Machine learning for music separation

Combining data-driven models and expert knowledge

Paul Magron, Researcher - INRIA Centre at Université de Lorraine

IRMA, Strasbourg - May 14th, 2025

MULTISPEECH

Music separation

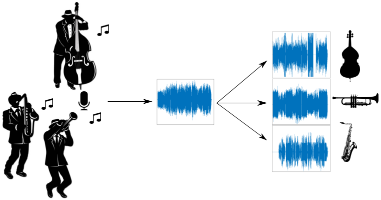
Music signals are composed of several instrumental tracks (*sources*) that add up together.

Music separation

Music signals are composed of several instrumental tracks (*sources*) that add up together.

Source separation or Demixing = recovering the sources from the mixture.

- An important preprocessing for many downstream tasks.
  - Automatic music transcription.
  - Music information retrieval.
- A goal in itself for synthesis purposes.
  - Augmented mixing, e.g., from mono to stereo.
  - Backing track generation / karaoke.

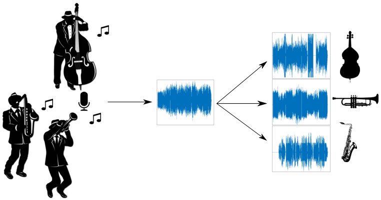


Music separation

Music signals are composed of several instrumental tracks (*sources*) that add up together.

Source separation or Demixing = recovering the sources from the mixture.

- An important preprocessing for many downstream tasks.
  - Automatic music transcription.
  - Music information retrieval.
- A goal in itself for synthesis purposes.
  - Augmented mixing, e.g., from mono to stereo.
  - Backing track generation / karaoke.



Beyond music:

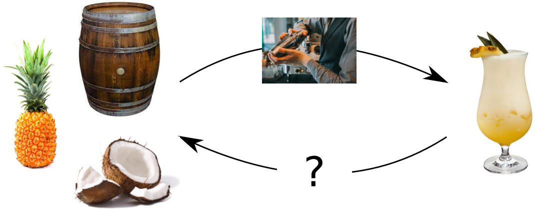
- Speech enhancement, speaker separation.
- Ambient / environmental sound analysis.
- Biomedical signals, astronomy imaging, fluorescence spectroscopy, etc.

A difficult task?

Mixing is easy . . .

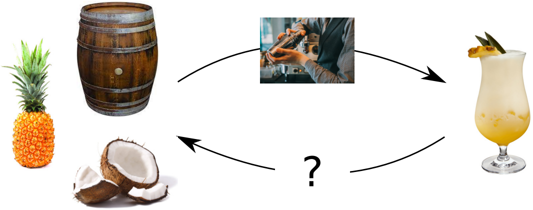
A difficult task?

Mixing is easy . . . but demixing is not.



A difficult task?

Mixing is easy . . . but demixing is not.

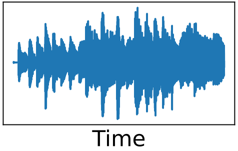


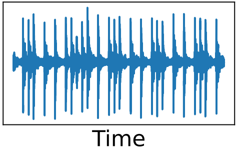
Finding $\{s_j\}_{j=1}^{J}$ such that $x = \sum_{j=1}^{J} s_j$ is an under-determined problem.

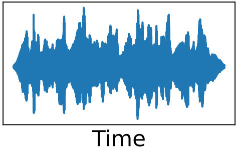
- Need to incorporate additional information / constraint / structure.
- Either via expert knowledge or by leveraging data.

Setting the stage

- The raw material: audio signals.

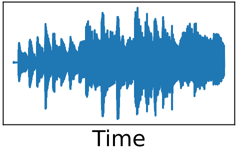


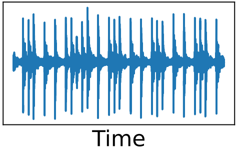


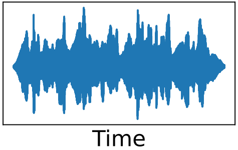


Setting the stage

▷ The raw material: audio signals.



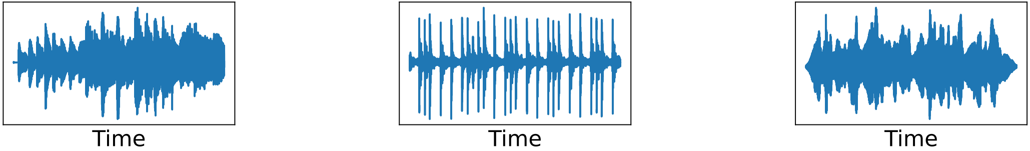




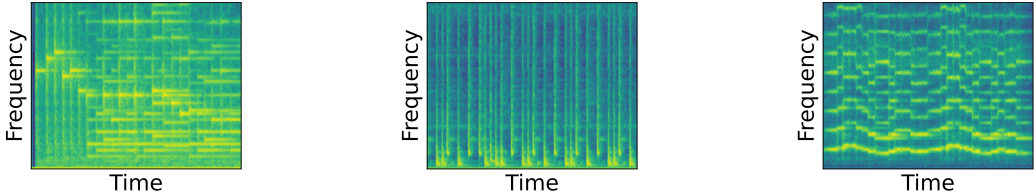
▷ It's hard to see structure there...

