Spectrogram Inversion for Audio Source Separation via Consistency, Mixing, and Magnitude Constraints

Paul Magron, Tuomas Virtanen

Université de Lorraine, CNRS, Inria, LORIA, Nancy, France

EUSIPCO2023

Introduction

Audio source separation

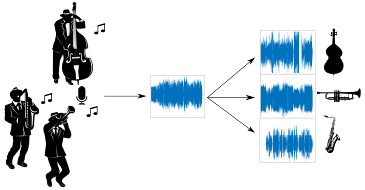
- Audio signals are composed of several constitutive sounds: multiple speakers, background noise, domestic sounds, musical instruments...

Audio source separation

- Audio signals are composed of several constitutive sounds: multiple speakers, background noise, domestic sounds, musical instruments...

Source separation = recovering the sources from the mixture.

- Augmented mixing (from mono to stereo).
- An important preprocessing for many analysis tasks (speech recognition, melody extraction...).

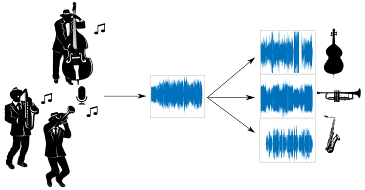


Audio source separation

- Audio signals are composed of several constitutive sounds: multiple speakers, background noise, domestic sounds, musical instruments...

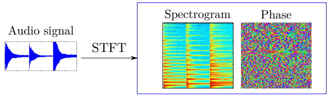
Source separation = recovering the sources from the mixture.

- Augmented mixing (from mono to stereo).
- An important preprocessing for many analysis tasks (speech recognition, melody extraction...).



Framework

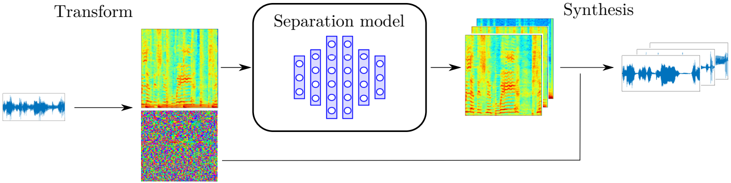
- Monaural signals.
- Short-time Fourier transform (STFT)-domain separation.
- Mixture model: $\mathbf{X} = \sum_{j=1}^{J} \mathbf{S}_j$.



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

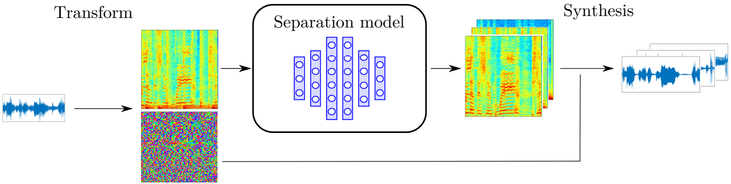
Typical separation pipeline

Nonnegative time-frequency (TF) masking:



Typical separation pipeline

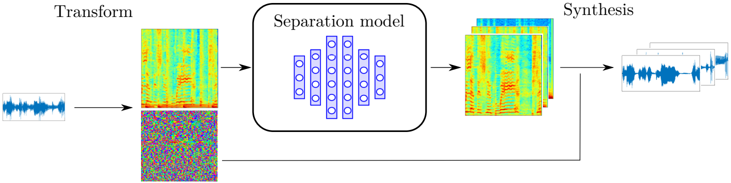
Nonnegative time-frequency (TF) masking:



- ▷ A nonnegative representation is processed (e.g., magnitude or power spectrogram).

Typical separation pipeline

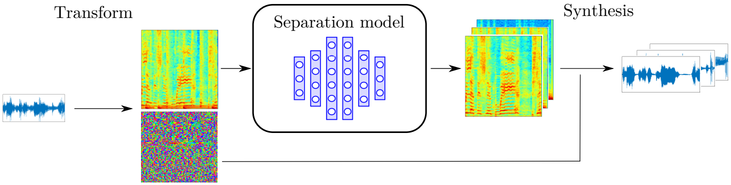
Nonnegative time-frequency (TF) masking:



- A nonnegative representation is processed (e.g., magnitude or power spectrogram).
- The separator is a deep neural network, trained using a (large) dataset with isolated sources.

Typical separation pipeline

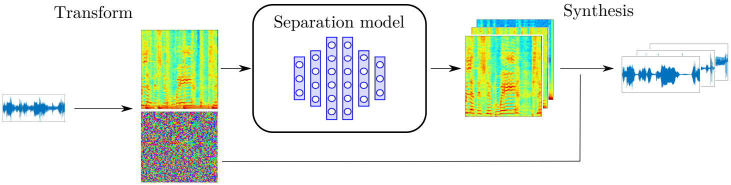
Nonnegative time-frequency (TF) masking:



- A nonnegative representation is processed (e.g., magnitude or power spectrogram).
- The separator is a deep neural network, trained using a (large) dataset with isolated sources.
- The mixture's phase is assigned to each source using a Wiener-like filter or masking process.

