Demixing Sounds with AI: Towards Deep Phase Recovery

Paul Magron, Researcher - INRIA Centre at Université de Lorraine

Seminar at the Observatoire Astronomique de Strasbourg - November 4th, 2024

MULTISPEECH

The audio realm

The audio realm

The audio realm

Speech

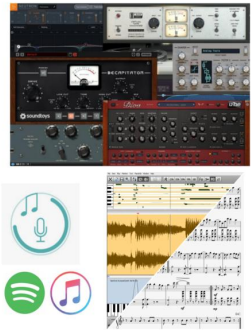
Ambient sounds

The audio realm

Speech

Ambient sounds

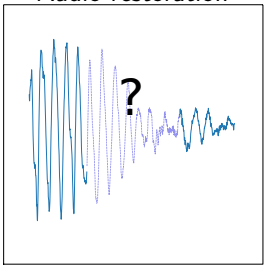
Music signals



My research themes

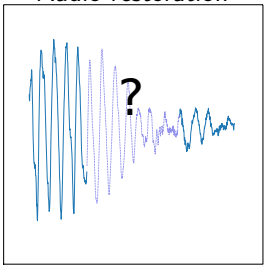
My research themes

Audio restoration

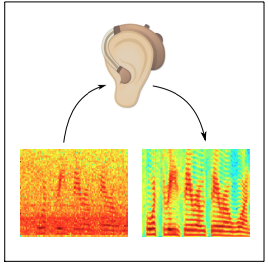


My research themes

Audio restoration

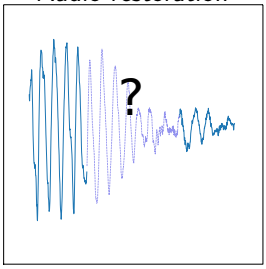


Speech enhancement

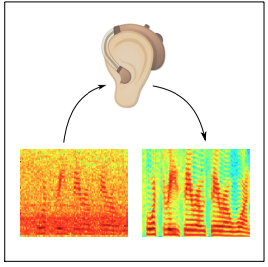


My research themes

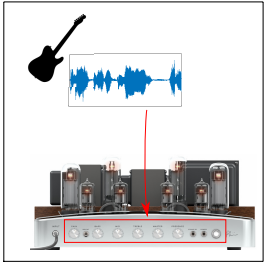
Audio restoration



Speech enhancement

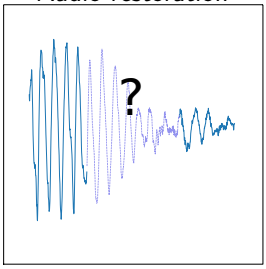


Tone matching

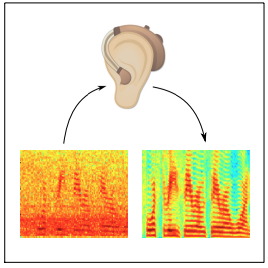


My research themes

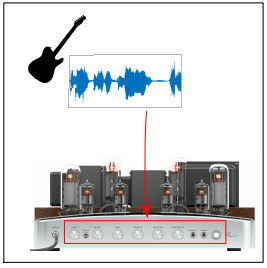
Audio restoration



Speech enhancement



Tone matching



- ▷ + some past applications: recommender systems, acoustic scene analysis...
- ▷ Main task: audio demixing.

Audio demixing

Audio demixing

Audio signals are composed of several constitutive sounds:

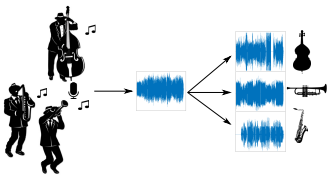
- ▷ Multiple speakers, background noise, domestic sounds, music instruments...

Audio demixing

Audio signals are composed of several constitutive sounds:

- ▷ Multiple speakers, background noise, domestic sounds, music instruments...

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



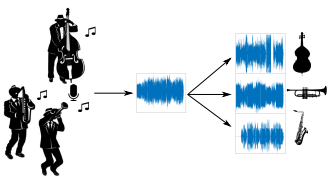
Audio demixing

Audio signals are composed of several constitutive sounds:

- Multiple speakers, background noise, domestic sounds, music instruments...

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

- An important preprocessing for many downstream tasks.
  - Automatic speech recognition / music transcription.
  - Music information retrieval.
  - Environmental scene analysis.
- A goal in itself for synthesis purposes.
  - Augmented mixing e.g., from mono to stereo.
  - Voice removal (karaoke).



Demixing for music backtrack generation

Situation: A guitarist wants to play a nice solo.

- ✗ Doesn't know how to.
- ✗ No bandmate around.

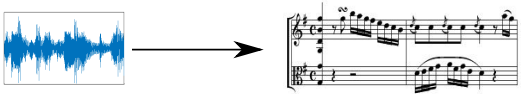
Demixing for music backtrack generation

Situation: A guitarist wants to play a nice solo.

- ✗ Doesn't know how to.
- ✗ No bandmate around.

Music separation can help you ☺

- ▷ Isolate the guitar track to learn your part.
  - ▷ Either by applying time-stretching to a slower tempo.
  - ▷ Or by generating a music score via automatic transcription.



Demixing for music backtrack generation

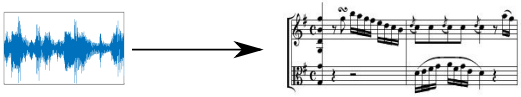
Situation: A guitarist wants to play a nice solo.

✗ Doesn't know how to.

✗ No bandmate around.

Music separation can help you ☺

- ▷ Isolate the guitar track to learn your part.
  - ▷ Either by applying time-stretching to a slower tempo.
  - ▷ Or by generating a music score via automatic transcription.



- ▷ Combine the other instruments and play over the backing track!

Mix 🔊 Backing track 🔊

Demixing for hearing aids

The cocktail party problem: many interfering sounds.

- ▷ Humans are very good as focusing on one target source.
- ▷ Much harder for people with hearing deficiencies.

Demixing for hearing aids

The cocktail party problem: many interfering sounds.

- ▷ Humans are very good as focusing on one target source.
- ▷ Much harder for people with hearing deficiencies.

Enhancement:

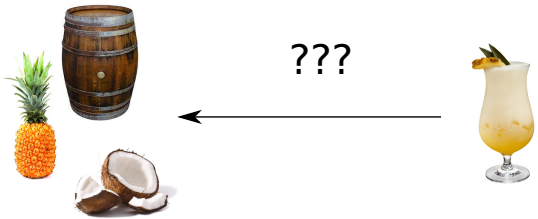
- ▷ With a brute-force gain:
- ▷ With a prior demixing:

A difficult task?

Mixing is easy . . .

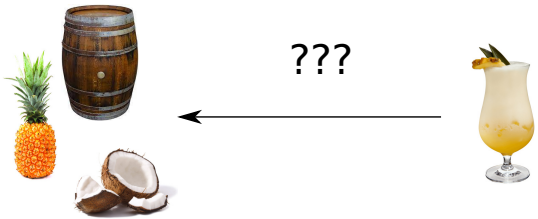
A difficult task?

Mixing is easy . . . but demixing is not.



A difficult task?

Mixing is easy . . . but demixing is not.

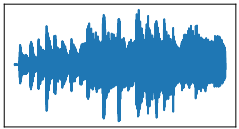


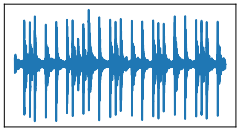
An under-determined problem.

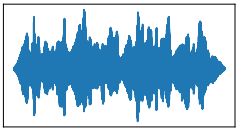
- ▷ Need to incorporate some knowledge / priors / additional information.

Setting the stage

- The raw material: audio signals.

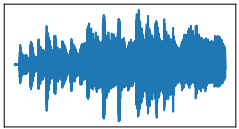


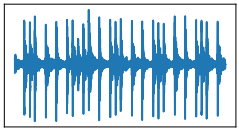


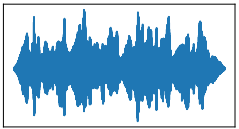


Setting the stage

- The raw material: audio signals.



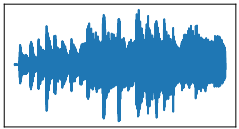


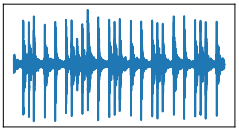


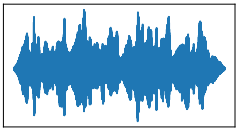
- It's hard to see structure there...