Setting the stage

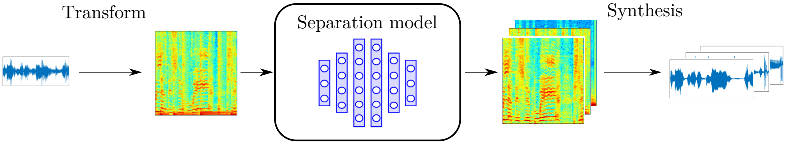
- The raw material: audio signals.



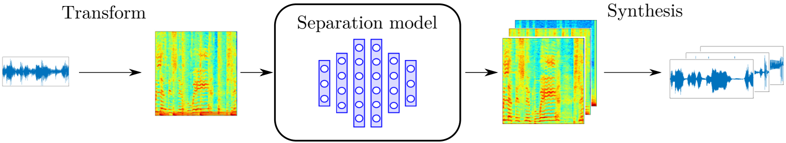
- It's hard to see structure there...
- We rather transform them into a time-frequency representation, e.g., a spectrogram.



The separation pipeline

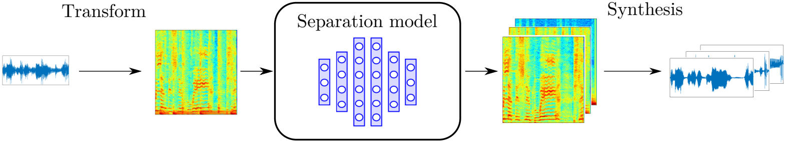


The separation pipeline



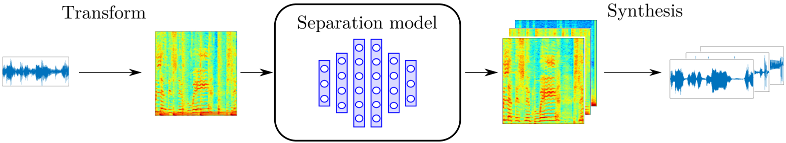
- ▷ The transform is usually the short-time Fourier transform (STFT).

The separation pipeline



- The transform is usually the short-time Fourier transform (STFT).
- The separator is based on:
  - Earlier approaches (independent / principal component analysis).
  - Nonnegative matrix factorization (NMF).
  - Deep neural networks (DNNs).

The separation pipeline



- The transform is usually the short-time Fourier transform (STFT).
- The separator is based on:
  - Earlier approaches (independent / principal component analysis).
  - Nonnegative matrix factorization (NMF).
  - Deep neural networks (DNNs).
- Synthesis is performed through inverse STFT.

Nonnegative matrix factorization (NMF)

Given a (nonnegative) spectrogram $\mathbf{V}$, find a factorization $\mathbf{WH}$ such that the factors $\mathbf{W}$ and $\mathbf{H}$ are:

- ▷ low rank.
- ▷ nonnegative.

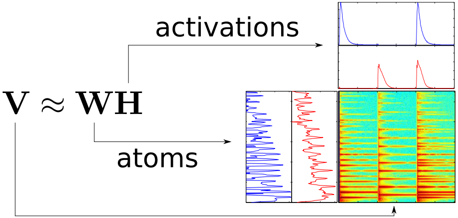
Nonnegative matrix factorization (NMF)

Given a (nonnegative) spectrogram $\mathbf{V}$, find a factorization $\mathbf{WH}$ such that the factors $\mathbf{W}$ and $\mathbf{H}$ are:

- ▷ low rank.
- ▷ nonnegative.

Nonnegativity favors interpretability.

- ▷ $\mathbf{W}$ is a dictionary of spectral atoms.
- ▷ $\mathbf{H}$ is a matrix of temporal activation.



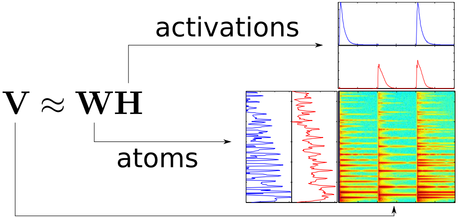
Nonnegative matrix factorization (NMF)

Given a (nonnegative) spectrogram $\mathbf{V}$, find a factorization $\mathbf{WH}$ such that the factors $\mathbf{W}$ and $\mathbf{H}$ are:

- ▷ low rank.
- ▷ nonnegative.

Nonnegativity favors interpretability.

- ▷ $\mathbf{W}$ is a dictionary of spectral atoms.
- ▷ $\mathbf{H}$ is a matrix of temporal activation.



Estimation via an optimization problem:

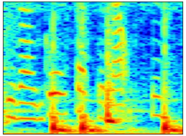
$$\min_{\mathbf{W},\mathbf{H}} D(\mathbf{V}, \mathbf{WH}) \quad + \text{regularizations}$$

- ▷ Many options for the divergence, the regularizations, the optimization technique...

