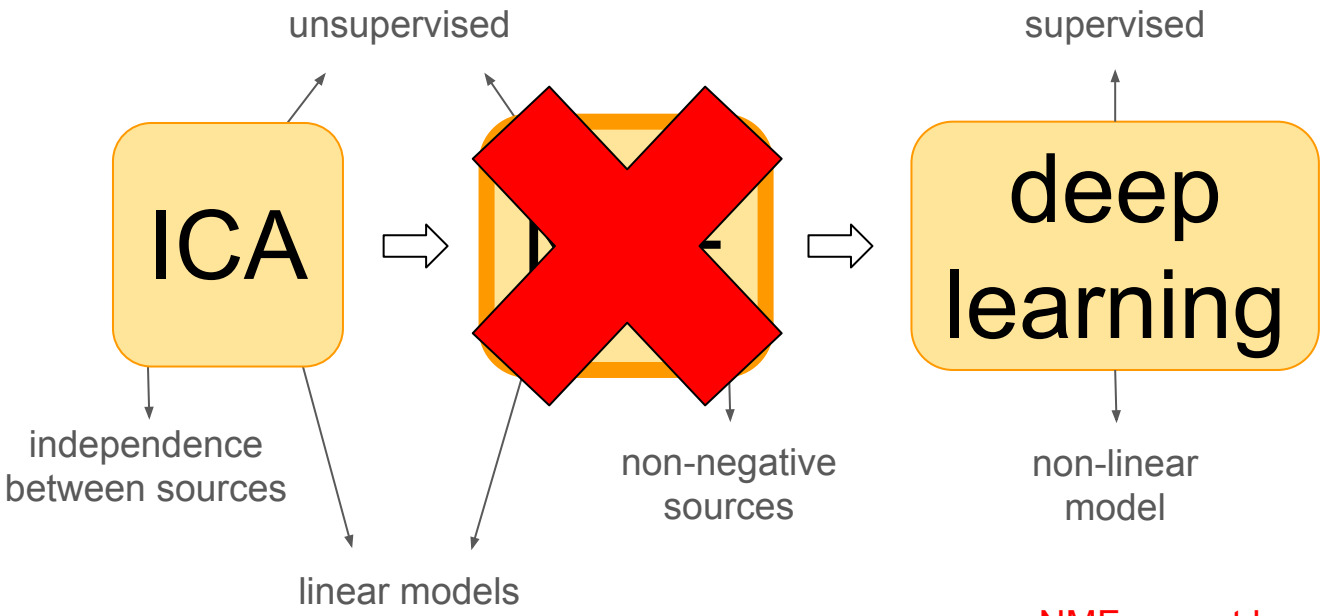
Problem 1: phase sensitive basis
Problem 2: simplicity of the linear model

Figure from: Blumensath and Davies, 2004. “Unsupervised learning of sparse and shift-invariant decompositions of polyphonic music,” in ICASSP.

Historical perspective: waveform-based models?

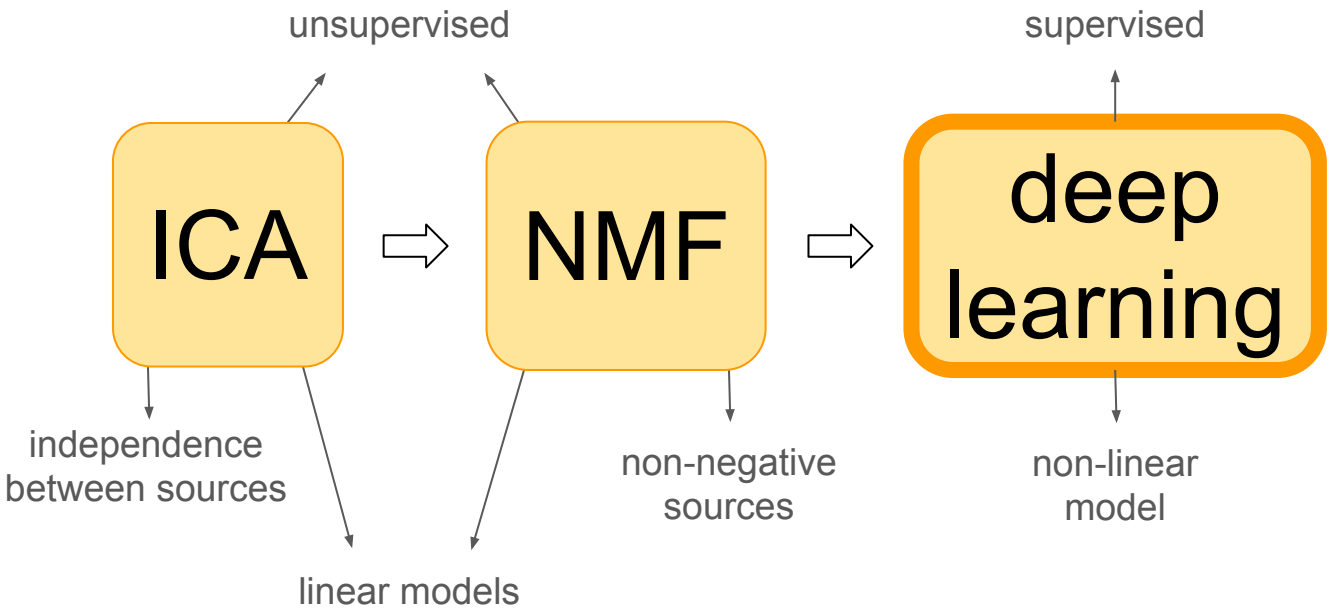


Historical perspective: waveform-based models?



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.

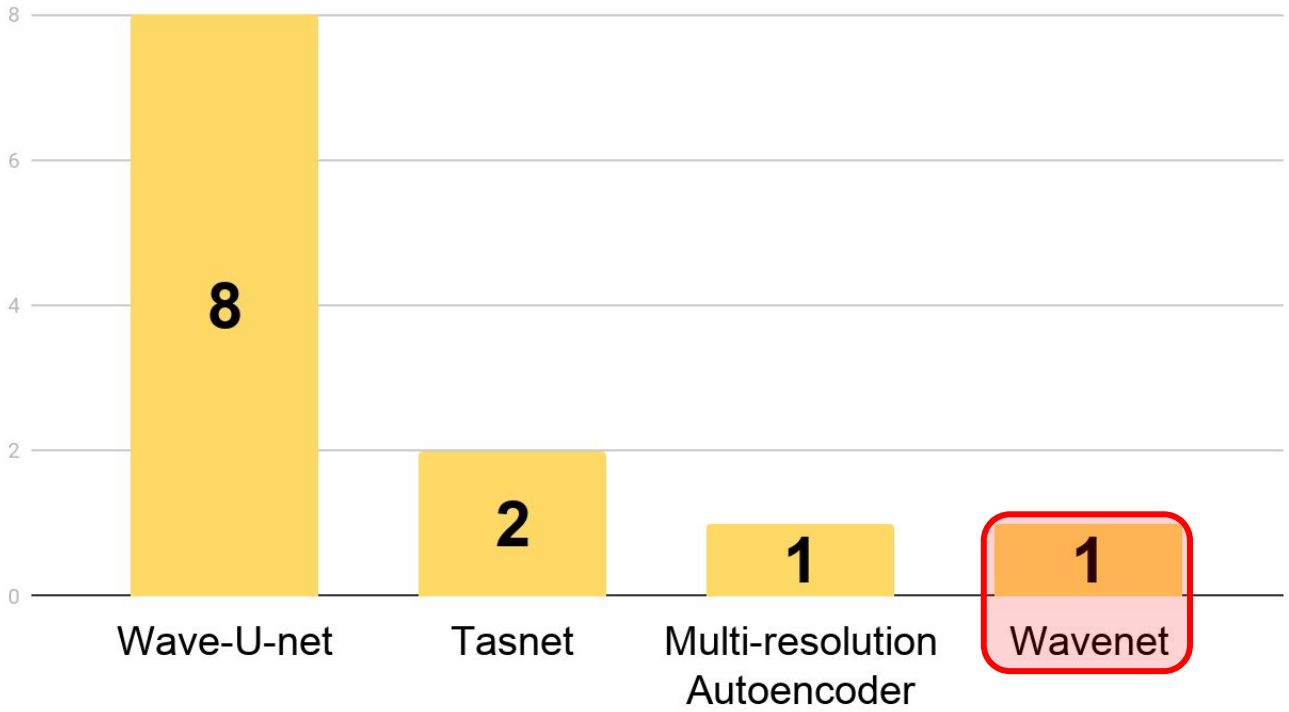
Défossez et al., 2019. “Music Source Separation in the Waveform Domain” in arXiv.