Setting the stage

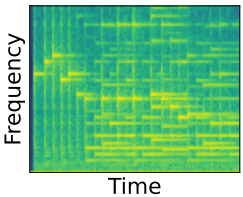
- The raw material: audio signals.

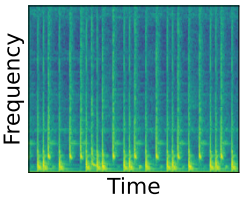


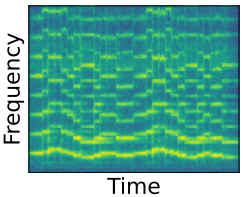




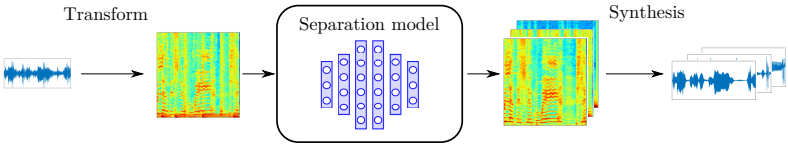
- It's hard to see structure there...
- We rather transform them into a time-frequency representation, e.g., a spectrogram.



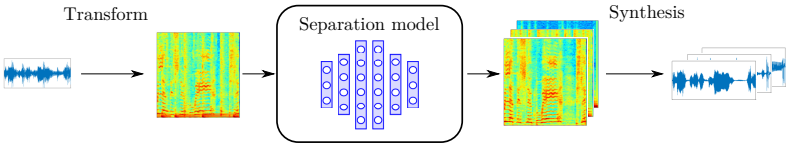




The demixing pipeline

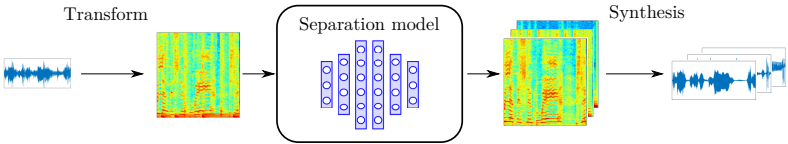


The demixing pipeline



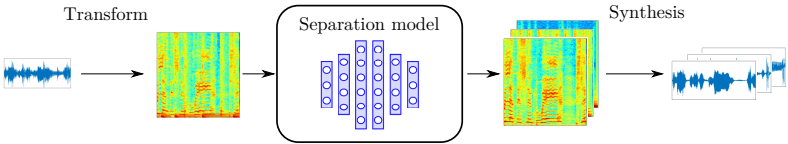
▷ The transform is usually the short-time Fourier transform (STFT).

The demixing pipeline



- The transform is usually the short-time Fourier transform (STFT).
- The separator is based on:
  - Nonnegative matrix factorization (NMF).
  - Deep neural networks (DNNs).

The demixing pipeline



- The transform is usually the short-time Fourier transform (STFT).
- The separator is based on:
  - 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.

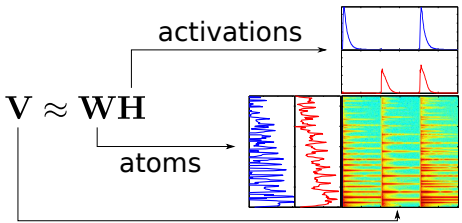
Nonnegative matrix factorization (NMF)

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.



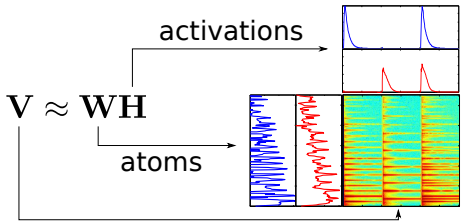
Nonnegative matrix factorization (NMF)

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:

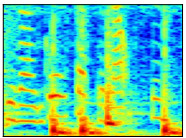
$$\min_{\mathbf{W},\mathbf{H}} D(\mathbf{V}, \mathbf{WH}) \quad + \text{regularizations}$$

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

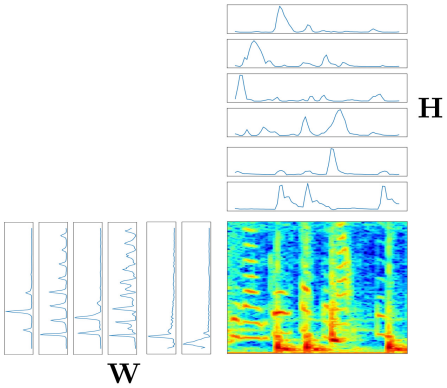
NMF for audio demixing

Exploit additivity for getting each source spectrogram.

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



NMF for audio demixing



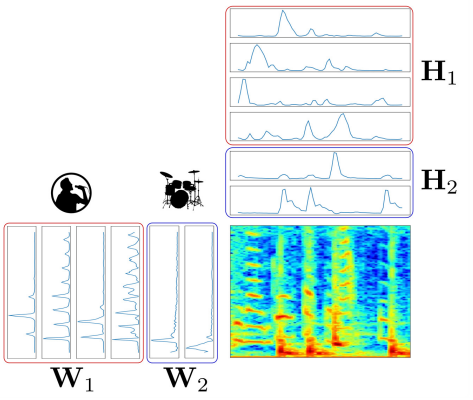
Exploit additivity for getting each source spectrogram.

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

Procedure

1. Factorize the mixture's spectrogram.

NMF for audio demixing



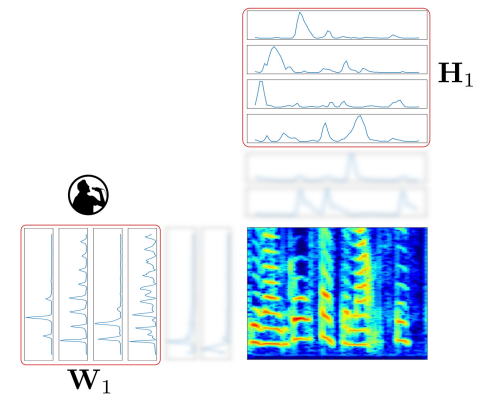
Exploit additivity for getting each source spectrogram.

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

Procedure

1. Factorize the mixture's spectrogram.
2. Cluster atoms that belong to the same source.

NMF for audio demixing



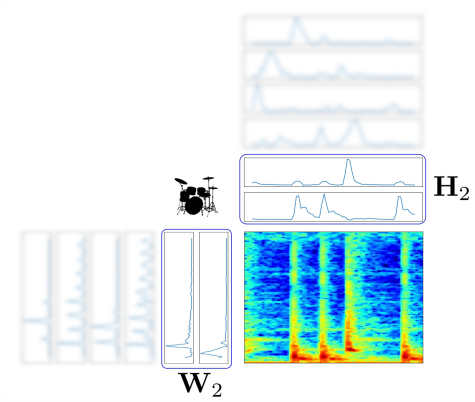
Exploit additivity for getting each source spectrogram.

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

Procedure

1. Factorize the mixture's spectrogram.
2. Cluster atoms that belong to the same source.
3. Multiply each dictionary with the corresponding activations.

NMF for audio demixing



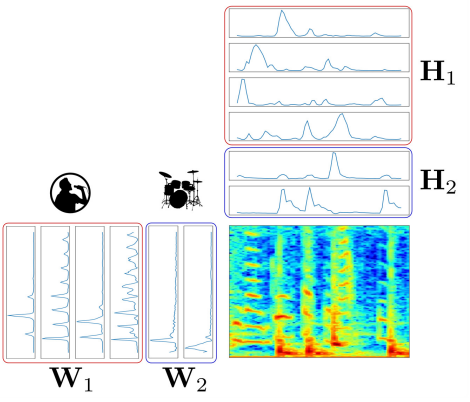
Exploit additivity for getting each source spectrogram.

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

Procedure

1. Factorize the mixture's spectrogram.
2. Cluster atoms that belong to the same source.
3. Multiply each dictionary with the corresponding activations.

NMF for audio demixing



Exploit additivity for getting each source spectrogram.