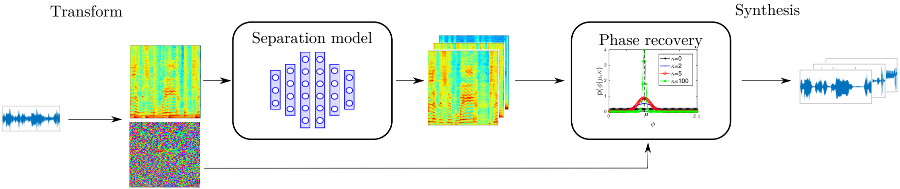
Typical separation pipeline

Nonnegative time-frequency (TF) masking:

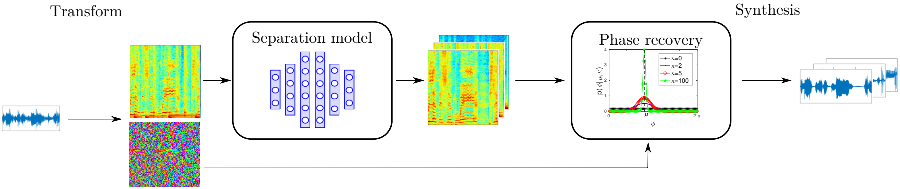


- ▷ A nonnegative representation is processed (e.g., magnitude or power spectrogram).
- ▷ The separator is a deep neural network, trained using a (large) dataset with isolated sources.
- ▷ The mixture's phase is assigned to each source using a Wiener-like filter or masking process.
  - ✗ Issues in sound quality when sources overlap in the TF domain.

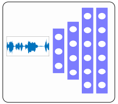
Phase recovery for source separation



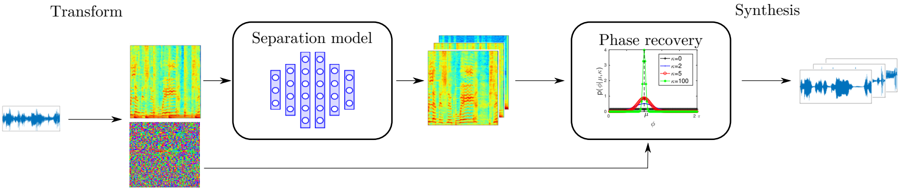
Phase recovery for source separation



Remark: what about current (complex-valued / time-domain) approaches?

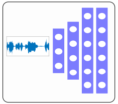


Phase recovery for source separation

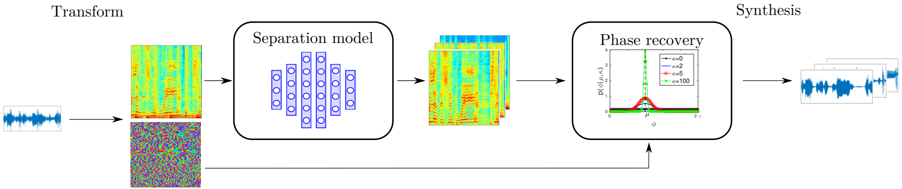


Remark: what about current (complex-valued / time-domain) approaches?

- ✓ State-of-the-art results, alleviate the phase issue.

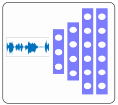


Phase recovery for source separation

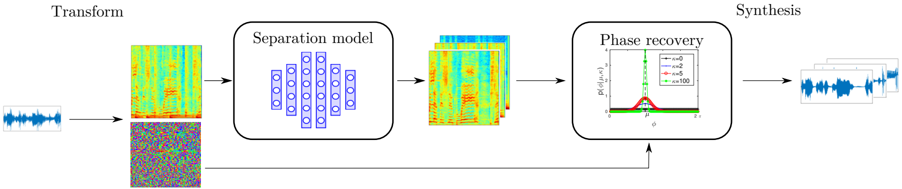


Remark: what about current (complex-valued / time-domain) approaches?

- ✓ State-of-the-art results, alleviate the phase issue.
- ✗ Larger models (more costly), less interpretable, lack robustness.



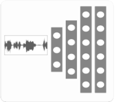
Phase recovery for source separation



Remark: what about current (complex-valued / time-domain) approaches?

✓ State-of-the-art results, alleviate the phase issue.

✗ Larger models (more costly), less interpretable, lack robustness.