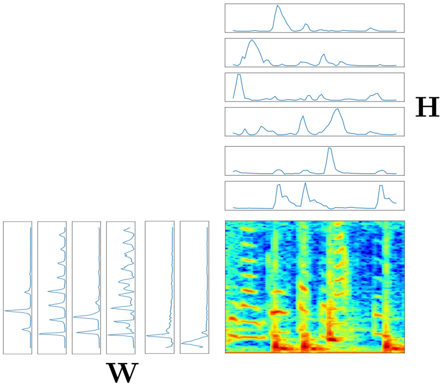
NMF for source separation

Exploit additivity for getting each source spectrogram.

$$\mathbf{V} \approx \mathbf{W}\mathbf{H} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$



NMF for source separation



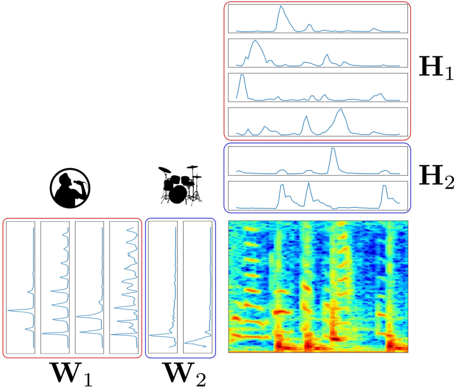
Exploit additivity for getting each source spectrogram.

$$\mathbf{V} \approx \mathbf{W}\mathbf{H} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$

Procedure

1. Factorize the mixture's spectrogram (i.e., find $\mathbf{W}$ and $\mathbf{H}$ by solving the optimization problem).

NMF for source separation



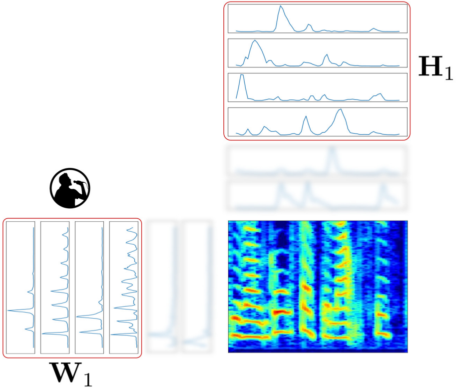
Exploit additivity for getting each source spectrogram.

$$\mathbf{V} \approx \mathbf{W}\mathbf{H} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$

Procedure

1. Factorize the mixture's spectrogram (i.e., find $\mathbf{W}$ and $\mathbf{H}$ by solving the optimization problem).
2. Cluster atoms $\mathbf{w}_k$ that belong to the same source to build source-specific matrices: $\mathbf{W}_j = \{\mathbf{w}_k\}_{k \in \mathcal{K}_j}$

NMF for source separation



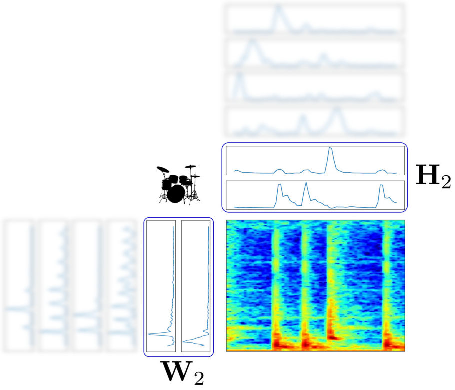
Exploit additivity for getting each source spectrogram.

$$\mathbf{V} \approx \mathbf{W}\mathbf{H} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$

Procedure

1. Factorize the mixture's spectrogram (i.e., find $\mathbf{W}$ and $\mathbf{H}$ by solving the optimization problem).
2. Cluster atoms $\mathbf{w}_k$ that belong to the same source to build source-specific matrices: $\mathbf{W}_j = \{\mathbf{w}_k\}_{k \in \mathcal{K}_j}$
3. Multiply each dictionary with the corresponding activations to retrieve each source spectrogram.

NMF for source separation



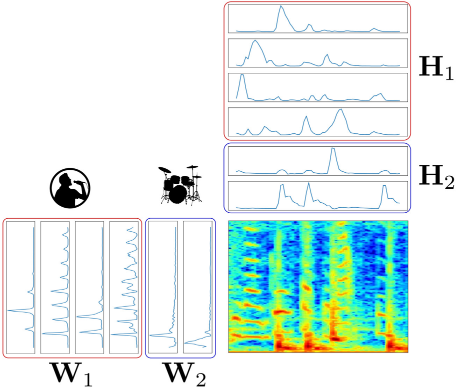
Exploit additivity for getting each source spectrogram.

$$\mathbf{V} \approx \mathbf{W}\mathbf{H} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$

Procedure

1. Factorize the mixture's spectrogram (i.e., find $\mathbf{W}$ and $\mathbf{H}$ by solving the optimization problem).
2. Cluster atoms $\mathbf{w}_k$ that belong to the same source to build source-specific matrices: $\mathbf{W}_j = \{\mathbf{w}_k\}_{k \in \mathcal{K}_j}$
3. Multiply each dictionary with the corresponding activations to retrieve each source spectrogram.