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

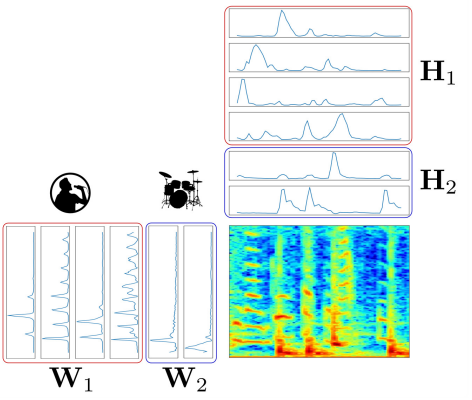
Procedure

1. Factorize the mixture's spectrogram.
2. Cluster atoms that belong to the same source.
3. Multiply each dictionary with the corresponding activations.

Supervised demixing

Pretrain $\mathbf{W}_1$ and $\mathbf{W}_2$ on subsets of isolated tracks.

NMF for audio demixing



Exploit additivity for getting each source spectrogram.

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

Procedure

1. Factorize the mixture's spectrogram.
2. Cluster atoms that belong to the same source.
3. Multiply each dictionary with the corresponding activations.

Supervised demixing

Pretrain $\mathbf{W}_1$ and $\mathbf{W}_2$ on subsets of isolated tracks.

✓ Light and interpretable model.

✗ Performance is 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 potentially high-dimensional (in audio: spectrograms).
- f_θ is built by assembling (many) neurons and activation functions (for demixing, $|\theta| \sim 10^7$).

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 potentially high-dimensional (in audio: spectrograms).
- f_θ is built by assembling (many) neurons and activation functions (for demixing, $|\theta| \sim 10^7$).

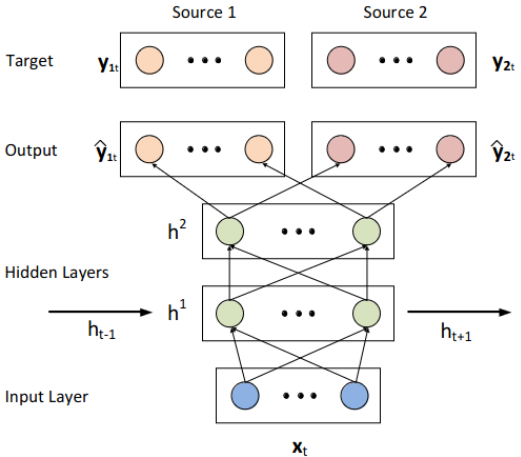
Supervised learning

- 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:

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

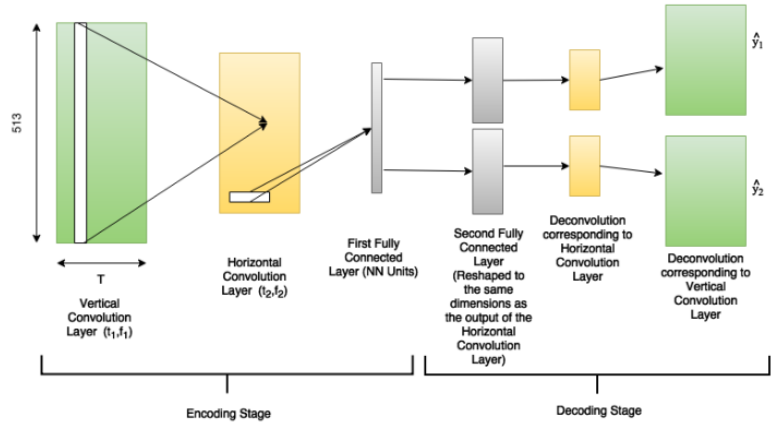
- Solved with a stochastic gradient descent algorithm (e.g., ADAM).

DNNs for audio demixing



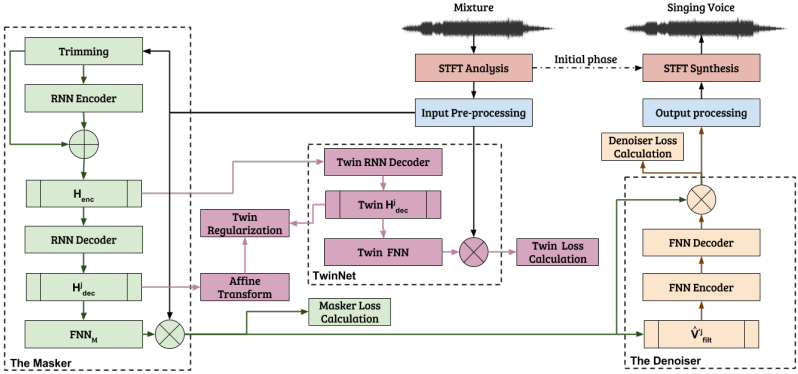
Source: Huang et al., "Deep learning for monaural speech separation", Proc. IEEE ICASSP, 2014.

DNNs for audio demixing



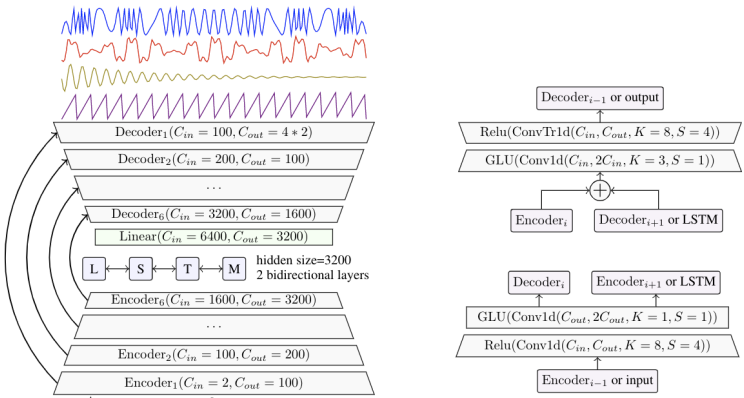
Source: Chandna et al., "Monoaural Audio Source Separation Using Deep Convolutional Neural Networks", Lecture Notes in Computer Science, 2017.

DNNs for audio demixing



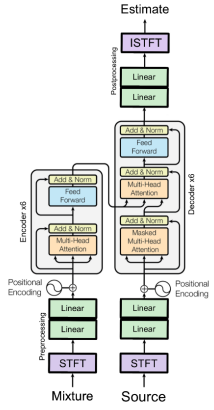
Source: Drossos et al., "MaD TwinNet: Masker-Denoiser Architecture with Twin Networks for Monaural Sound Source Separation", Proc. IEEE IJCNN, 2018.

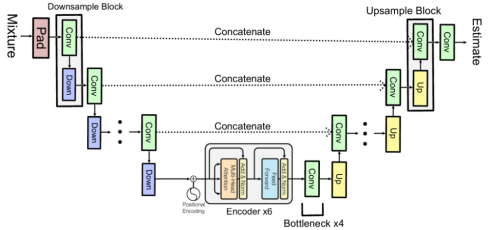
DNNs for audio demixing



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

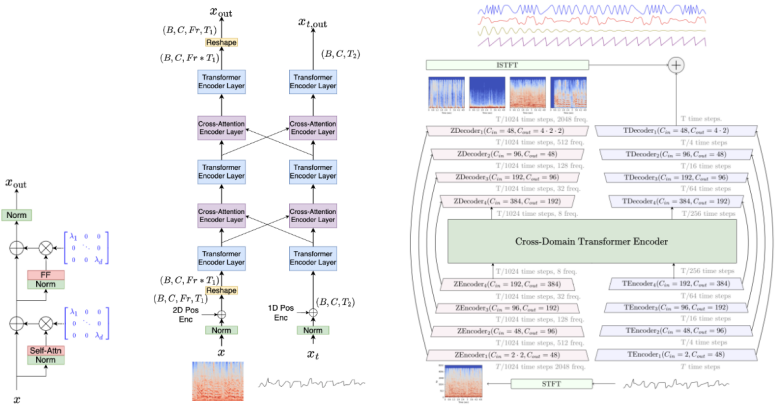
DNNs for audio demixing





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

DNNs for audio demixing



(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.

What is the research like?

Mostly “machine learning” themes:

- ▷ Architectures.
- ▷ Optimization strategies.
- ▷ Datasets / data augmentation.

What is the research like?