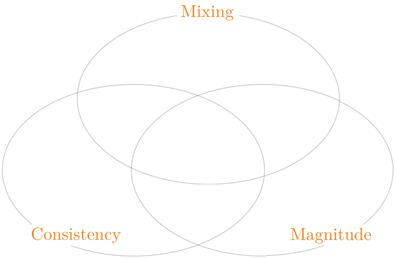
Optimization-based algorithms

- ▷ Preserves the magnitude/phase structure.
- ▷ Allow for time-domain training through deep unfolding.
- ▷ Can be combined with deep phase priors as initialization.

Spectrogram inversion algorithms

Key ingredients to derive such algorithms:

- Important properties in the STFT domain.
- Hard constraints vs. soft penalties.
- Optimization strategy.



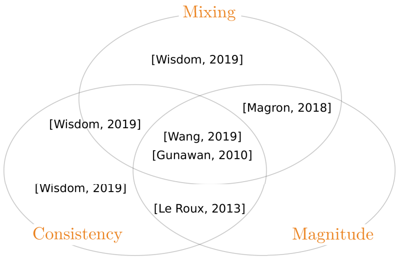
Spectrogram inversion algorithms

Key ingredients to derive such algorithms:

- ▷ Important properties in the STFT domain.
- ▷ Hard constraints vs. soft penalties.
- ▷ Optimization strategy.

Problem

- ▷ Many algorithms in the literature!
- ▷ Which formulation is the most appropriate?



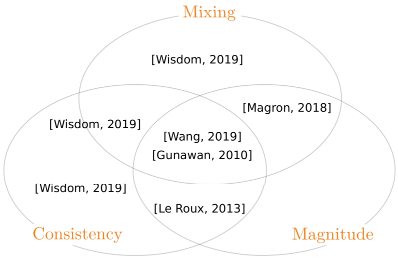
Spectrogram inversion algorithms

Key ingredients to derive such algorithms:

- ▷ Important properties in the STFT domain.
- ▷ Hard constraints vs. soft penalties.
- ▷ Optimization strategy.

Problem

- ▷ Many algorithms in the literature!
- ▷ Which formulation is the most appropriate?



Proposal

A general framework for deriving spectrogram inversion algorithms based on these STFT constraints.

Proposed framework

Overview

Proposed framework

- For each property/objective/constraint, define a loss function (and an auxiliary function).
- Combine them (soft penalties / hard constraints) to formulate optimization problems.
- Derive algorithms that alternate projections on the corresponding constraints subspaces.

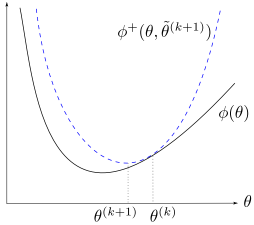
Overview

Proposed framework

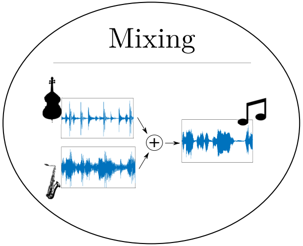
- ▷ For each property/objective/constraint, define a loss function (and an auxiliary function).
- ▷ Combine them (soft penalties / hard constraints) to formulate optimization problems.
- ▷ Derive algorithms that alternate projections on the corresponding constraints subspaces.

Auxiliary function method

- ▷ Considering minimization of ϕ, construct ϕ^+ such that: $\phi(\theta) = \min_{\tilde{\theta}} \phi^+(\theta, \tilde{\theta})$.
- ▷ ϕ is non-increasing when minimizing ϕ^+ with respect to θ and $\tilde{\theta}$ alternately.
  - ✓ Convergence, successfully used in audio, no hyperparameter to tune.

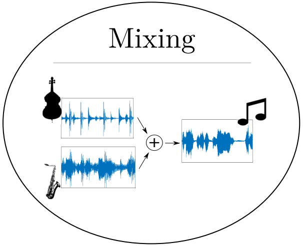


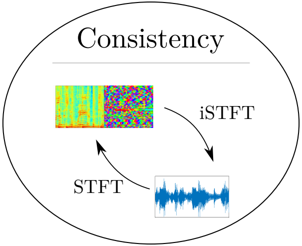
STFT-domain constraints



- Mixing: the estimates should be *conservative* = sum up to the mixture, such that there is no creation/destruction of energy.

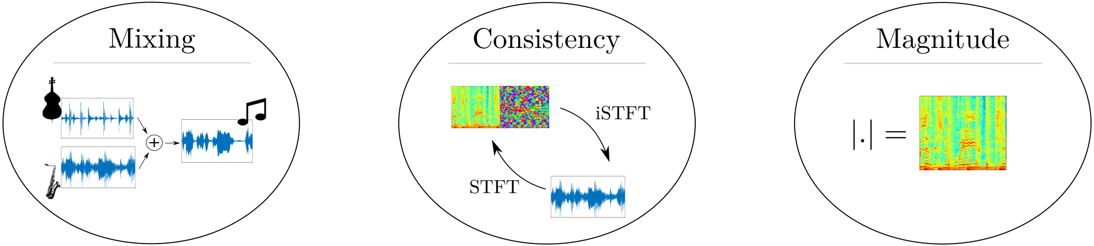
STFT-domain constraints





- Mixing: the estimates should be *conservative* = sum up to the mixture, such that there is no creation/destruction of energy.
- Consistency: the estimates (=complex-valued matrices) should be the STFT of time-domain signals.