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

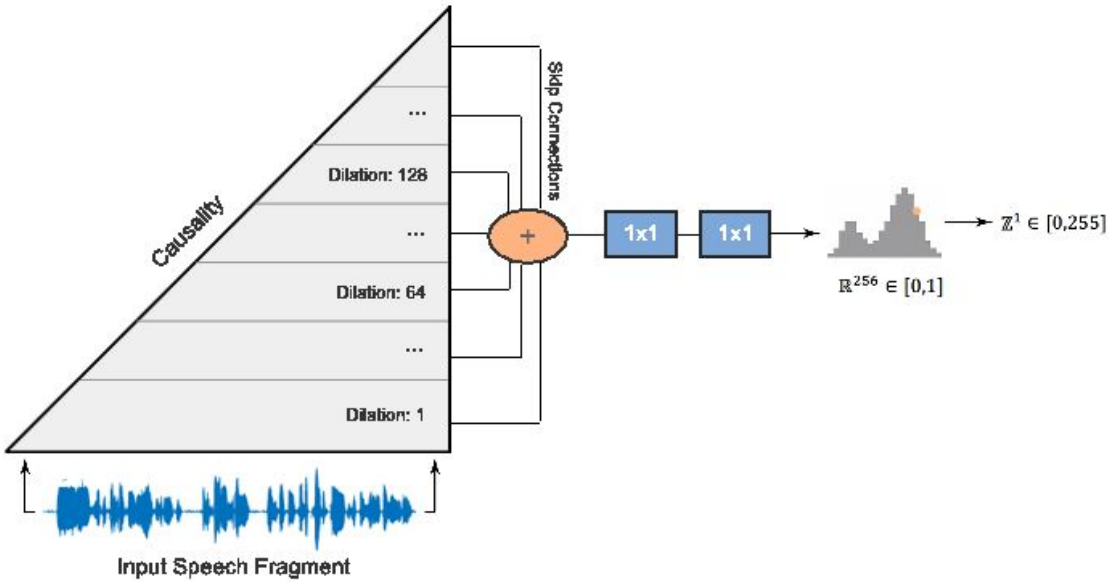
Nakamura et al., 2020. “Time-domain audio source separation based on wave-u-net combined with discrete wavelet transform” in ICASSP.

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

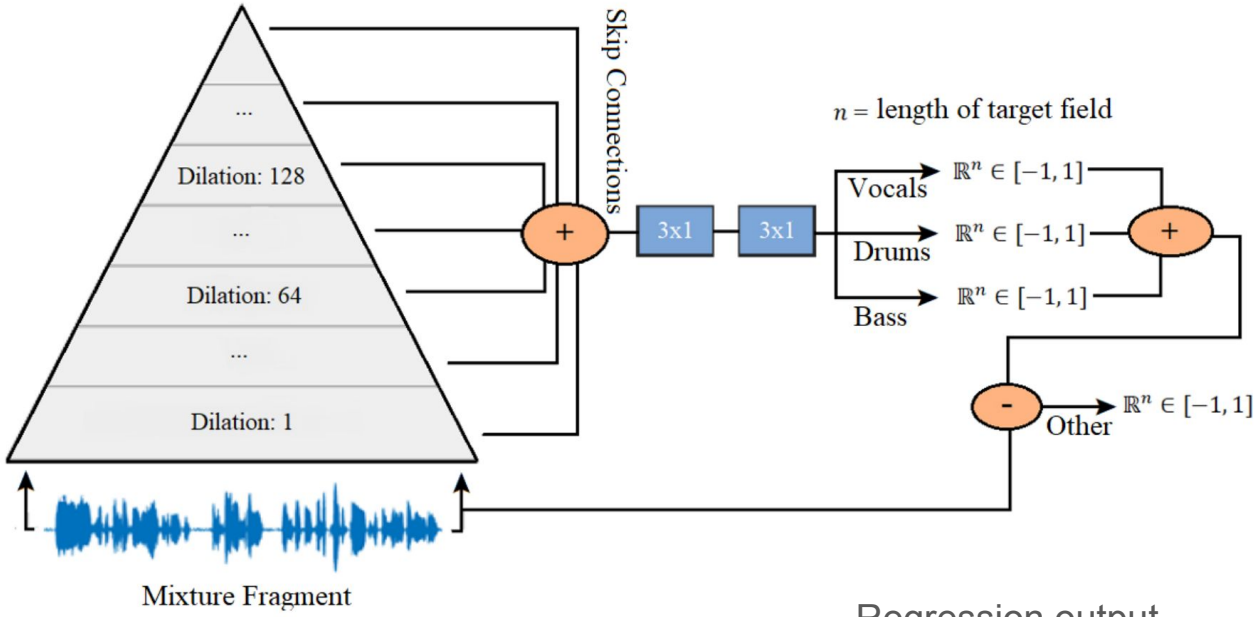


Causal

Softmax-output distribution
modeling p(x)

Van den Oord et al., 2016. “Wavenet: a generative model for audio” in arXiv.

A “regression” Wavenet for music source separation

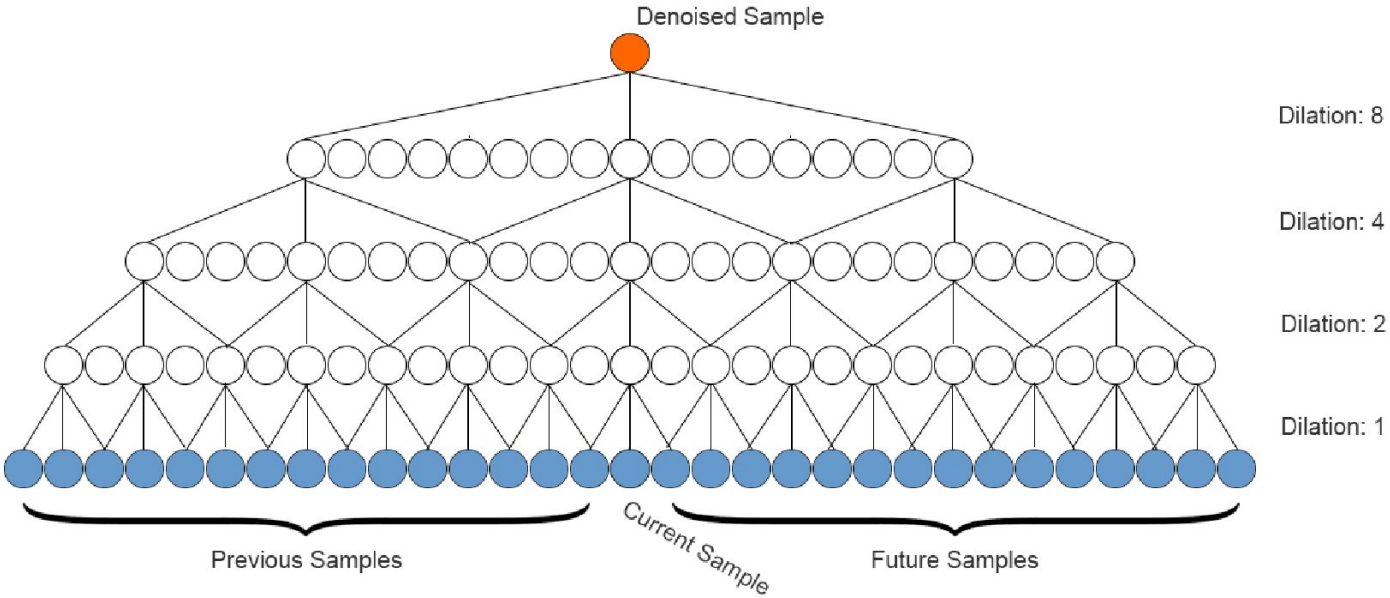


Non-causal

Regression output
p(y|x) discriminative post-processing!