Mostly “machine learning” themes:

- ▷ Architectures.
- ▷ Optimization strategies.
- ▷ Datasets / data augmentation.

Is it good?

- ✓ Impressive performance.
- ☹ But a bit boring.

What is the research like?

Mostly "machine learning" themes:

- ▷ Architectures.
- ▷ Optimization strategies.
- ▷ Datasets / data augmentation.

Is it good?

- ✓ Impressive performance.
- ☹ But a bit boring.

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 is the research like?

Mostly "machine learning" themes:

- ▷ Architectures.
- ▷ Optimization strategies.
- ▷ Datasets / data augmentation.

Is it good?

- ✓ Impressive performance.
- ☹ But a bit boring.

4.6. Ablation Study

We investigate the effects of three aspects: 1) the number of Transformer blocks; 2) RoPE or absolute positional encoding; 3) TC or OA deframing methods. To this end, we implement three smaller variants of the proposed model with L=6. In one variant, we remove the RoPE and add learnable absolute positional embeddings in the attention module, and call it "BS-Transformer," since it uses a standard Transformer. We train a dedicated separation model for the "other" stem, except "BS-RoFormer (L=12, OA)," which is our MDX'23 submission. The numbers of parameters for BS-RoFormer and BS-Transformer with L=6 are 72.2M and 72.5M, respectively. Models with L=6 are trained solely on the Musdb18HQ training set using 16 Nvidia V100-32GB GPUs. We do not use the In-House dataset for ablation study. The effective batch size is 64 (i.e., 4 for each GPU) using accumulate_grad_batches=2.

Table 2 presents the comparison between our proposed models and existing models. We report the median SDR across the median SDRs over all 1 second chunks of each test song in Musdb18HQ following prior works. First, BS-RoFormer with L=6 is still very competitive and can achieve state-of-the-art performance compared to models trained without extra training data. "BS-RoFormer† (L=12, OA)" outperforms all existing models by a large margin (over 2 dB on average). Second, BS-Transformer without RoPE does not seem to work given the low SDRs, demonstrating that RoPE is crucial in our proposed architecture as discussed in Section 3.2.1. According to our observations, the training progress of BS-Transformer is very slow, and it still remains low SDRs after two weeks of training on Musdb18HQ. Instead, BS-RoFormer models with L=6 get converged within a week. Lastly, the OA deframing shows better performance than TC except vocals. Qualitatively, OA offers smoother song-level quality that can improve the overall listening experience.

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

What is the research like?

Mostly "machine learning" themes:

- ▷ Architectures.
- ▷ Optimization strategies.
- ▷ Datasets / data augmentation.

Is it good?

- ✓ Impressive performance.
- ☹ But a bit boring.

Let us investigate something different ☺

4.6. Ablation Study

We investigate the effects of three aspects: 1) the number of Transformer blocks; 2) RoPE or absolute positional encoding; 3) TC or OA deframing methods. To this end, we implement three smaller variants of the proposed model with L=6. In one variant, we remove the RoPE and add learnable absolute positional embeddings in the attention module, and call it "BS-Transformer," since it uses a standard Transformer. We train a dedicated separation model for the "other" stem, except "BS-RoFormer (L=12, OA)," which is our MDX'23 submission. The numbers of parameters for BS-RoFormer and BS-Transformer with L=6 are 72.2M and 72.5M, respectively. Models with L=6 are trained solely on the Musdb18HQ training set using 16 Nvidia V100-32GB GPUs. We do not use the In-House dataset for ablation study. The effective batch size is 64 (i.e., 4 for each GPU) using accumulate_grad_batches=2.

Table 2 presents the comparison between our proposed models and existing models. We report the median SDR across the median SDRs over all 1 second chunks of each test song in Musdb18HQ following prior works. First, BS-RoFormer with L=6 is still very competitive and can achieve state-of-the-art performance compared to models trained without extra training data. "BS-RoFormer† (L=12, OA)" outperforms all existing models by a large margin (over 2 dB on average). Second, BS-Transformer without RoPE does not seem to work given the low SDRs, demonstrating that RoPE is crucial in our proposed architecture as discussed in Section 3.2.1. According to our observations, the training progress of BS-Transformer is very slow, and it still remains low SDRs after two weeks of training on Musdb18HQ. Instead, BS-RoFormer models with L=6 get converged within a week. Lastly, the OA deframing shows better performance than TC except vocals. Qualitatively, OA offers smoother song-level quality that can improve the overall listening experience.

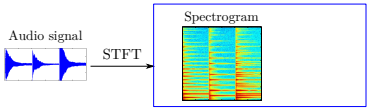
Source: Lu et al., "Music Source Separation with Band-Split RoPE Transformer", 2024.

Phase recovery

The phase catch

$$\mathbf{x} \in \mathbb{R}^N \xrightarrow{\text{STFT}} \mathbf{X} \in \mathbb{C}^{F \times T}$$

The STFT produces a spectrogram $|\mathbf{X}|$

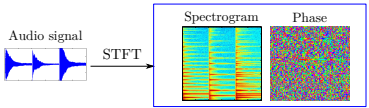


The phase catch

$$\mathbf{x} \in \mathbb{R}^N \xrightarrow{\text{STFT}} \mathbf{X} \in \mathbb{C}^{F \times T}$$

The STFT produces a spectrogram $|\mathbf{X}|$

. . . but also a phase $\angle \mathbf{X}$.

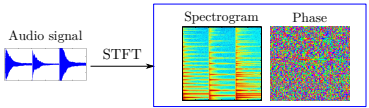


The phase catch

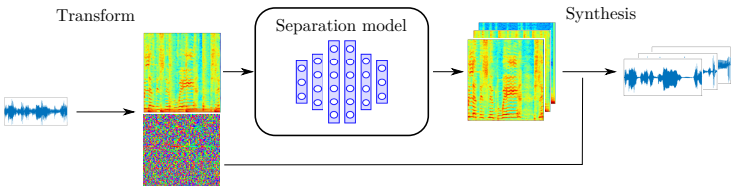
$$\mathbf{x} \in \mathbb{R}^N \xrightarrow{\text{STFT}} \mathbf{X} \in \mathbb{C}^{F \times T}$$

The STFT produces a spectrogram $|\mathbf{X}|$

. . . but also a phase $\angle \mathbf{X}$.

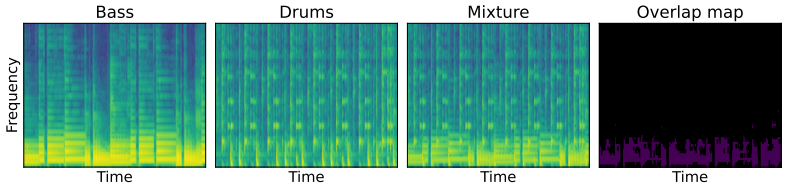


- ▷ For sound demixing, the phase of each source must be *recovered*.
- ▷ Nonnegative masking (e.g., Wiener filter) assigns the mixture's phase to each source:
  $\forall j, \quad \angle \mathbf{S}_j = \angle \mathbf{X}$



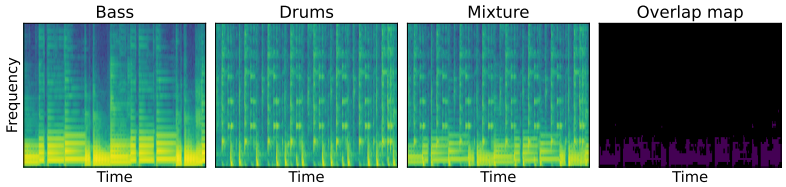
The potential of phase recovery

Audio signals often *overlap* in the TF domain: more than 1 active source in a given TF point.



The potential of phase recovery

Audio signals often *overlap* in the TF domain: more than 1 active source in a given TF point.



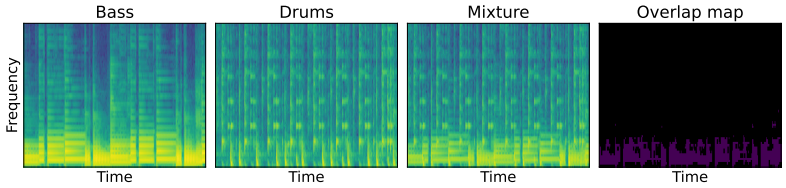
When sources *overlap*: $\angle \mathbf{X} \neq \angle \mathbf{S}_j$.