NMF for source separation



Exploit additivity for getting each source spectrogram.

$$\mathbf{V} \approx \mathbf{W}\mathbf{H} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$

Procedure

1. Factorize the mixture's spectrogram (i.e., find $\mathbf{W}$ and $\mathbf{H}$ by solving the optimization problem).
2. Cluster atoms $\mathbf{w}_k$ that belong to the same source to build source-specific matrices: $\mathbf{W}_j = \{\mathbf{w}_k\}_{k \in \mathcal{K}_j}$
3. Multiply each dictionary with the corresponding activations to retrieve each source spectrogram.

✓ Light and interpretable model.

✗ Performance is limited due to the clustering / search space is too large.

Introducing supervision data

Assume some isolated tracks containing one instrument are available.

- For each instrument j, pretrain the dictionary from it's spectrogram $\mathbf{V}_j^{\text{pretrain}}$:

$$\mathbf{W}_j^{\text{pretrain}} = \arg\min_{\mathbf{W},\mathbf{H}} D(\mathbf{V}_j^{\text{pretrain}}, \mathbf{WH})$$

Introducing supervision data

Assume some isolated tracks containing one instrument are available.

- For each instrument j, pretrain the dictionary from it's spectrogram $\mathbf{V}_j^{\text{pretrain}}$:

$$\mathbf{W}_j^{\text{pretrain}} = \operatorname*{arg\,min}_{\mathbf{W},\mathbf{H}} D(\mathbf{V}_j^{\text{pretrain}}, \mathbf{W}\mathbf{H})$$

- On the mixture, fix the dictionaries and only estimate the activation:

$$[\mathbf{H}_1, \dots, \mathbf{H}_J] = \operatorname*{arg\,min}_{\mathbf{H}} D(\mathbf{V}, [\mathbf{W}_1^{\text{pretrain}}, \dots, \mathbf{W}_J^{\text{pretrain}}]\mathbf{H})$$

Introducing supervision data

Assume some isolated tracks containing one instrument are available.

- For each instrument j, pretrain the dictionary from it's spectrogram $\mathbf{V}_j^{\text{pretrain}}$:

$$\mathbf{W}_j^{\text{pretrain}} = \arg\min_{\mathbf{W},\mathbf{H}} D(\mathbf{V}_j^{\text{pretrain}}, \mathbf{W}\mathbf{H})$$

- On the mixture, fix the dictionaries and only estimate the activation:

$$[\mathbf{H}_1, \dots, \mathbf{H}_J] = \arg\min_{\mathbf{H}} D(\mathbf{V}, [\mathbf{W}_1^{\text{pretrain}}, \dots, \mathbf{W}_J^{\text{pretrain}}]\mathbf{H})$$

- Retrieve each source's spectrogram via $\mathbf{V}_j = \mathbf{W}_j^{\text{pretrain}}\mathbf{H}_j$

Introducing supervision data

Assume some isolated tracks containing one instrument are available.

- ▷ For each instrument j, pretrain the dictionary from it's spectrogram $\mathbf{V}_j^{\text{pretrain}}$:

$$\mathbf{W}_j^{\text{pretrain}} = \underset{\mathbf{W},\mathbf{H}}{\arg\min}\, D(\mathbf{V}_j^{\text{pretrain}}, \mathbf{WH})$$

- ▷ On the mixture, fix the dictionaries and only estimate the activation:

$$[\mathbf{H}_1, \dots, \mathbf{H}_J] = \underset{\mathbf{H}}{\arg\min}\, D(\mathbf{V}, [\mathbf{W}_1^{\text{pretrain}}, \dots, \mathbf{W}_J^{\text{pretrain}}]\mathbf{H})$$

- ▷ Retrieve each source's spectrogram via $\mathbf{V}_j = \mathbf{W}_j^{\text{pretrain}}\mathbf{H}_j$

(✓) Performance is better, but still limited: low-rankness, additivity. . . .

Deep neural networks (DNNs)

Model: a mapping function f with parameters θ between inputs x and outputs y:

$$y \approx f_\theta(x)$$

- x and y are high-dimensional audio data (e.g., spectrograms).
- f_θ is built by assembling (many) *neurons* and *activation* functions ($|\theta| \sim 10^7$).

Deep neural networks (DNNs)

Model: a mapping function f with parameters θ between inputs x and outputs y:

$$y \approx f_\theta(x)$$

- x and y are high-dimensional audio data (e.g., spectrograms).
- f_θ is built by assembling (many) *neurons* and *activation* functions ($|\theta| \sim 10^7$).

Supervised learning

- Consider a collection of inputs/outputs pairs $\{x_i, y_i\}_{i=1}^{I}$ (= a *training* dataset).
- The parameters of the network are learned via:

$$\min_\theta \sum_{i=1}^{I} \mathcal{L}(y_i, f_\theta(x_i))$$

- Solved with a stochastic gradient descent algorithm (e.g., ADAM).

What is research like?

From expert knowledge research

- ▷ How do I refine this model to overcome its limitation (e.g., convolutive NMF)?
- ▷ Which regularization would fit this intrument (sparsity, (in)harmonicity)?
- ▷ How do I model it mathematically (trade-off between complexity and generalizability)?
- ▷ Which loss would be more perceptually-relevant?
- ▷ How do I (efficiently) solve the new optimization problem?