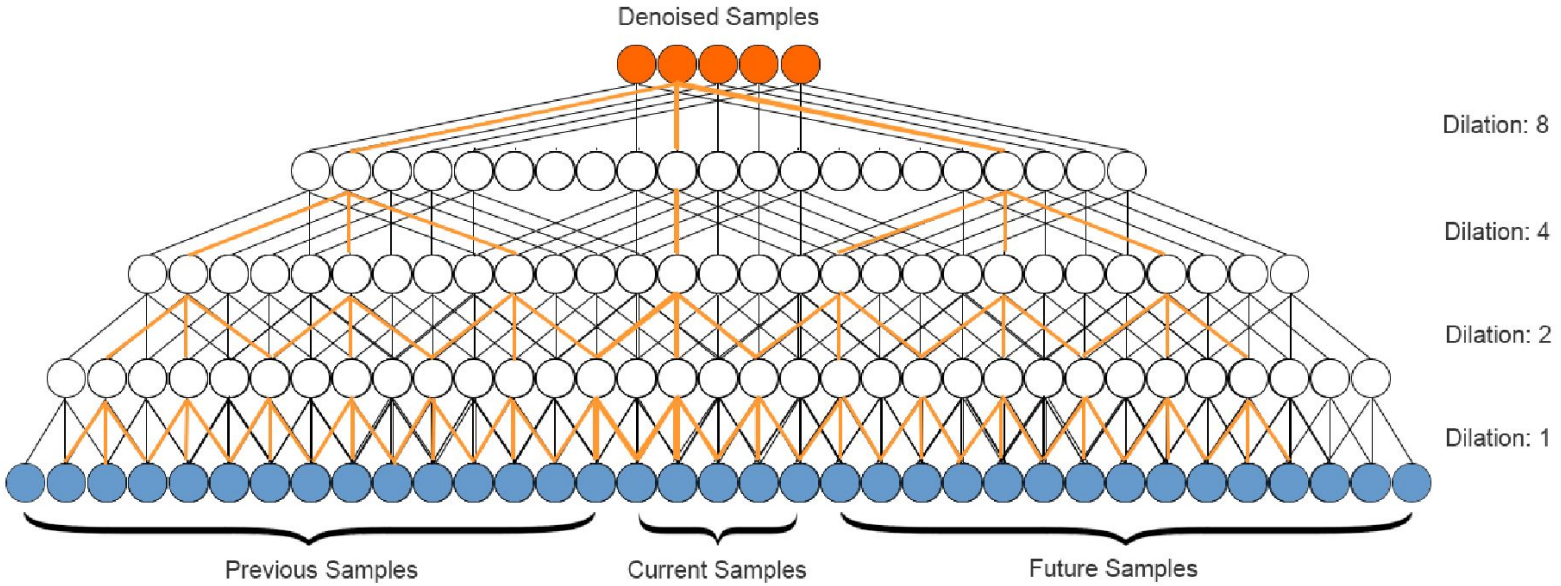
Lluis, et al., 2019. “End-to-end music source separation: is it possible in the waveform domain?” in Interspeech.

Fully convolutional & deterministic



Lluis, et al., 2019. “End-to-end music source separation: is it possible in the waveform domain?” in Interspeech.

Fully convolutional & deterministic

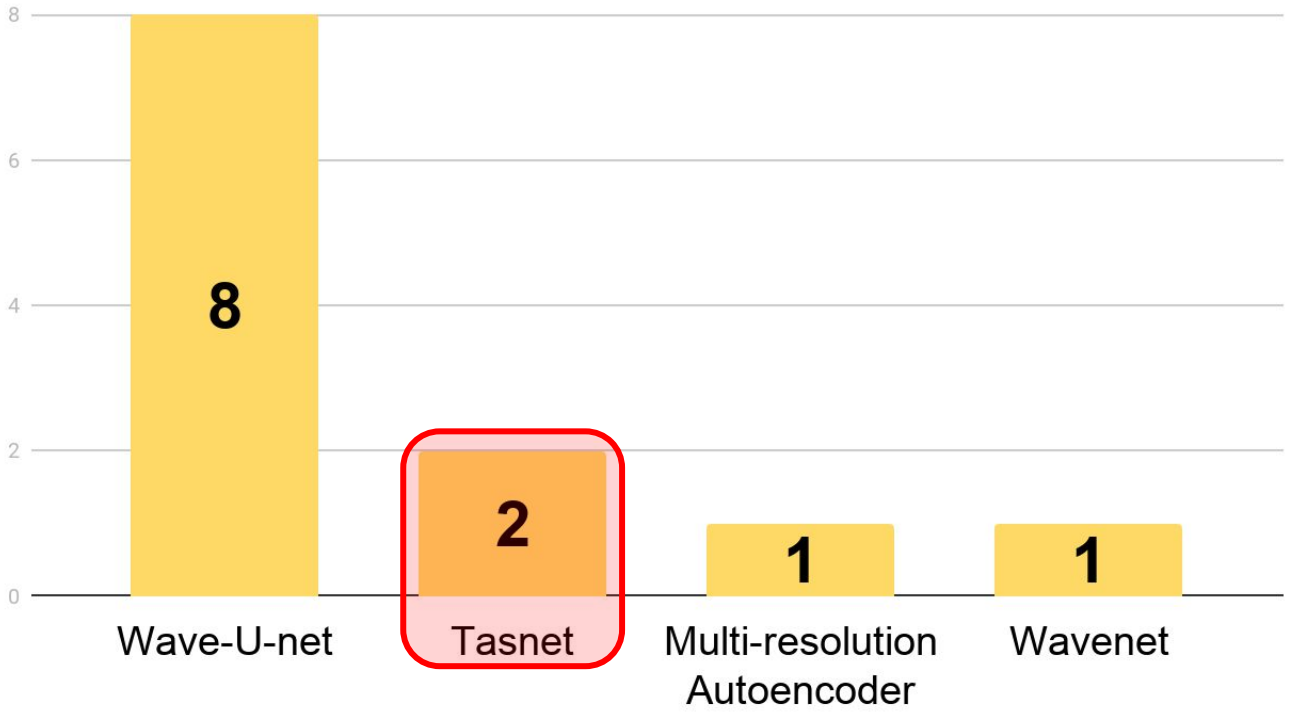


Real time inference!

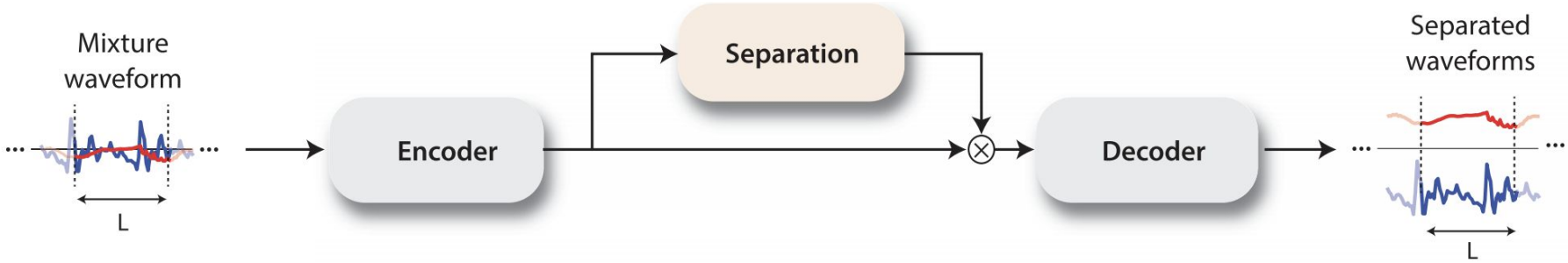
1601 samples input ⟶ denoising time: ≈ 0.56 sec per second of music on GPU!

Lluis, et al., 2019. “End-to-end music source separation: is it possible in the waveform domain?” in Interspeech.