✗ Issues in sound quality with nonnegative masking.

✓ More potential gain in phase recovery than in magnitude estimation.

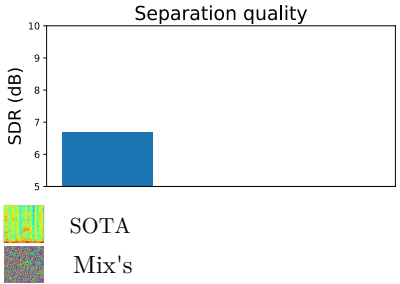
The potential of phase recovery

Audio signals often *overlap* in the TF domain: more than 1 active source in a given TF point.



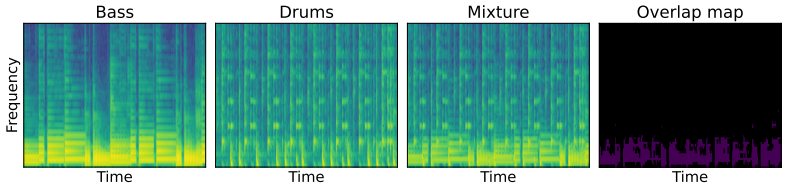
When sources *overlap*: $\angle \mathbf{X} \neq \angle \mathbf{S}_j$.

- ✗ Issues in sound quality with nonnegative masking.
- ✓ More potential gain in phase recovery than in magnitude estimation.



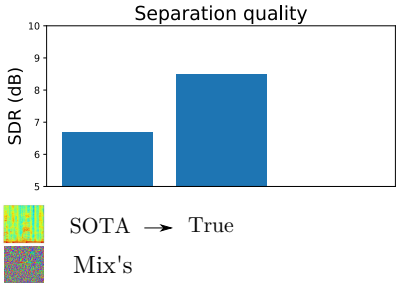
The potential of phase recovery

Audio signals often *overlap* in the TF domain: more than 1 active source in a given TF point.



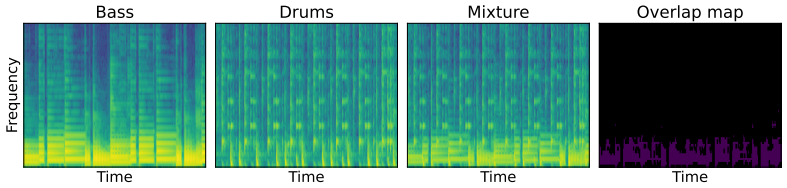
When sources *overlap*: $\angle \mathbf{X} \neq \angle \mathbf{S}_j$.

- ✗ Issues in sound quality with nonnegative masking.
- ✓ More potential gain in phase recovery than in magnitude estimation.



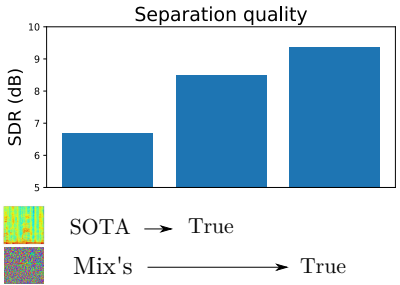
The potential of phase recovery

Audio signals often *overlap* in the TF domain: more than 1 active source in a given TF point.

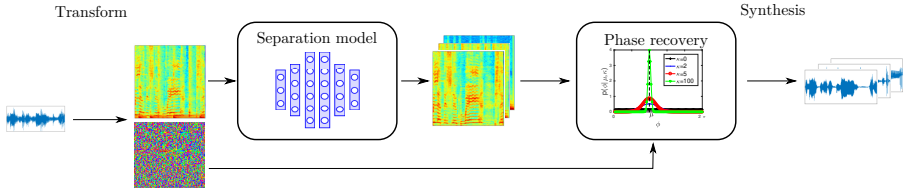


When sources *overlap*: $\angle \mathbf{X} \neq \angle \mathbf{S}_j$.

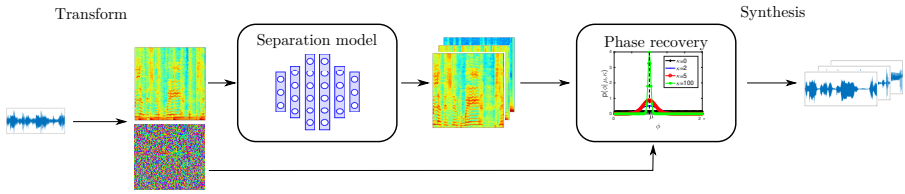
- ✗ Issues in sound quality with nonnegative masking.
- ✓ More potential gain in phase recovery than in magnitude estimation.



Phase recovery for audio demixing

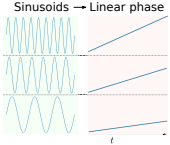


Phase recovery for audio demixing

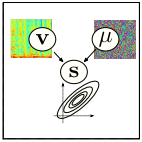


Main contributions

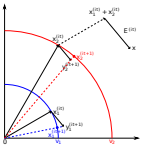
Phase models



Probabilistic framework



Inversion algorithms



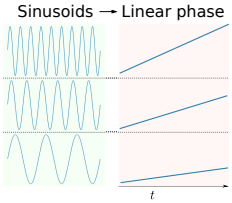
Sinusoidal phase model

Consider a mixture of sinusoids: $x(n) = \sum_{p=1}^{P} \underbrace{A_p}_{\text{amplitude}} \sin(2\pi \underbrace{\nu_p}_{\text{frequency}} n + \phi_{0,p})$.

Sinusoidal phase model

Consider a mixture of sinusoids: $x(n) = \sum_{p=1}^{P} \underbrace{A_p}_{\text{amplitude}} \sin(2\pi \underbrace{\nu_p}_{\text{frequency}} n + \phi_{0,p})$.

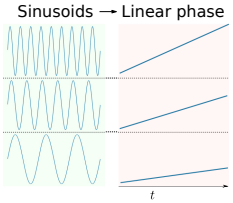
The STFT phase follows: $\mu_{f,t} = \mu_{f,t-1} + l\nu_{f,t}$

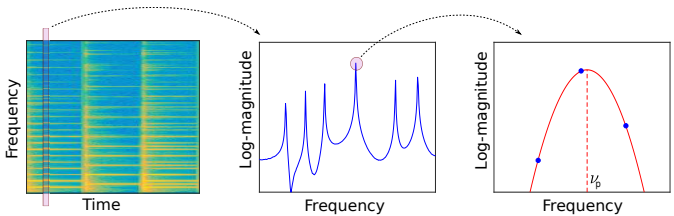


Sinusoidal phase model

Consider a mixture of sinusoids: $x(n) = \sum_{p=1}^{P} \underbrace{A_p}_{\text{amplitude}} \sin(2\pi \underbrace{\nu_p}_{\text{frequency}} n + \phi_{0,p})$.

The STFT phase follows: $\mu_{f,t} = \mu_{f,t-1} + l\nu_{f,t}$

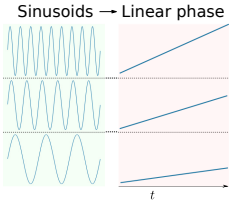


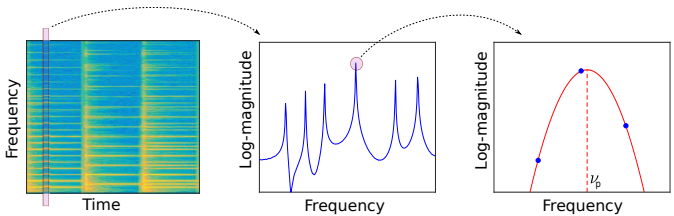


Sinusoidal phase model

Consider a mixture of sinusoids: $x(n) = \sum_{p=1}^{P} \underbrace{A_p}_{\text{amplitude}} \sin(2\pi \underbrace{\nu_p}_{\text{frequency}} n + \phi_{0,p})$.

The STFT phase follows: $\mu_{f,t} = \mu_{f,t-1} + l\nu_{f,t}$





✓ Accounts for non-stationary signals, suitable for real-time processing.