What is research like?

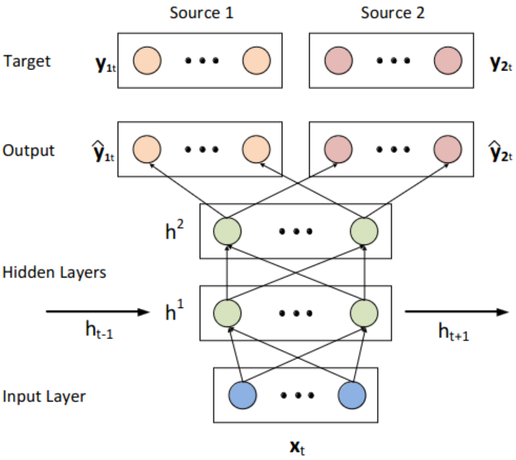
From expert knowledge research

- ▷ How do I refine this model to overcome its limitation (e.g., convolutive NMF)?
- ▷ Which regularization would fit this intrument (sparsity, (in)harmonicity)?
- ▷ How do I model it mathematically (trade-off between complexity and generalizability)?
- ▷ Which loss would be more perceptually-relevant?
- ▷ How do I (efficiently) solve the new optimization problem?

to data-driven model engineering.

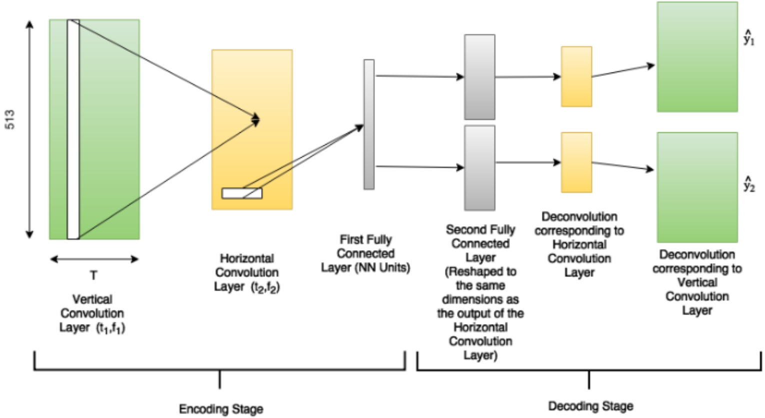
- ▷ Which architecture would be more powerful?
- ▷ Should I re-test every hyperparameter value upon a minor additional change?
- ▷ How can I parallelize / reduce training time / optimally use my hardware?
- ▷ How can I use more data / better exploit my available data / cope with data scarcity?

A few architectures



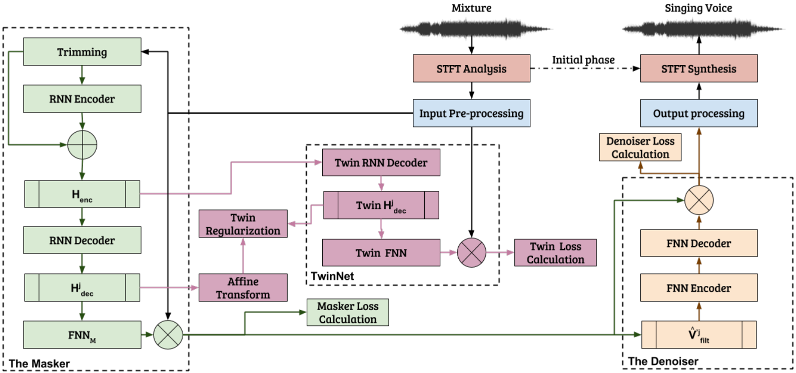
Source: Huang et al., "Deep learning for monaural speech separation", Proc. IEEE ICASSP, 2014.

A few architectures



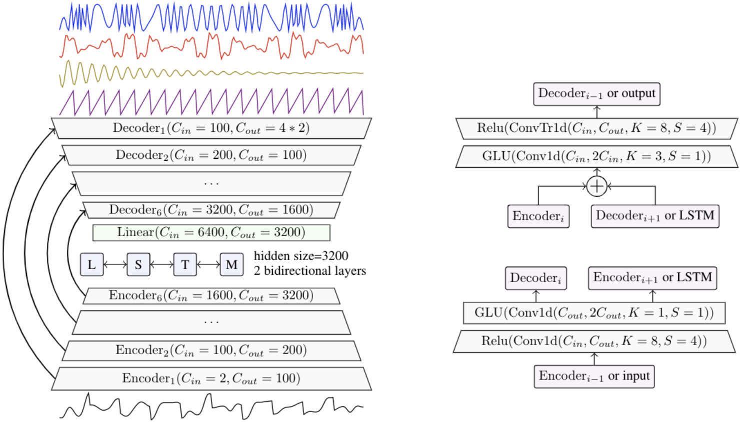
Source: Chandna et al., "Monoaural Audio Source Separation Using Deep Convolutional Neural Networks", Lecture Notes in Computer Science, 2017.