End-to-end music source separation: architectures

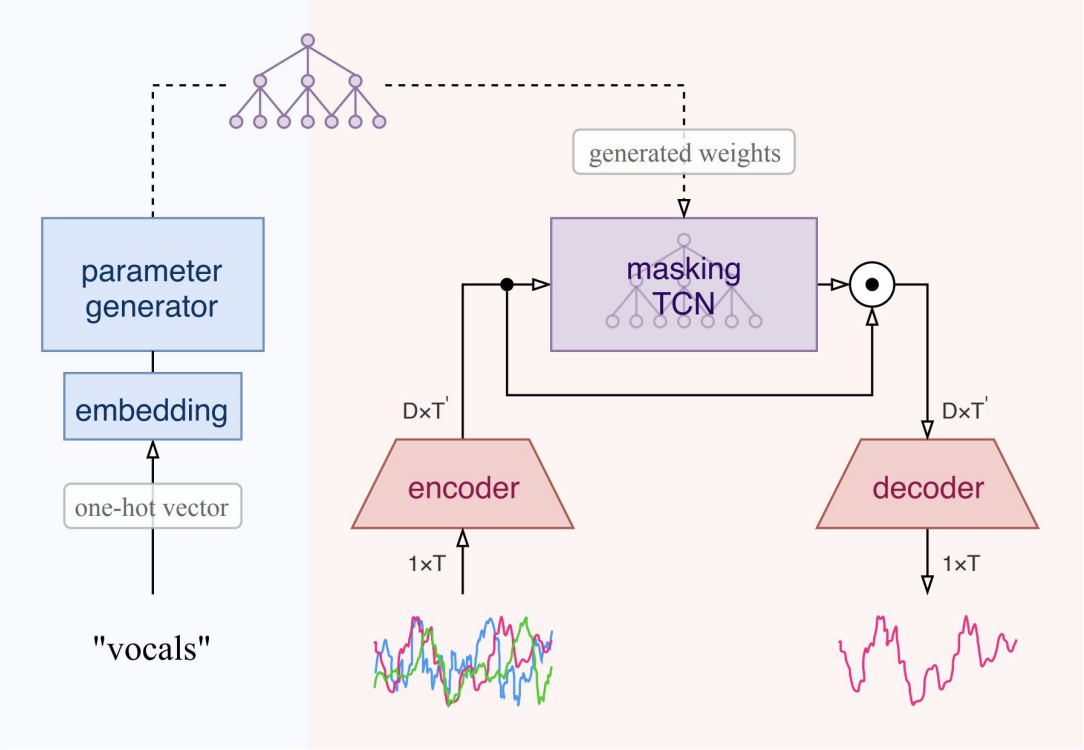


TasNet: encoder + separator + decoder



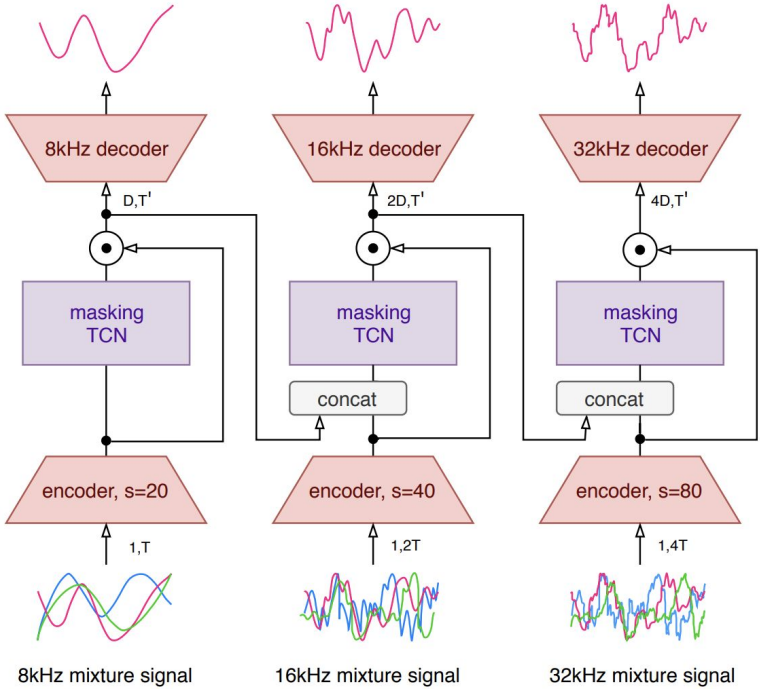
Défossez, et al., 2019. “Music source separation in the waveform domain” in arxiv.
Luo, et al. 2018. “Tasnet: time-domain audio separation network for real-time, single-channel speech separation” in ICASSP.

Separator: metalearning with TasNet



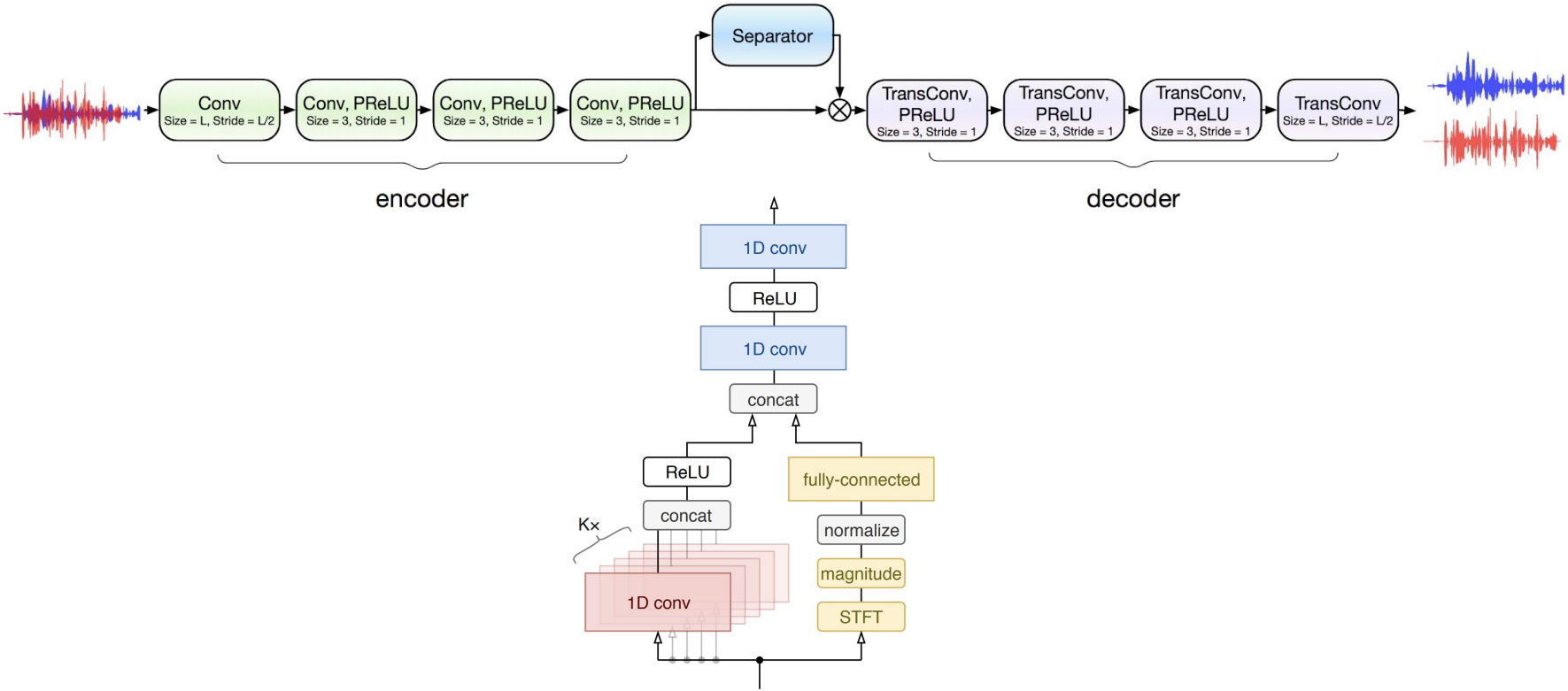
Samuel et al., 2019. “Meta-learning Extractors for Music Source Separation” in ICASSP.