Original Random phase Recovered

Spectrogram inversion algorithms

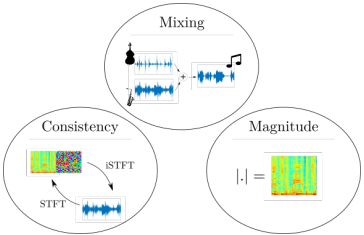
Problem setting: Given the mixture $\mathbf{X}$ and estimated sources' magnitudes $\mathbf{V}_j$, recover time-domain signals $\mathbf{s}_j$ (or equivalently their STFTs $\mathbf{S}_j$).

Spectrogram inversion algorithms

Problem setting: Given the mixture $\mathbf{X}$ and estimated sources' magnitudes $\mathbf{V}_j$, recover time-domain signals $\mathbf{s}_j$ (or equivalently their STFTs $\mathbf{S}_j$).

General framework

- ▷ Identify important properties in the STFT domain.
- ▷ Combine them to formulate optimization problems.

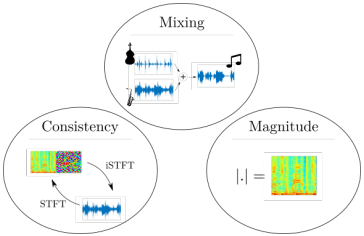


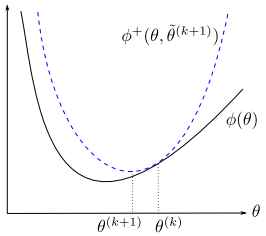
Spectrogram inversion algorithms

Problem setting: Given the mixture $\mathbf{X}$ and estimated sources' magnitudes $\mathbf{V}_j$, recover time-domain signals $\mathbf{s}_j$ (or equivalently their STFTs $\mathbf{S}_j$).

General framework

- ▷ Identify important properties in the STFT domain.
- ▷ Combine them to formulate optimization problems.
- ▷ Solve these optimization problems with the auxiliary function method.





An example of spectrogram inversion algorithm

$$\min_{\{\mathbf{S}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \mathbf{S}_j||^2 \quad \text{subject to} \quad |\mathbf{S}_j| = \mathbf{V}_j$$

An example of spectrogram inversion algorithm

$$\min_{\{\mathbf{S}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \mathbf{S}_j||^2 \quad \text{subject to} \quad |\mathbf{S}_j| = \mathbf{V}_j$$

Optimization procedure

- ▷ Incorporate the constraint (method of Lagrange multipliers).
- ▷ Build an auxiliary function (introduce auxiliary parameters).
- ▷ Set the partial derivatives at 0 and solve.
- ✓ Convergence-guaranteed, no hyperparameter to tune.

An example of spectrogram inversion algorithm

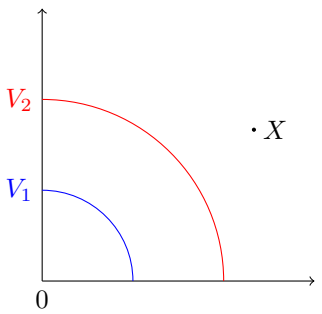
$$\min_{\{\mathbf{S}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \mathbf{S}_j||^2 \quad \text{subject to} \quad |\mathbf{S}_j| = \mathbf{V}_j$$

Optimization procedure

- ▷ Incorporate the constraint (method of Lagrange multipliers).
- ▷ Build an auxiliary function (introduce auxiliary parameters).
- ▷ Set the partial derivatives at 0 and solve.
- ✓ Convergence-guaranteed, no hyperparameter to tune.

$$\mathbf{Y}_j \leftarrow \mathbf{S}_j + \frac{1}{J}(\mathbf{X} - \sum_k \mathbf{S}_k)$$

$$\mathbf{S}_j \leftarrow \frac{\mathbf{Y}_j}{|\mathbf{Y}_j|} \odot \mathbf{V}_j$$



An example of spectrogram inversion algorithm

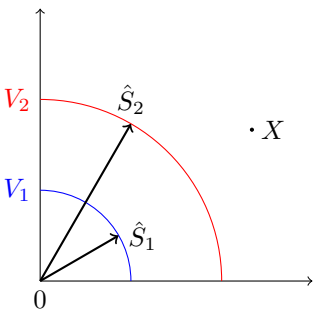
$$\min_{\{\mathbf{S}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \mathbf{S}_j||^2 \quad \text{subject to} \quad |\mathbf{S}_j| = \mathbf{V}_j$$

Optimization procedure

- ▷ Incorporate the constraint (method of Lagrange multipliers).
- ▷ Build an auxiliary function (introduce auxiliary parameters).
- ▷ Set the partial derivatives at 0 and solve.
- ✓ Convergence-guaranteed, no hyperparameter to tune.

$$\mathbf{Y}_j \leftarrow \mathbf{S}_j + \frac{1}{J}(\mathbf{X} - \sum_k \mathbf{S}_k)$$

$$\mathbf{S}_j \leftarrow \frac{\mathbf{Y}_j}{|\mathbf{Y}_j|} \odot \mathbf{V}_j$$



An example of spectrogram inversion algorithm

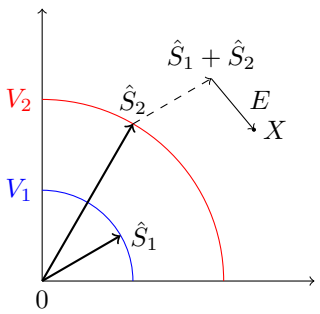
$$\min_{\{\mathbf{S}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \mathbf{S}_j||^2 \quad \text{subject to} \quad |\mathbf{S}_j| = \mathbf{V}_j$$

Optimization procedure

- ▷ Incorporate the constraint (method of Lagrange multipliers).
- ▷ Build an auxiliary function (introduce auxiliary parameters).
- ▷ Set the partial derivatives at 0 and solve.
- ✓ Convergence-guaranteed, no hyperparameter to tune.

$$\mathbf{Y}_j \leftarrow \mathbf{S}_j + \frac{1}{J}(\mathbf{X} - \sum_k \mathbf{S}_k)$$

$$\mathbf{S}_j \leftarrow \frac{\mathbf{Y}_j}{|\mathbf{Y}_j|} \odot \mathbf{V}_j$$



An example of spectrogram inversion algorithm

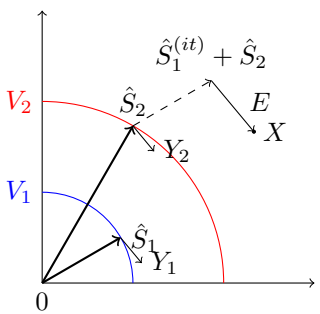
$$\min_{\{\mathbf{S}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \mathbf{S}_j||^2 \quad \text{subject to} \quad |\mathbf{S}_j| = \mathbf{V}_j$$

Optimization procedure

- ▷ Incorporate the constraint (method of Lagrange multipliers).
- ▷ Build an auxiliary function (introduce auxiliary parameters).
- ▷ Set the partial derivatives at 0 and solve.
- ✓ Convergence-guaranteed, no hyperparameter to tune.

$$\mathbf{Y}_j \leftarrow \mathbf{S}_j + \frac{1}{J}(\mathbf{X} - \sum_k \mathbf{S}_k)$$

$$\mathbf{S}_j \leftarrow \frac{\mathbf{Y}_j}{|\mathbf{Y}_j|} \odot \mathbf{V}_j$$



An example of spectrogram inversion algorithm

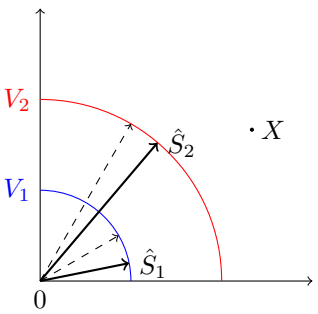
$$\min_{\{\mathbf{S}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \mathbf{S}_j||^2 \quad \text{subject to} \quad |\mathbf{S}_j| = \mathbf{V}_j$$

Optimization procedure

- ▷ Incorporate the constraint (method of Lagrange multipliers).
- ▷ Build an auxiliary function (introduce auxiliary parameters).
- ▷ Set the partial derivatives at 0 and solve.
- ✓ Convergence-guaranteed, no hyperparameter to tune.

$$\mathbf{Y}_j \leftarrow \mathbf{S}_j + \frac{1}{J}(\mathbf{X} - \sum_k \mathbf{S}_k)$$

$$\mathbf{S}_j \leftarrow \frac{\mathbf{Y}_j}{|\mathbf{Y}_j|} \odot \mathbf{V}_j$$



An example of spectrogram inversion algorithm

$$\min_{\{\mathbf{S}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \mathbf{S}_j||^2 \quad \text{subject to} \quad |\mathbf{S}_j| = \mathbf{V}_j$$

Optimization procedure

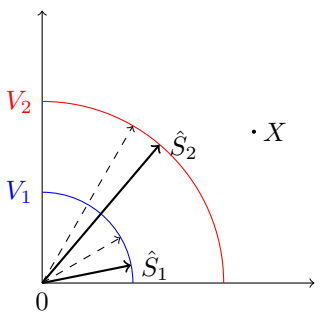
- ▷ Incorporate the constraint (method of Lagrange multipliers).
- ▷ Build an auxiliary function (introduce auxiliary parameters).
- ▷ Set the partial derivatives at 0 and solve.
- ✓ Convergence-guaranteed, no hyperparameter to tune.