STFT-domain constraints

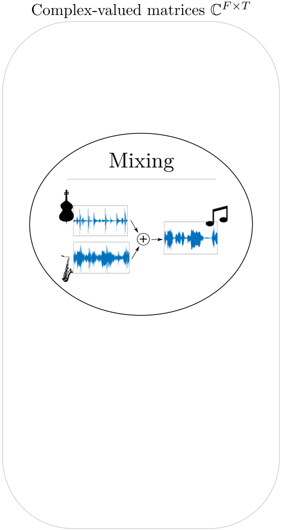


- Mixing: the estimates should be *conservative* = sum up to the mixture, such that there is no creation/destruction of energy.
- Consistency: the estimates (=complex-valued matrices) should be the STFT of time-domain signals.
- Magnitude match: the estimates' magnitude should remain close to the output of the DNN computed beforehand.

Mixing constraint

Loss function that promotes conservative estimates:

$$h(\mathbf{S}) = ||\mathbf{X} - \sum_j \mathbf{S}_j||^2$$



Mixing constraint

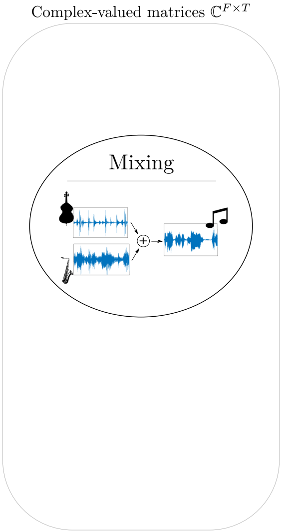
Loss function that promotes conservative estimates:

$$h(\mathbf{S}) = ||\mathbf{X} - \sum_j \mathbf{S}_j||^2$$

Auxiliary function

- ▷ Auxiliary parameters $\mathbf{Y}$ such that $\sum_j \mathbf{Y}_j = \mathbf{X}$.
- ▷ Positive weights $\mathbf{\Lambda}_j$ such that $\sum_j \lambda_{j,f,t} = 1$.
- ▷ Then the following is an auxiliary function for h:

$$h^+(\mathbf{S}, \mathbf{Y}) = \sum_{j,f,t} \frac{|y_{j,f,t} - s_{j,f,t}|^2}{\lambda_{j,f,t}}$$



Mixing constraint

Loss function that promotes conservative estimates:

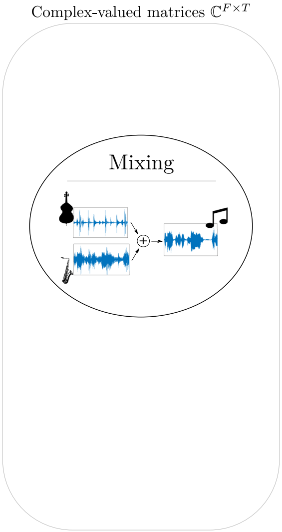
$$h(\mathbf{S}) = ||\mathbf{X} - \sum_j \mathbf{S}_j||^2$$

Auxiliary function

- ▷ Auxiliary parameters $\mathbf{Y}$ such that $\sum_j \mathbf{Y}_j = \mathbf{X}$.
- ▷ Positive weights $\mathbf{\Lambda}_j$ such that $\sum_j \lambda_{j,f,t} = 1$.
- ▷ Then the following is an auxiliary function for h:

$$h^+(\mathbf{S}, \mathbf{Y}) = \sum_{j,f,t} \frac{|y_{j,f,t} - s_{j,f,t}|^2}{\lambda_{j,f,t}}$$

Auxiliary parameters update: $\mathbf{Y}_j = \mathbf{S}_j + \mathbf{\Lambda}_j \odot (\mathbf{X} - \sum_k \mathbf{S}_k)$