Multi-stage TasNet



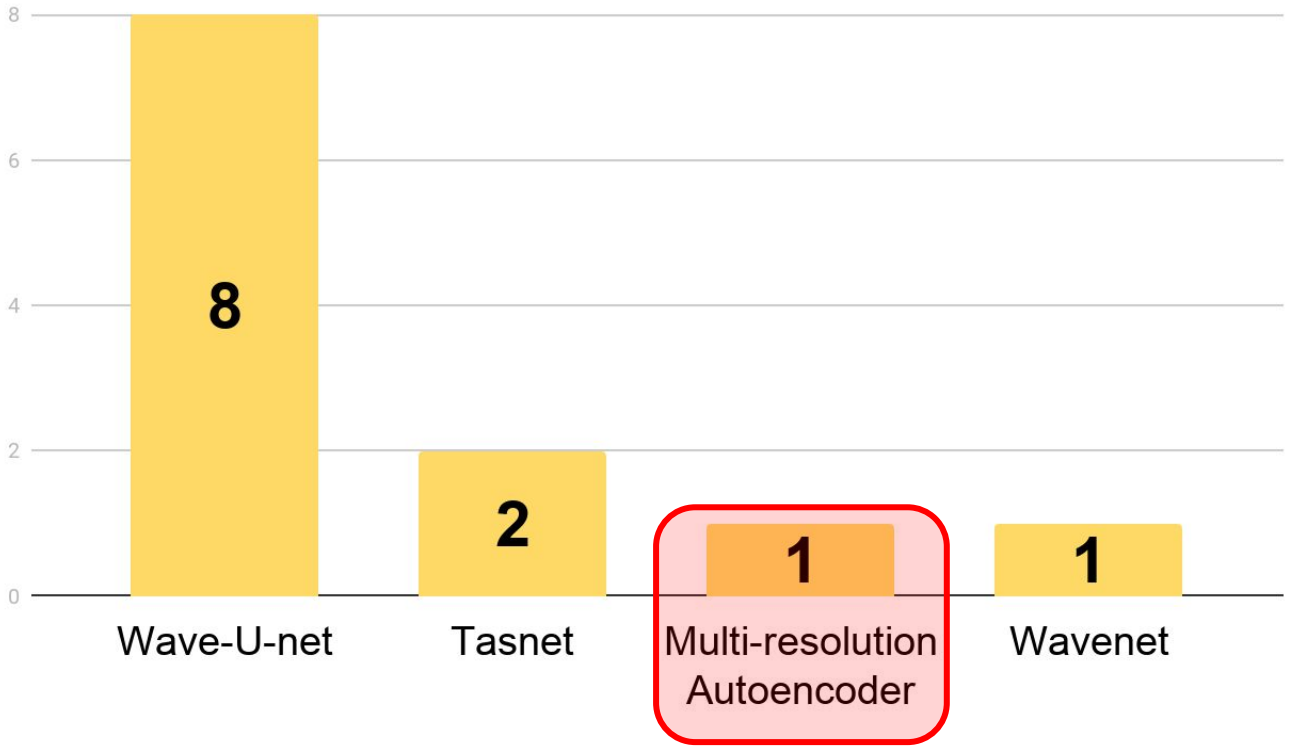
Samuel et al., 2019. “Meta-learning Extractors for Music Source Separation” in ICASSP.

“Fancy” encoders-decoders

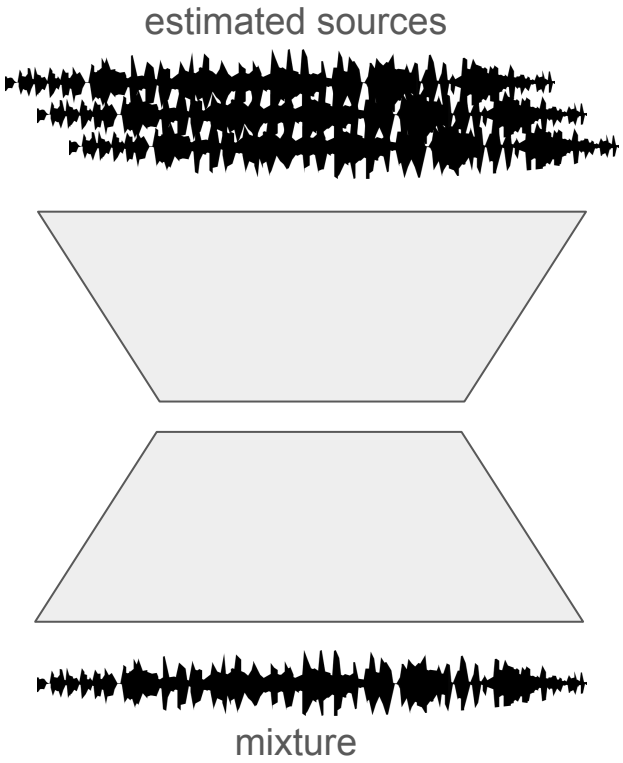


Samuel et al., 2020. “Meta-learning Extractors for Music Source Separation” in ICASSP.
Kadıoglu et al., 2020. “An empirical study of Conv-TasNet” in ICASSP.

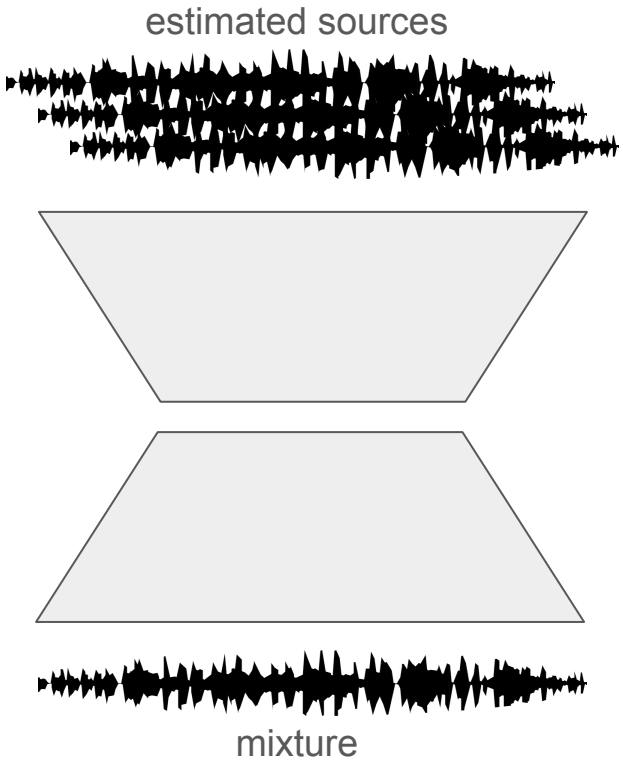
End-to-end music source separation: architectures