A few architectures



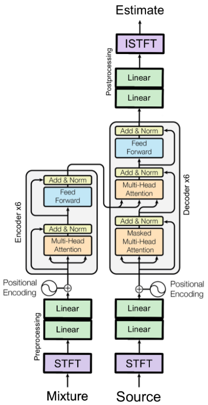
Source: Drossos et al., "MaD TwinNet: Masker-Denoiser Architecture with Twin Networks for Monaural Sound Source Separation", Proc. IEEE IJCNN, 2018.

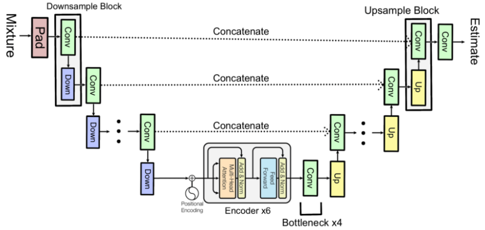
A few architectures



Source: Défossez, "Hybrid Spectrogram and Waveform Source Separation", Proc. ISMIR Workshop on Music Source Separation, 2021.

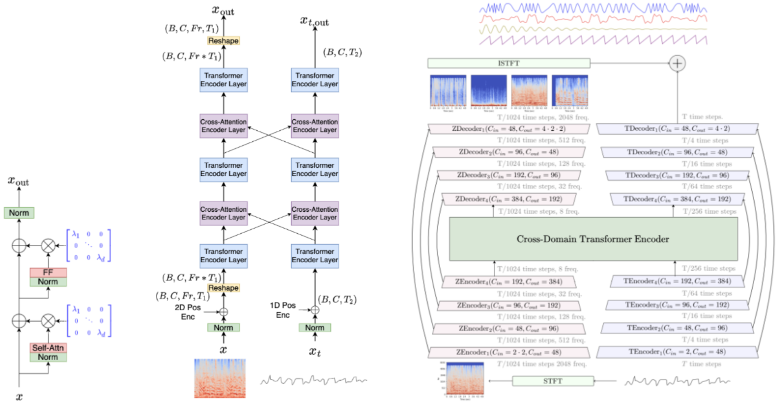
A few architectures





Source: Yang et al., "A Transformer-Based Approach to Music Separation", Tech report, 2023.

A few architectures



(a) Transformer Encoder Layer

(b) Cross-domain Transformer Encoder of depth 5

(c) Hybrid Transformer Demucs

Source: Rouard et al., "Hybrid transformers for music source separation", Proc. IEEE ICASSP, 2023.

Results

✓ Separation performance is impressive

	Vocals	Bass	Drums	Guitar
Open Unmix (2018)				
BSRNN (2023)				

Results

✓ Separation performance is impressive

	Vocals	Bass	Drums	Guitar
Open Unmix (2018)				
BSRNN (2023)				

✗ But. . .

- ▷ Tedious / endless experiments.
- ▷ Exacerbates the reproductibility crisis.
- ▷ Energy / environmental costs.
- ▷ Black boxes / lack of interpretability.
- ▷ Difficult to adapt to new / slightly different tasks.

lator [36], which approximates energy consumption base on hardware specifications (we consider a 3 W power pe 8 GB of memory).[3] This amounts to 19,030 kWh, whic is more than 44 times the energy consumption of trainin the best model, or 150 times that of the base model.

In all fairness, part of this cost is due to our own im plementation errors, which resulted in, e.g., interrupted o redundant training runs. However, we believe that mos

Source: Magron et al., "A case for reproductibility: replicating 'Band-split RNN for music separation'", 2025.

well as the mixed precision, where the STFT and iSTFT modules use FP32 and all the others use FP16.

We trained three separation models respectively for vocals, bass, and drums using In-House and the Musdb18HQ training set. For the "other" stem, we subtracted the vocals, bass, and drums signals from the input mixture in the time domain. For each model, the training process lasted for 4 weeks using 16 Nvidia A100-80GB GPUs with a total batch size of 128 (i.e., 8 for each GPU). The model checkpoint with the best validation result was selected.

Enframe & Deframe. We use a hop size of 4 seconds for

Source: Lu et al., "Music Source Separation with Band-Split RoPE Transformer", 2024.

What to do?

Clear pros and cons for both data-driven and expert knowledge-based approaches.

Deep Learning approach – Revolution I

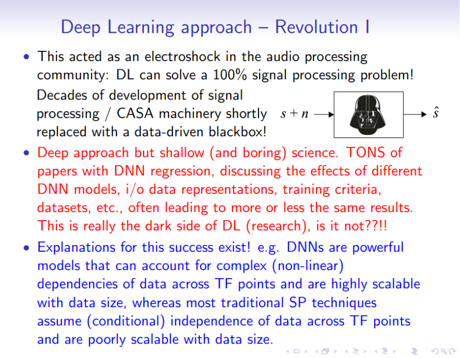
- This acted as an electroshock in the audio processing community: DL can solve a 100% signal processing problem!
  Decades of development of signal processing / CASA machinery shortly replaced with a data-driven blackbox!