Mixing constraint

Loss function that promotes conservative estimates:

$$h(\mathbf{S}) = ||\mathbf{X} - \sum_j \mathbf{S}_j||^2$$

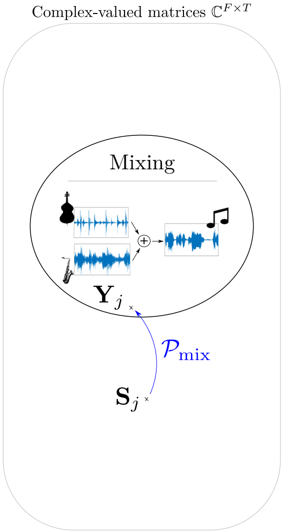
Auxiliary function

- ▷ Auxiliary parameters $\mathbf{Y}$ such that $\sum_j \mathbf{Y}_j = \mathbf{X}$.
- ▷ Positive weights $\mathbf{\Lambda}_j$ such that $\sum_j \lambda_{j,f,t} = 1$.
- ▷ Then the following is an auxiliary function for h:

$$h^+(\mathbf{S}, \mathbf{Y}) = \sum_{j,f,t} \frac{|y_{j,f,t} - s_{j,f,t}|^2}{\lambda_{j,f,t}}$$

Auxiliary parameters update: $\mathbf{Y}_j = \mathbf{S}_j + \mathbf{\Lambda}_j \odot (\mathbf{X} - \sum_k \mathbf{S}_k)$

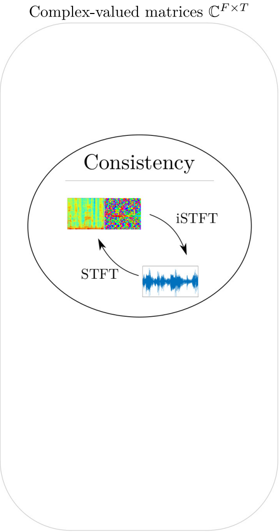
- ▷ Defines a projector $\mathcal{P}_{\text{mix}}$ onto the subspace of matrices complying with the mixing constraint.



Consistency constraint

Loss function that promotes consistent estimates:

$$i(\mathbf{S}) = \sum_j ||\mathbf{S}_j - \mathcal{G}(\mathbf{S}_j)||^2 \text{ with } \mathcal{G} = \mathsf{STFT} \circ \mathsf{iSTFT}$$



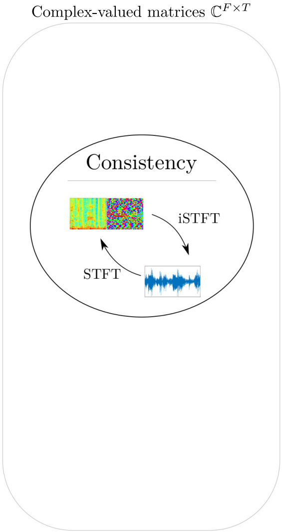
Consistency constraint

Loss function that promotes consistent estimates:

$$i(\mathbf{S}) = \sum_j ||\mathbf{S}_j - \mathcal{G}(\mathbf{S}_j)||^2 \text{ with } \mathcal{G} = \text{STFT} \circ \text{iSTFT}$$

Auxiliary function

- $\mathcal{G}(\mathbf{S}_j)$ is the closest consistent matrix to $\mathbf{S}_j$.
- Then $i^+(\mathbf{S}, \mathbf{Z}) = \sum_j ||\mathbf{S}_j - \mathbf{Z}_j||^2$ (where $\mathbf{Z}_j \in \text{Im(STFT)}$) is an auxiliary function for i.



Consistency constraint

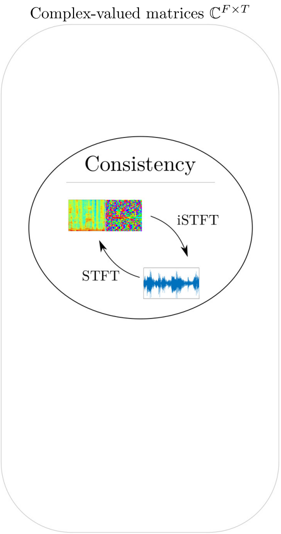
Loss function that promotes consistent estimates:

$$i(\mathbf{S}) = \sum_j ||\mathbf{S}_j - \mathcal{G}(\mathbf{S}_j)||^2 \text{ with } \mathcal{G} = \text{STFT} \circ \text{iSTFT}$$

Auxiliary function

- ▷ $\mathcal{G}(\mathbf{S}_j)$ is the closest consistent matrix to $\mathbf{S}_j$.
- ▷ Then $i^+(\mathbf{S}, \mathbf{Z}) = \sum_j ||\mathbf{S}_j - \mathbf{Z}_j||^2$ (where $\mathbf{Z}_j \in \text{Im(STFT)}$) is an auxiliary function for i.