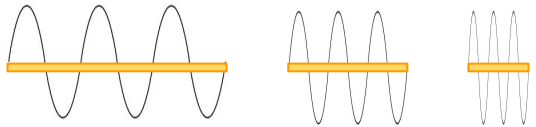


Autoencoders



Multi-resolution & Convolutional autoencoder



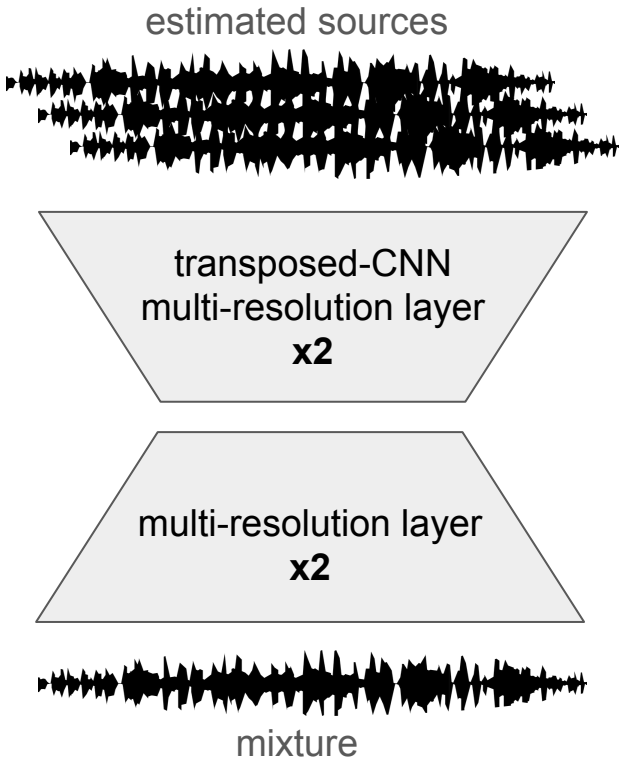


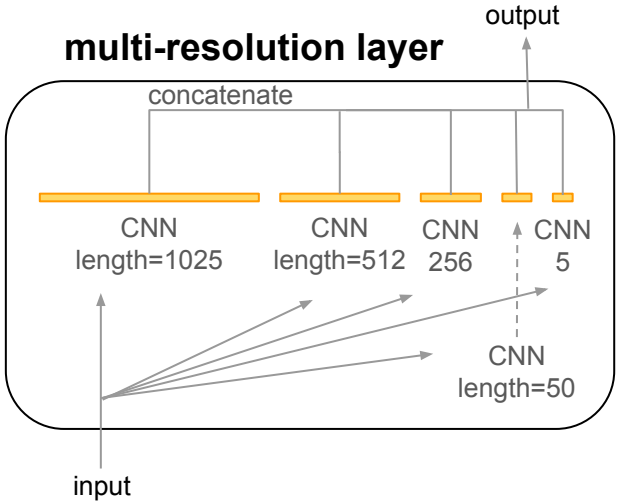
Multi-resolution CNN: efficient way to represent 3 periods!

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

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

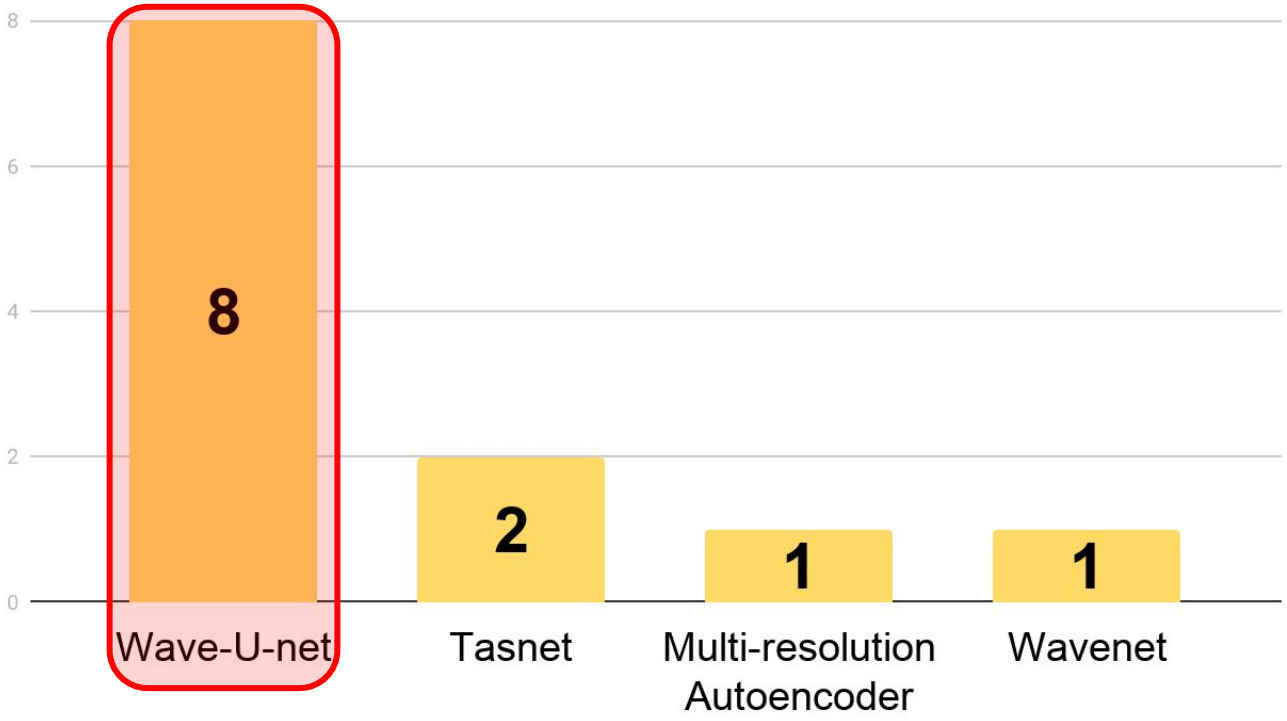
Multi-resolution & Convolutional autoencoder



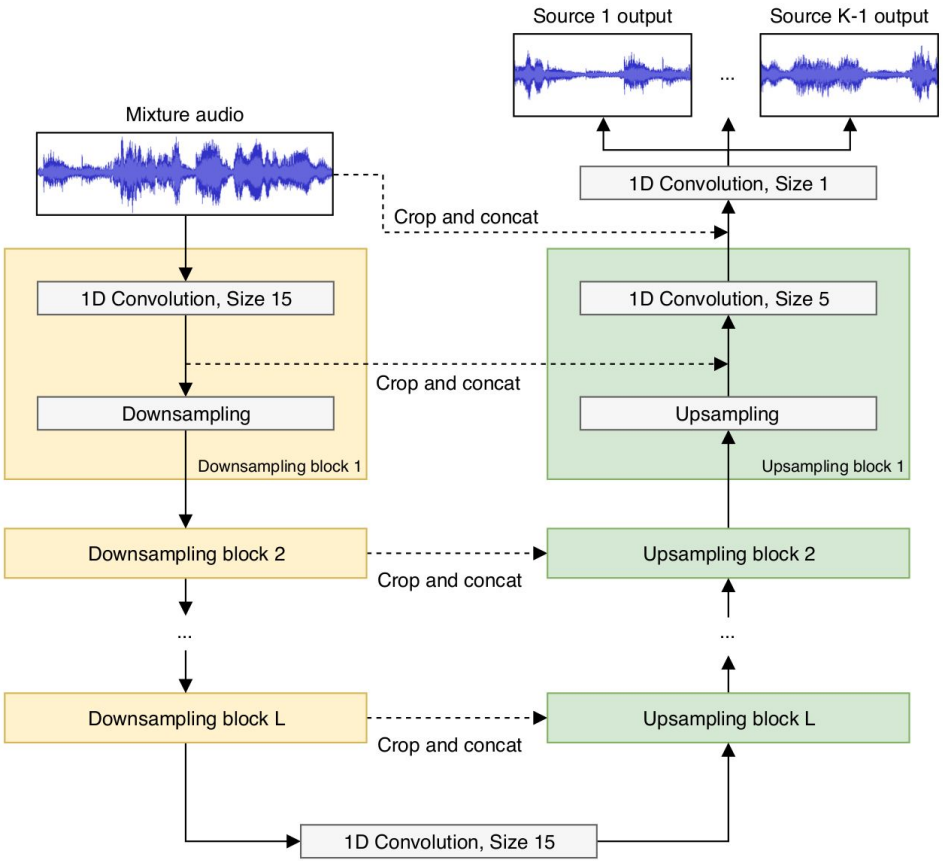


Grais et al., 2018. “Raw Multi-Channel Audio Source Separation using Multi-Resolution Convolutional Auto-Encoders” in EUSIPCO.

End-to-end music source separation: architectures



Wave-U-net



Stoller et al., 2018. “Wave-u-net: A multi-scale neural network for end-to-end audio source separation” 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: ≈ 1 dB SDR improvement.

  Cohen-Hadria et al., 2019. “Improving singing voice separation using Deep U-Net and Wave-U-Net with data augmentation” 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: ≈ 1 dB SDR improvement.

  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.