- Deep approach but shallow (and boring) science. TONS of papers with DNN regression, discussing the effects of different DNN models, i/o data representations, training criteria, datasets, etc., often leading to more or less the same results. This is really the dark side of DL (research), is it not??!!
- Explanations for this success exist! e.g. DNNs are powerful models that can account for complex (non-linear) dependencies of data across TF points and are highly scalable with data size, whereas most traditional SP techniques assume (conditional) independence of data across TF points and are poorly scalable with data size.

Source: Girin, "Deep Learning for Speech Enhancement", 2018.

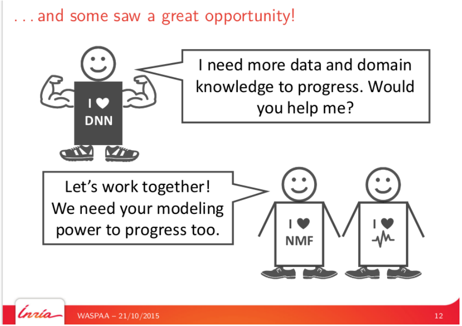
What to do?

Clear pros and cons for both data-driven and expert knowledge-based approaches.



Source: Girin, "Deep Learning for Speech Enhancement", 2018.

The obvious solution: combine them. (not exactly breaking news)



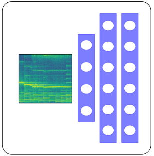
Source: Vincent, "Is audio signal processing still useful in the era of machine learning?", 2015.

Features / data representation

- ▷ Input features = the essence of data-driven models.
- ▷ Tradeoff between hand-crafted features (more robust) and raw data (more powerful).

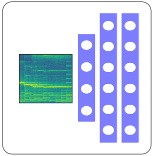
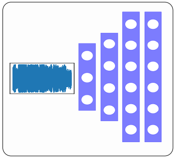
Features / data representation

- ▷ Input features = the essence of data-driven models.
- ▷ Tradeoff between hand-crafted features (more robust) and raw data (more powerful).

Feature domain	Magnitude
	
Performance	(✓)
Data need / model size	✓
Robustness / flexibility	✓

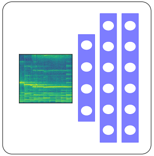
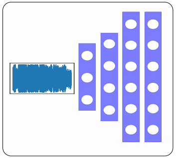
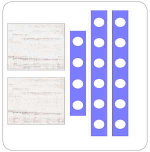
Features / data representation

- ▷ Input features = the essence of data-driven models.
- ▷ Tradeoff between hand-crafted features (more robust) and raw data (more powerful).

Feature domain	Magnitude	Waveform
		
Performance	(✓)	✓
Data need / model size	✓	✗
Robustness / flexibility	✓	✗

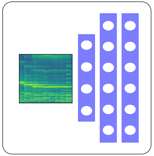
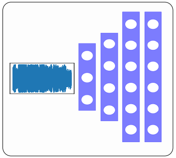
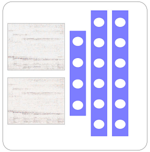
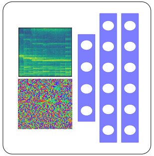
Features / data representation

- ▷ Input features = the essence of data-driven models.
- ▷ Tradeoff between hand-crafted features (more robust) and raw data (more powerful).

Feature domain	Magnitude	Waveform	STFT real / imaginary
			
Performance	(✓)	✓	✓
Data need / model size	✓	✗	(✓)
Robustness / flexibility	✓	✗	(✓)

Features / data representation

- ▷ Input features $=$ the essence of data-driven models.
- ▷ Tradeoff between hand-crafted features (more robust) and raw data (more powerful).

Feature domain	Magnitude	Waveform	STFT real / imaginary	STFT magnitude / phase
				
Performance	(✓)	✓	✓	✓
Data need / model size	✓	✗	(✓)	✓
Robustness / flexibility	✓	✗	(✓)	✓

Hybrid architectures

Traditional NMF: a statistical framework / generative model.

- ▷ Estimation by alternating spectral and spatial parameters.
- ▷ Sources are recovered via filtering.

Algorithm

Input:

$\mathbf{x}_{fn}$ ▷ STFT of mixture: $I \times 1$

$z_{xfn} \leftarrow$ preprocess $(\mathbf{x}_{fn})$

for $j \leftarrow 1, J$ **do**
    $\mathbf{R}_{jfn} \leftarrow I \times I$ identity matrix

for $l \leftarrow 1, L$ **do**
    for $k \leftarrow 1, K$ **do**
        for $j \leftarrow 1, J$ **do**
            $\widehat{\mathbf{c}}_{jfn} \leftarrow (3)$
            $\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}} \leftarrow (5)$
            $\mathbf{R}_{jf} \leftarrow (8)$
    for $j \leftarrow 1, J$ **do**
        $z_{jfn} \leftarrow (7)$
    $[v_{1fn}, \ldots, v_{Jfn}] \leftarrow \left[\text{NMF}\left(\sqrt{z_{1fn}}, \ldots, \sqrt{z_{jfn}}\right)\right]^2$

for $j \leftarrow 1, J$ **do**
    $\widehat{\mathbf{c}}_{jfn} \leftarrow (3)$

Output:

$\widehat{\mathbf{c}}_{jfn}$ ▷ STFT of sources images

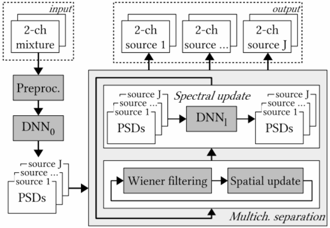
Hybrid architectures

Traditional NMF: a statistical framework / generative model.