Auxiliary parameters update: $\mathbf{Z}_j = \mathcal{G}(\mathbf{S}_j)$



Consistency constraint

Loss function that promotes consistent estimates:

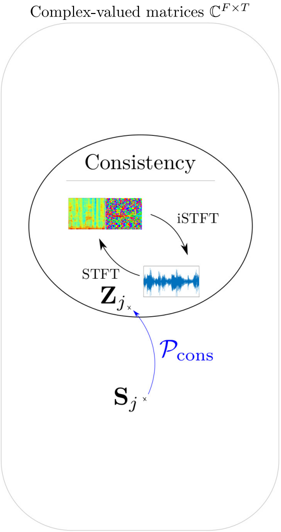
$$i(\mathbf{S}) = \sum_j ||\mathbf{S}_j - \mathcal{G}(\mathbf{S}_j)||^2 \text{ with } \mathcal{G} = \text{STFT} \circ \text{iSTFT}$$

Auxiliary function

- $\mathcal{G}(\mathbf{S}_j)$ is the closest consistent matrix to $\mathbf{S}_j$.
- Then $i^+(\mathbf{S}, \mathbf{Z}) = \sum_j ||\mathbf{S}_j - \mathbf{Z}_j||^2$ (where $\mathbf{Z}_j \in \text{Im(STFT)}$) is an auxiliary function for i.

Auxiliary parameters update: $\mathbf{Z}_j = \mathcal{G}(\mathbf{S}_j)$

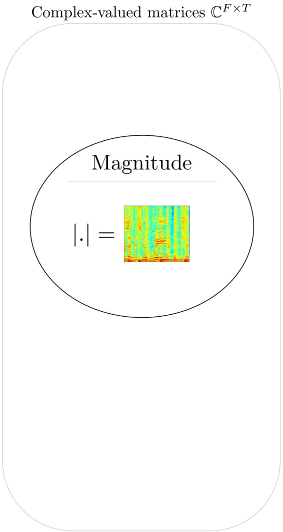
- Defines a projector $\mathcal{P}_{\text{cons}}$ onto the subspace of consistent matrices.



Magnitude constraint

Loss function that ensures the estimates' magnitudes remain close to the target value $\mathbf{V}_j$ estimated beforehand (e.g., using a DNN):

$$m(\mathbf{S}) = \sum_j || |\mathbf{S}_j| - \mathbf{V}_j ||^2$$



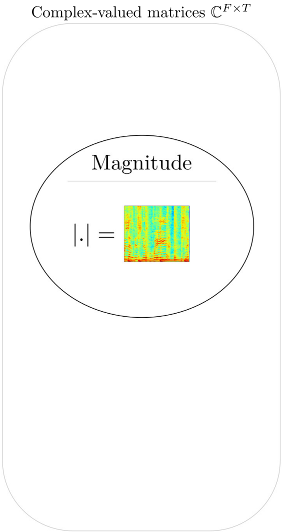
Magnitude constraint

Loss function that ensures the estimates' magnitudes remain close to the target value $\mathbf{V}_j$ estimated beforehand (e.g., using a DNN):

$$m(\mathbf{S}) = \sum_j |||\mathbf{S}_j| - \mathbf{V}_j||^2$$

Auxiliary function

- Auxiliary parameters $\mathbf{U}$ such that $|\mathbf{U}_j| = \mathbf{V}_j$.
- $m^+(\mathbf{S}, \mathbf{Z}) = \sum_j ||\mathbf{S}_j - \mathbf{U}_j||^2$ is an auxiliary function for m.



Magnitude constraint

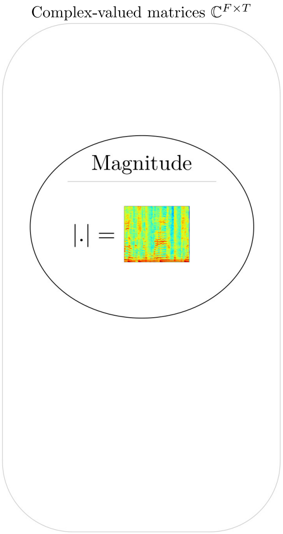
Loss function that ensures the estimates' magnitudes remain close to the target value $\mathbf{V}_j$ estimated beforehand (e.g., using a DNN):

$$m(\mathbf{S}) = \sum_j |||\mathbf{S}_j| - \mathbf{V}_j||^2$$

Auxiliary function

- ▷ Auxiliary parameters $\mathbf{U}$ such that $|\mathbf{U}_j| = \mathbf{V}_j$.
- ▷ $m^+(\mathbf{S}, \mathbf{Z}) = \sum_j ||\mathbf{S}_j - \mathbf{U}_j||^2$ is an auxiliary function for m.