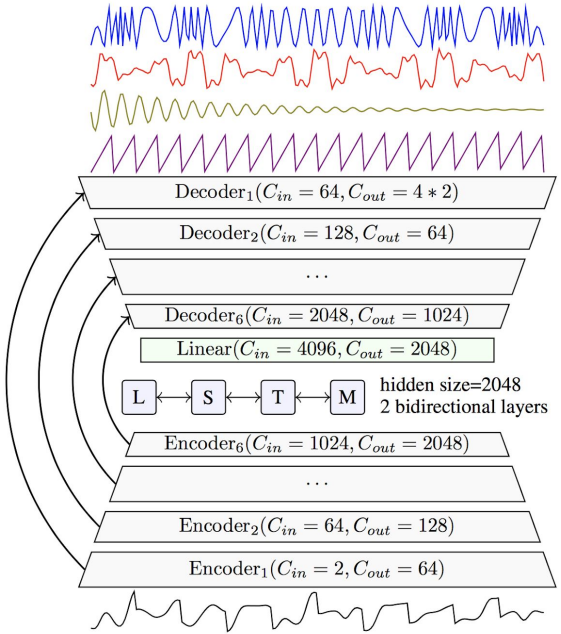
- Architectural changes:
  - Add BiLSTMs at the bottleneck: ≈ 1 dB SDR improvement.
    Kaspersen, 2019. “HydraNet: A Network For Singing Voice Separation”. Master Thesis.
  - Use dilated convolutions and dense CNNs.
    Narayanaswamy et al., 2019. “Audio Source Separation via Multi-Scale Learning with 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.

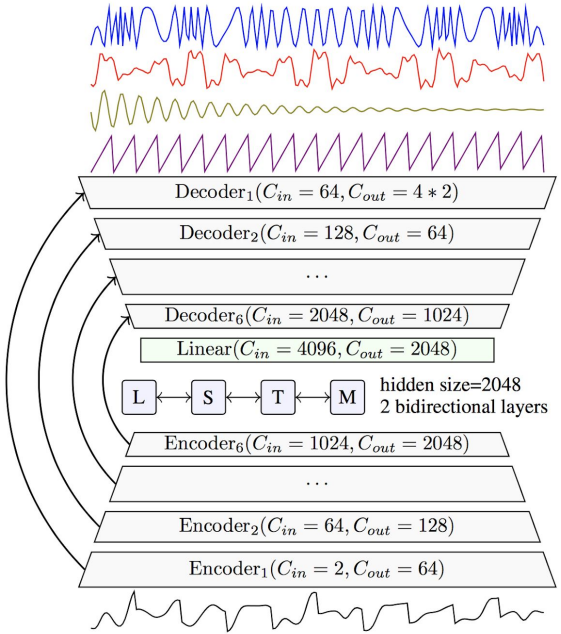
  w/ BiLSTMs at the bottleneck, data augmentation, and some additional architectural changes: ≈ 1.6 dB SDR improvement.

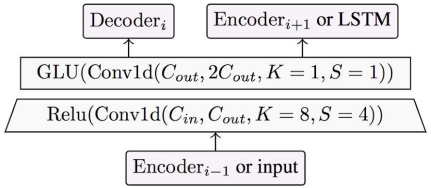
  Défossez et al., 2019. “Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed” in arXiv.

Wave-u-net extensions: Demucs

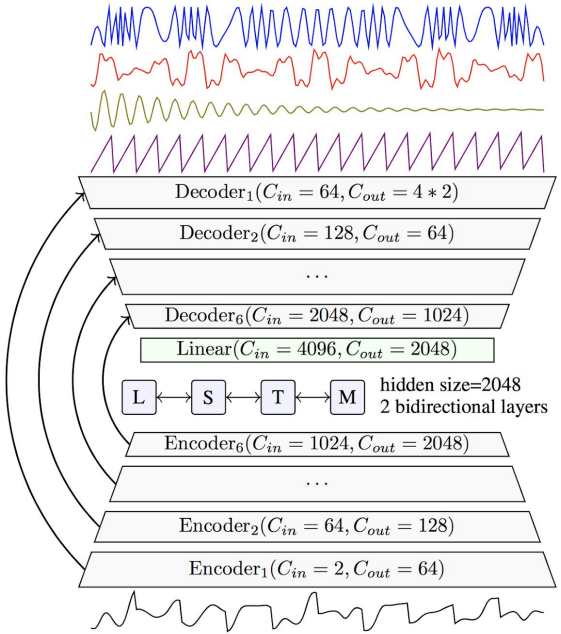


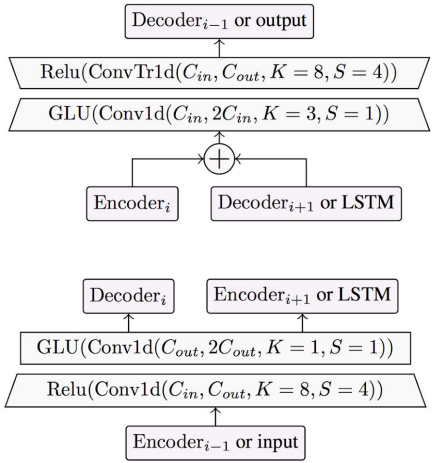
Wave-u-net extensions: Demucs





Wave-u-net extensions: Demucs



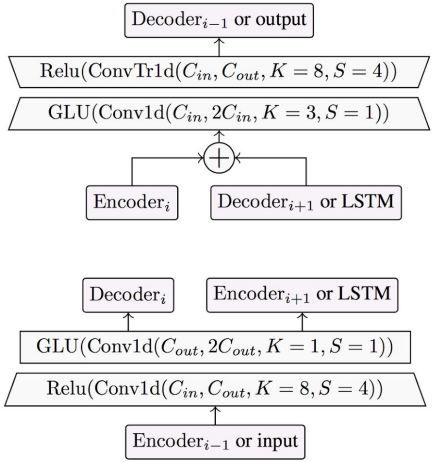


Wave-u-net extensions: Wave-U-net vs. Demucs

Block	Operation	Shape
	Input	$(16384, 1)$
DS, repeated for $i = 1, \ldots, L$	`Conv1D`$(F_c \cdot i, f_d)$	
	`Decimate`	$(4, 288)$
	`Conv1D`$(F_c \cdot (L+1), f_d)$	$(4, 312)$
US, repeated for $i = L, \ldots, 1$	`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

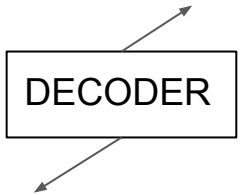


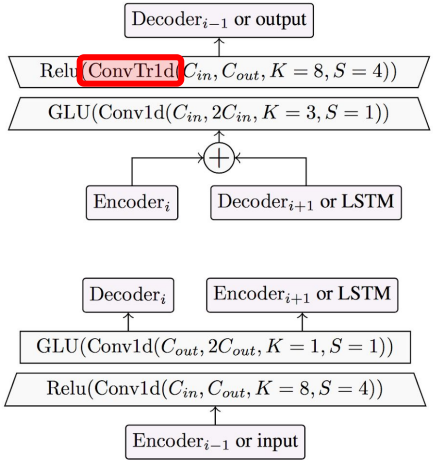
Demucs: building blocks

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

Block	Operation	Shape
	Input	$(16384, 1)$
DS, repeated for $i = 1, \ldots, L$	`Conv1D`$(F_c \cdot i, f_d)$	
	`Decimate`	$(4, 288)$
	`Conv1D`$(F_c \cdot (L+1), f_d)$	$(4, 312)$
US, repeated for $i = L, \ldots, 1$	`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

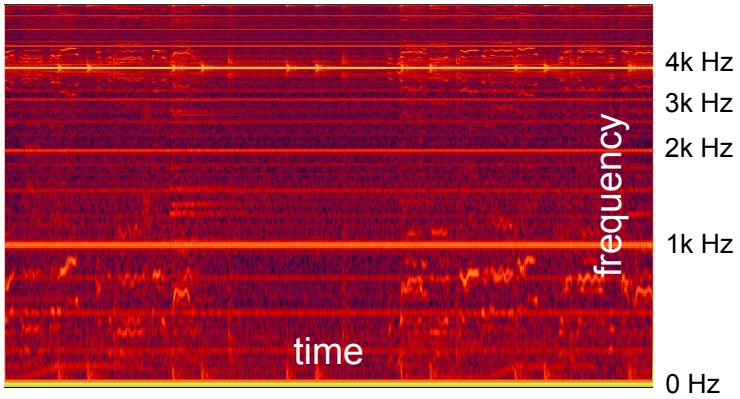




Demucs: building blocks

Deconvolutions and high-frequency artifacts

Checkerboard artifacts in images



High-frequency buzz in audio

Odena et al., 2016. "Deconvolution and Checkerboard Artifacts" in Distill.