- ▷ Estimation by alternating spectral and spatial parameters.
- ▷ Sources are recovered via filtering.

Hybrid approach

- ▷ Filtering in cunjunction with DNN-based spectral modeling.
- ▷ Lighter and more robust than end-to-end approaches.



Algorithm

Input:

$\mathbf{x}_{fn}$ ▷ STFT of mixture: $I \times 1$

$z_{xfn} \leftarrow \text{preprocess}(\mathbf{x}_{fn})$

for $j \leftarrow 1, J$ **do**
  $\mathbf{R}_{jfn} \leftarrow I \times I$ identity matrix
for $l \leftarrow 1, L$ **do**
  for $k \leftarrow 1, K$ **do**
    for $j \leftarrow 1, J$ **do**
      $\widehat{\mathbf{c}}_{jfn} \leftarrow (3)$
      $\widehat{\mathbf{R}}_{\mathbf{c}_{jfn}} \leftarrow (5)$
      $\mathbf{R}_{jf} \leftarrow (8)$
  for $j \leftarrow 1, J$ **do**
    $z_{jfn} \leftarrow (7)$
  $[v_{1fn}, \ldots, v_{Jfn}] \leftarrow \left[\text{DNN}\left(\sqrt{z_{1fn}}, \ldots, \sqrt{z_{Jfn}}\right)\right]^2$
for $j \leftarrow 1, J$ **do**
  $\widehat{\mathbf{c}}_{jfn} \leftarrow (3)$

Output:

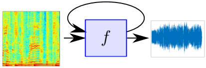
$\widehat{\mathbf{c}}_{jfn}$ ▷ STFT of sources images

Source: Nugraha et al., "Multichannel Music Separation with Deep Neural Networks", Proc. EUSIPCO, 2016.

Unfolding algorithms

Problem

- Estimated spectrograms have to be reverted back to waveforms.
- This is done via *spectrogram inversion* iterative algorithms.



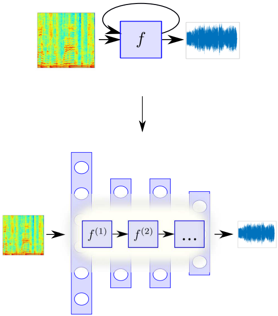
Unfolding algorithms

Problem

- ▷ Estimated spectrograms have to be reverted back to waveforms.
- ▷ This is done via *spectrogram inversion* iterative algorithms.

Deep unfolding

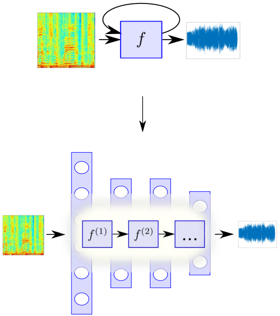
- ▷ Each algorithm's iteration = one layer of a neural network.
- ▷ Train via backpropagation through the unfolded algorithm.



Unfolding algorithms

Problem

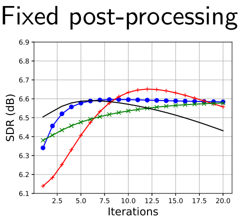
- ▷ Estimated spectrograms have to be reverted back to waveforms.
- ▷ This is done via *spectrogram inversion* iterative algorithms.

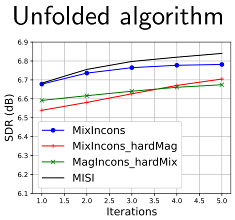


Deep unfolding

- ▷ Each algorithm's iteration = one layer of a neural network.
- ▷ Train via backpropagation through the unfolded algorithm.

Results





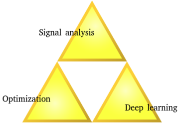
✓ Improved performance over a fixed post-processing, with (almost) no additional parameter.

Conclusion

Key message

Combining expert knowledge and data-driven models:
a promising approach for machine learning-based music separation.

- ▷ Enable networks to exploit prior information.
- ▷ Improve their robustness and reduce their size.
- ▷ More interpretable and principled networks.

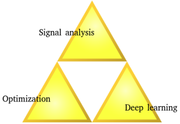


Conclusion

Key message

Combining expert knowledge and data-driven models:
a promising approach for machine learning-based music separation.

- ▷ Enable networks to exploit prior information.
- ▷ Improve their robustness and reduce their size.
- ▷ More interpretable and principled networks.



Perspectives

- ▷ A more systematic use of this approach.
- ▷ Adaptation to specific sources / instruments / setups.
- ▷ Extension to other tasks, e.g., musical motif discovery.