Auxiliary parameters update: $\mathbf{U}_j = \dfrac{\mathbf{S}_j}{|\mathbf{S}_j|} \odot \mathbf{V}_j$



Magnitude constraint

Loss function that ensures the estimates' magnitudes remain close to the target value $\mathbf{V}_j$ estimated beforehand (e.g., using a DNN):

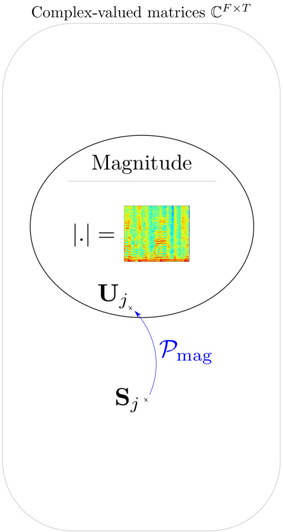
$$m(\mathbf{S}) = \sum_j || |\mathbf{S}_j| - \mathbf{V}_j ||^2$$

Auxiliary function

- ▷ Auxiliary parameters $\mathbf{U}$ such that $|\mathbf{U}_j| = \mathbf{V}_j$.
- ▷ $m^+(\mathbf{S}, \mathbf{Z}) = \sum_j ||\mathbf{S}_j - \mathbf{U}_j||^2$ is an auxiliary function for m.

Auxiliary parameters update: $\mathbf{U}_j = \dfrac{\mathbf{S}_j}{|\mathbf{S}_j|} \odot \mathbf{V}_j$

- ▷ Defines a projector $\mathcal{P}_{\text{mag}}$ onto the subspace of matrices whose magnitude equals the target value.



Algorithm derivation example: problem setting

Main problem: optimize the mixing objective + soft consistency penalty + hard magnitude constraint (call that `Mix+Incons_hardMag`).

$$\min_{\mathbf{S}} h(\mathbf{S}) + \sigma i(\mathbf{S}) \text{ such that } |\mathbf{S}_j| = \mathbf{V}_j$$

Algorithm derivation example: problem setting

Main problem: optimize the mixing objective + soft consistency penalty + hard magnitude constraint (call that `Mix+Incons_hardMag`).

$$\min_{\mathbf{S}} h(\mathbf{S}) + \sigma i(\mathbf{S}) \text{ such that } |\mathbf{S}_j| = \mathbf{V}_j$$

Using our auxiliary function framework, this rewrites:

$$\min_{\mathbf{S},\mathbf{Y},\mathbf{Z}} h^+(\mathbf{S},\mathbf{Y}) + \sigma i^+(\mathbf{S},\mathbf{Z}) \quad \text{such that} \quad \begin{cases} |\mathbf{S}_j| = \mathbf{V}_j \\ \sum_j \mathbf{Y}_j = \mathbf{X} \\ \mathbf{Z}_j \in \mathrm{Im}(\mathsf{STFT}) \end{cases}$$

Algorithm derivation example: problem setting

Main problem: optimize the mixing objective + soft consistency penalty + hard magnitude constraint (call that `Mix+Incons_hardMag`).

$$\min_{\mathbf{S}} h(\mathbf{S}) + \sigma i(\mathbf{S}) \text{ such that } |\mathbf{S}_j| = \mathbf{V}_j$$

Using our auxiliary function framework, this rewrites:

$$\min_{\mathbf{S},\mathbf{Y},\mathbf{Z}} h^+(\mathbf{S},\mathbf{Y}) + \sigma i^+(\mathbf{S},\mathbf{Z}) \quad \text{such that} \quad \begin{cases} |\mathbf{S}_j| = \mathbf{V}_j \\ \sum_j \mathbf{Y}_j = \mathbf{X} \\ \mathbf{Z}_j \in \mathrm{Im}(\mathrm{STFT}) \end{cases}$$

- ▷ Auxiliary parameters updates ($\mathbf{Y}$ and $\mathbf{Z}$) are already known.

Algorithm derivation example: problem setting

Main problem: optimize the mixing objective + soft consistency penalty + hard magnitude constraint (call that `Mix+Incons_hardMag`).

$$\min_{\mathbf{S}} h(\mathbf{S}) + \sigma i(\mathbf{S}) \text{ such that } |\mathbf{S}_j| = \mathbf{V}_j$$

Using our auxiliary function framework, this rewrites:

$$\min_{\mathbf{S},\mathbf{Y},\mathbf{Z}} h^+(\mathbf{S},\mathbf{Y}) + \sigma i^+(\mathbf{S},\mathbf{Z}) \quad \text{such that} \quad \begin{cases} |\mathbf{S}_j| = \mathbf{V}_j \\ \sum_j \mathbf{Y}_j = \mathbf{X} \\ \mathbf{Z}_j \in \mathrm{Im}(\mathrm{STFT}) \end{cases}$$

- ▷ Auxiliary parameters updates ($\mathbf{Y}$ and $\mathbf{Z}$) are already known.
- ▷ So let's focus on the update on $\mathbf{S}$.