Other key Demucs features?

Other key Demucs features?

Difference	Valid set L1 loss	Test set SDR
no initial weight rescaling	0.172	4.94
no BiLSTM	0.175	5.12
ReLU instead of GLU	0.177	5.19
MSE loss	N/A	5.55
Reference	0.164	5.58

OUT = IN / 20

Output 20x smaller than input!

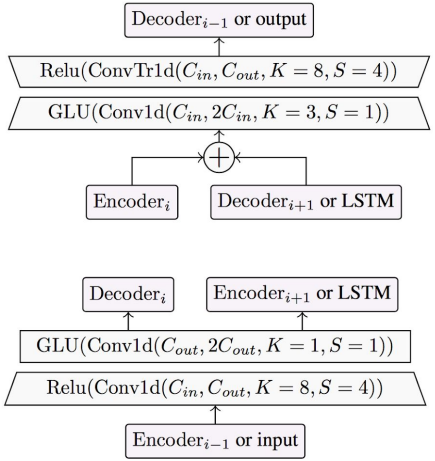
Reescale to allow it to be the same scale!

Other key Demucs features?

Difference	Valid set L1 loss	Test set SDR
no initial weight rescaling	0.172	4.94
no BiLSTM	0.175	5.12
ReLU instead of GLU	0.177	5.19
MSE loss	N/A	5.55
Reference	0.164	5.58

Other key Demucs features?

Difference	Valid set L1 loss	Test set SDR
no initial weight rescaling	0.172	4.94
no BiLSTM	0.175	5.12
ReLU instead of GLU	0.177	5.19
MSE loss	N/A	5.55
Reference	0.164	5.58

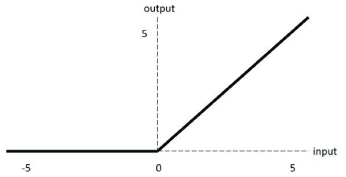


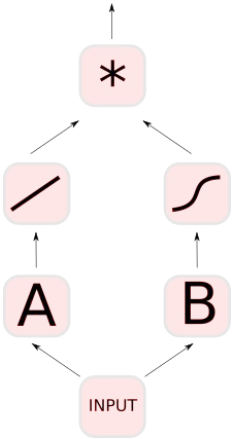
Other key Demucs features?

Difference	Valid set L1 loss	Test set SDR
no initial weight rescaling	0.172	4.94
no BiLSTM	0.175	5.12
ReLU instead of GLU	0.177	5.19
MSE loss	N/A	5.55
Reference	0.164	5.58

ReLU vs. GLU

GLU: Gated Linear Unit





Other key Demucs features?

Difference	Valid set L1 loss	Test set SDR
no initial weight rescaling	0.172	4.94
no BiLSTM	0.175	5.12
ReLU instead of GLU	0.177	5.19
MSE loss	N/A	5.55
Reference	0.164	5.58

Evaluation metrics: SDR, SIR, SAR

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

"overall performance"

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

"interference from other sources"

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

"algorithmic artifacts"

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

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

Vincent et al., 2006. "Performance measurement in blind audio source separation" in IEEE TASLP.

Which architectures seem to work the best?

Which architectures seem to work the best?

Model	Domain	Extra data?	Overall SDR	MOS Quality	MOS Contamination
Open-Unmix	spectrogram	no	5.3	3.0	3.3

Which architectures seem to work the best?

Model	Domain	Extra data?	Overall SDR	MOS Quality	MOS Contamination
Open-Unmix	spectrogram	no	5.3	3.0	3.3
Wave-U-Net	waveform	no	3.2	-	-

Which architectures seem to work the best?

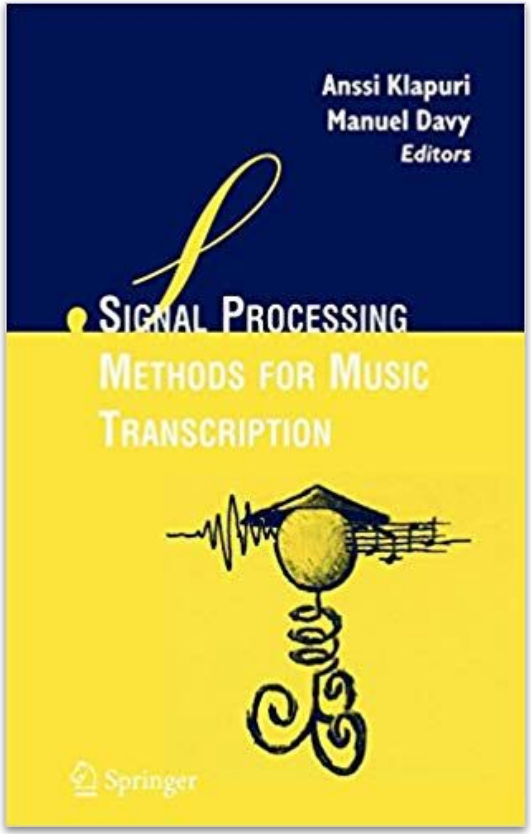
Model	Domain	Extra data?	Overall SDR	MOS Quality	MOS Contamination
Open-Unmix	spectrogram	no	5.3	3.0	3.3
Wave-U-Net	waveform	no	3.2	-	-
Demucs (this)	waveform	no	5.6	**3.2**	3.3

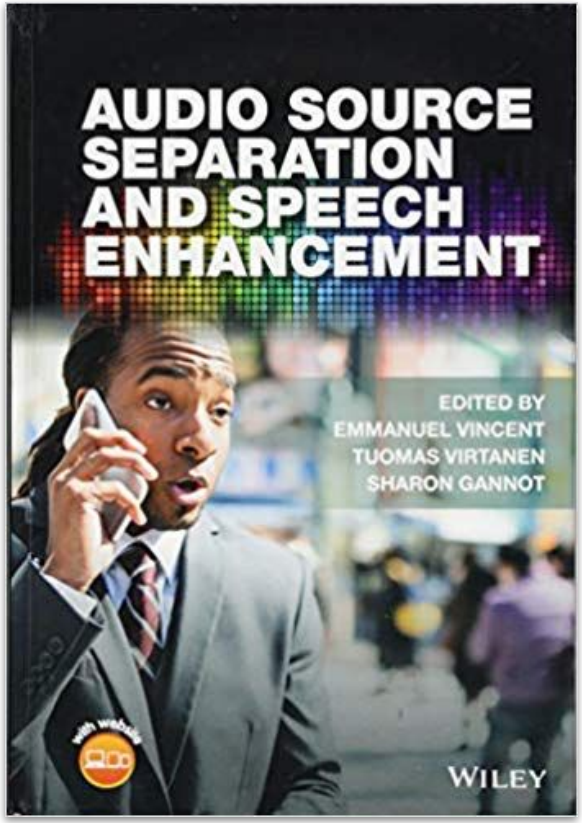
Which architectures seem to work the best?

Model	Domain	Extra data?	Overall SDR	MOS Quality	MOS Contamination
Open-Unmix	spectrogram	no	5.3	3.0	3.3
Wave-U-Net	waveform	no	3.2	-	-
Demucs (this)	waveform	no	5.6	**3.2**	3.3
Conv-Tasnet (this)	waveform	no	**5.7**	2.9	**3.4**

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**

Additional references





Estefanía Cano, Derry FitzGerald, Antoine Liutkus,
Mark D. Plumbley, and Fabian-Robert Stöter

Musical Source Separation

An introduction

Digital Object Identifier 10.1109/MSP.2018.2874719
Date of publication: 24 December 2018

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 reverberation (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.

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

*Francesc Lluís** *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 spectrogram as input, and are therefore by default omitting part of the signal: the phase. To avoid omitting potentially 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 spectrogram-based deep learning model. Namely: a Wavenet-based model we propose and Wave-U-Net can outperform DeepConvSep, a recent spectrogram-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 spectrogram 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 spectrogram 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[1] 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][2].

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 waveform 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][2], 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

*Contributed equally.

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

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

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

Music source separation with deep learning

Jordi Pons / www.jordipons.me / @jordiponsdotme