Performance (SDR in dB)

Mixture phase	7.5
Random initialization	9.5
Sinusoidal initialization	**13.6**

$$\mathbf{Y}_j \leftarrow \mathbf{S}_j + \frac{1}{J}(\mathbf{X} - \sum_k \mathbf{S}_k)$$

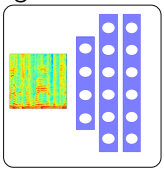
$$\mathbf{S}_j \leftarrow \frac{\mathbf{Y}_j}{|\mathbf{Y}_j|} \odot \mathbf{V}_j$$



Towards deep phase recovery

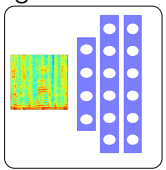
Current trends

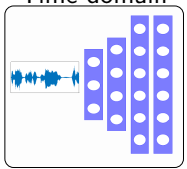
Magnitude domain



Performance	(✓)
Data efficiency / model size	✓
Robustness / flexibility	✓

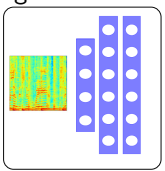
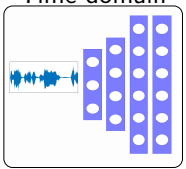
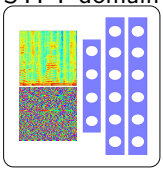
Current trends



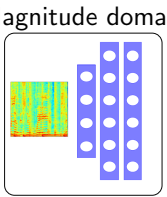


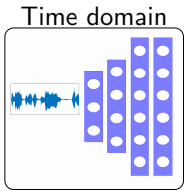
Performance	(✓)	✓
Data efficiency / model size	✓	✗
Robustness / flexibility	✓	✗

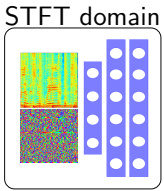
Current trends

	Magnitude domain	Time domain	STFT domain
			
Performance	(✓)	✓	✓
Data efficiency / model size	✓	✗	(✓)
Robustness / flexibility	✓	✗	(✓)

Current trends







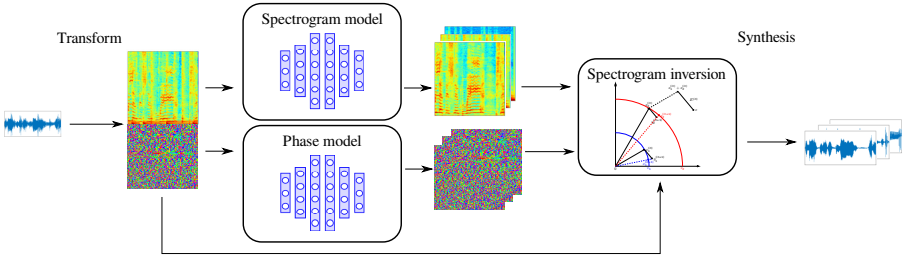
	Magnitude domain	Time domain	STFT domain
Performance	(✓)	✓	✓
Data efficiency / model size	✓	✗	(✓)
Robustness / flexibility	✓	✗	(✓)

STFT-domain approaches process a real/imaginary part decomposition of the (complex-valued) data.

- ✗ Sub-optimal: redundancies, loss of low rankness, no specific magnitude / phase structure.

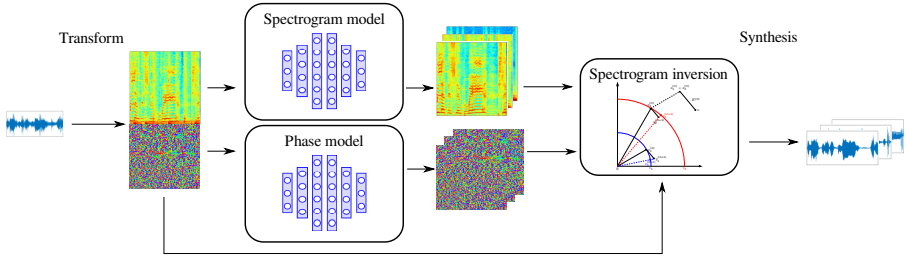
The proposed alternative

Leverage a magnitude / phase decomposition in the STFT domain.



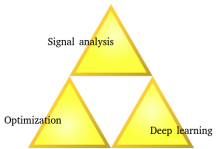
The proposed alternative

Leverage a magnitude / phase decomposition in the STFT domain.



The best of all worlds!

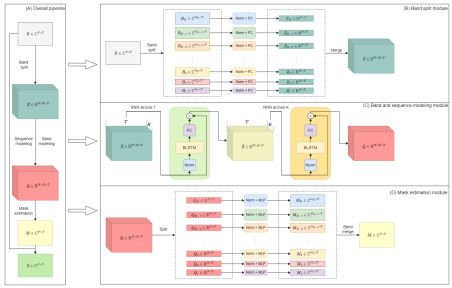
- ▷ Learn the structure of phase via deep phase models.
- ▷ Unfold iterative algorithms into neural networks for time-domain separation.



Spectrogram modeling

Band-Split RNN (BSRNN)

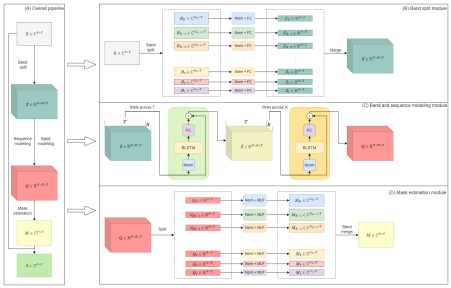
- ▷ A state-of-the-art architecture.
- ▷ Group the frequency channels consistently with the auditory system.
- ▷ Learn dependencies over time and across bands.
  - ✗ Real+imaginary parts processing.



Spectrogram modeling

Band-Split RNN (BSRNN)

- ▷ A state-of-the-art architecture.
- ▷ Group the frequency channels consistently with the auditory system.
- ▷ Learn dependencies over time and across bands.
  - ✗ Real+imaginary parts processing.



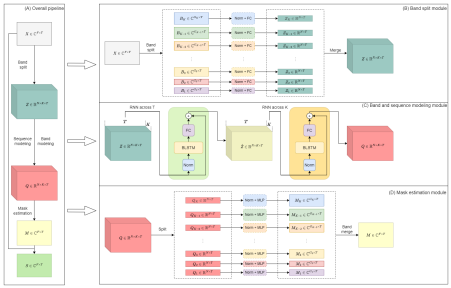
Proposed Magnitude BSRNN (MagBS)

- ✓ Process magnitudes (inputs/outputs).
- ▷ Enforce nonnegative masks with a ReLU.
- ▷ Use an additional light attention mechanism.

Spectrogram modeling

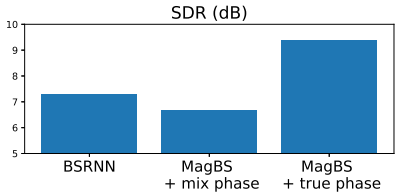
Band-Split RNN (BSRNN)

- ▷ A state-of-the-art architecture.
- ▷ Group the frequency channels consistently with the auditory system.
- ▷ Learn dependencies over time and across bands.
- ✗ Real+imaginary parts processing.



Proposed Magnitude BSRNN (MagBS)

- ✓ Process magnitudes (inputs/outputs).
- ▷ Enforce nonnegative masks with a ReLU.
- ▷ Use an additional light attention mechanism.



Deep phase model

How to acccount for the particular phase structure and properties?

Deep phase model

How to acccount for the particular phase structure and properties?

Recent attempts

- ▷ Generic architectures (MLP, stacks of CNNs).
- ▷ Cumbersome two-stage approaches (modeling the phase *derivatives* + integration scheme).

Deep phase model

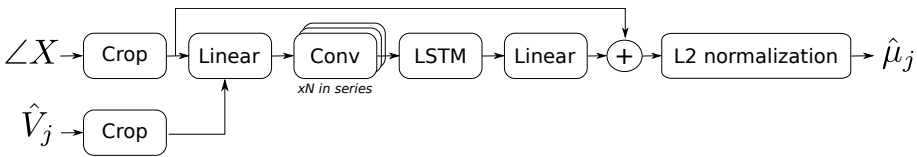
How to acccount for the particular phase structure and properties?

Recent attempts

- ▷ Generic architectures (MLP, stacks of CNNs).
- ▷ Cumbersome two-stage approaches (modeling the phase *derivatives* + integration scheme).

Proposal

- ▷ CRNNs to extend signal models ($\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + l\boldsymbol{\nu}_t$).
- ▷ Residual networks to facilitate learning and alleviate the periodicity issue.

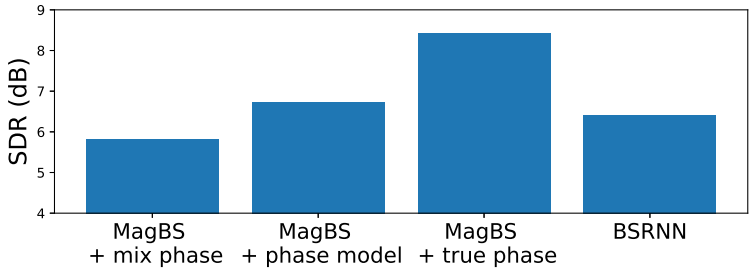


Deep phase model performance

- ▷ The model is first trained using *true* magnitudes.
- ▷ Then the whole $\{\text{MagBS} + \text{phase model}\}$ system is fine-tuned.

Deep phase model performance

- ▷ The model is first trained using *true* magnitudes.
- ▷ Then the whole {MagBS + phase model} system is fine-tuned.
- ▷ Results for the bass track:



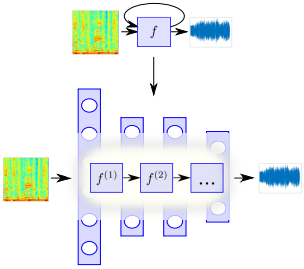
✓ Promising results (magnitude/phase > real/imaginary).

Still some room for improvement!

Unfolding spectrogram inversion

Key idea

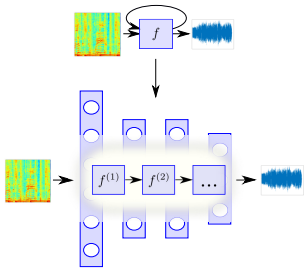
- ▷ Each algorithm's iteration = one layer of a neural network.
- ▷ Train via backpropagation through the unfolded algorithm.



Unfolding spectrogram inversion

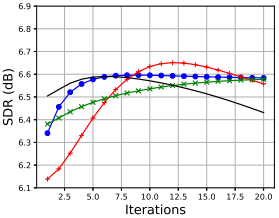
Key idea

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

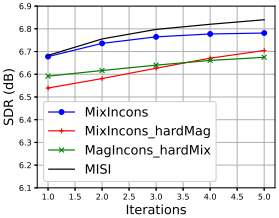


Results

Fixed post-processing



Unfolded algorithm

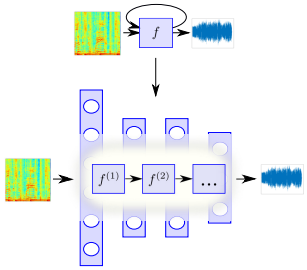


✓ Improved performance over a fixed post-processing, with (almost) no additional parameter.

Unfolding spectrogram inversion

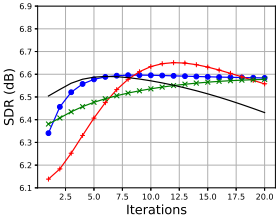
Key idea

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

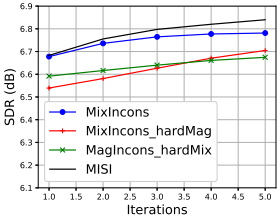


Results

Fixed post-processing



Unfolded algorithm



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

- ▷ Next step: combine it with the deep phase model.

Conclusion

Conclusion

Key messages

- Audio demixing: a fundamental task with many applications.
- Impressive performance in controlled conditions thanks to the advent of deep learning.
- Magnitude/phase processing: an original approach for improved robustness and performance.

Conclusion

Key messages

- ▷ Audio demixing: a fundamental task with many applications.
- ▷ Impressive performance in controlled conditions thanks to the advent of deep learning.
- ▷ Magnitude/phase processing: an original approach for improved robustness and performance.

Perspectives / futur challenges

- ▷ Real-time separation / low-ressource devices (e.g., hearing aids).
- ▷ Generalization to larger / more diverse datasets (beyond research benchmarks).
- ▷ Adpatation to specific sources / instruments / setups.

A link between worlds?

Common methodological tools

- ▷ Matrix and tensor factorization.
- ▷ DNNs for ∼everything.
- ▷ Numerical methods / optimization.

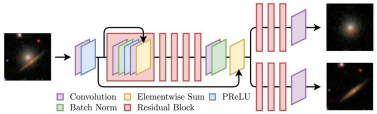
A link between worlds?

Common methodological tools

- ▷ Matrix and tensor factorization.
- ▷ DNNs for ∼everything.
- ▷ Numerical methods / optimization.

Applications

- ▷ Source separation.



Source: Reiman and Göhre, "Deblending galaxy superpositions with branched generative adversarial networks", MNRAS, 2019.

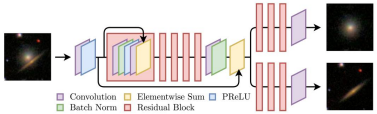
A link between worlds?

Common methodological tools

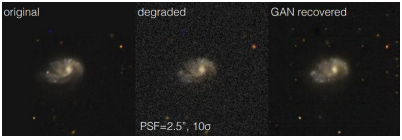
- ▷ Matrix and tensor factorization.
- ▷ DNNs for ∼everything.
- ▷ Numerical methods / optimization.

Applications

- ▷ Source separation.
- ▷ Noise reduction.



Source: Reiman and Göhre, "Deblending galaxy superpositions with branched generative adversarial networks", MNRAS, 2019.



Schawinski et al., "Generative Adversarial Networks recover features in astrophysical images of galaxies beyond the deconvolution limit", MNRAS:Letters, 2017.

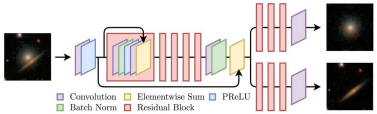
A link between worlds?

Common methodological tools

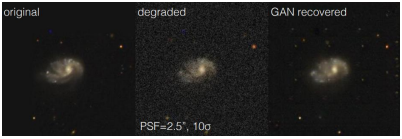
- ▷ Matrix and tensor factorization.
- ▷ DNNs for ∼everything.
- ▷ Numerical methods / optimization.

Applications

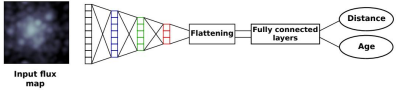
- ▷ Source separation.
- ▷ Noise reduction.
- ▷ Classification / regression.



Source: Reiman and Göhre, "Deblending galaxy superpositions with branched generative adversarial networks", MNRAS, 2019.



Schawinski et al., "Generative Adversarial Networks recover features in astrophysical images of galaxies beyond the deconvolution limit", MNRAS:Letters, 2017.



Chardin and Bianchini, "Predicting images for the dynamics of stellar clusters (π-DOC): a deep learning framework to predict mass, distance, and age of globular clusters", MNRAS, 2021.

Until next time... Thanks!

https://magronp.github.io/

https://github.com/magronp/