Algorithm derivation example: update

New problem

- Incorporate the hard constraint using the method of Lagrange multipliers.
- Find a critical point for:

$$h^+(\mathbf{S}, \mathbf{Y}) + \sigma i^+(\mathbf{S}, \mathbf{Z}) + \sum_{j,f,t} \delta_{j,f,t}(|s_{j,f,t}|^2 - v_{j,f,t}^2)$$

Algorithm derivation example: update

New problem

- Incorporate the hard constraint using the method of Lagrange multipliers.
- Find a critical point for:

$$h^{+}(\mathbf{S}, \mathbf{Y}) + \sigma i^{+}(\mathbf{S}, \mathbf{Z}) + \sum_{j,f,t} \delta_{j,f,t}(|s_{j,f,t}|^2 - v_{j,f,t}^2)$$

Update

- Set the partial derivative with respect to $\mathbf{S}$ at 0 and solve:

$$\mathbf{S}_j = \frac{\mathbf{Y}_j + \sigma \mathbf{\Lambda}_j \odot \mathbf{Z}_j}{|\mathbf{Y}_j + \sigma \mathbf{\Lambda}_j \odot \mathbf{Z}_j|} \odot \mathbf{V}_j$$

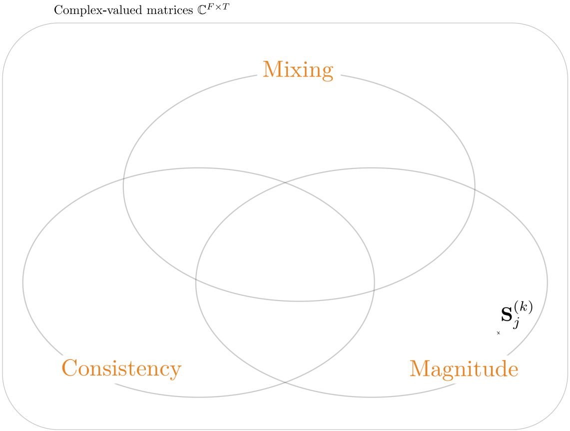
- Generalizes particular cases from the literature ($\sigma = 0$ and $\sigma = +\infty$).

Algorithm derivation example: illustration

Compact update rule using the projectors: $\mathcal{P}_{\mathsf{mag}}\left(\mathcal{P}_{\mathsf{mix}}(\mathbf{S}) + \sigma\mathbf{\Lambda} \odot \mathcal{P}_{\mathsf{cons}}(\mathbf{S})\right)$

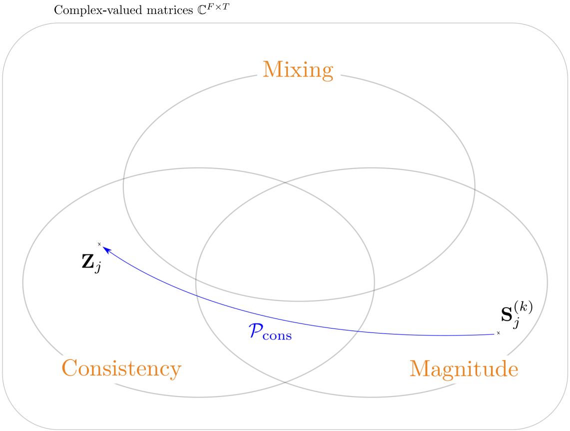
Algorithm derivation example: illustration

Compact update rule using the projectors: $\mathcal{P}_{\textsf{mag}}\left(\mathcal{P}_{\textsf{mix}}(\mathbf{S}) + \sigma\mathbf{\Lambda}\odot\mathcal{P}_{\textsf{cons}}(\mathbf{S})\right)$



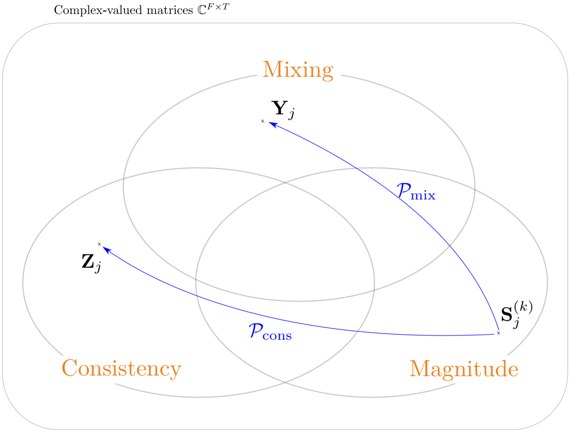
Algorithm derivation example: illustration

Compact update rule using the projectors: $\mathcal{P}_{\mathsf{mag}}\left(\mathcal{P}_{\mathsf{mix}}(\mathbf{S}) + \sigma\boldsymbol{\Lambda} \odot \mathcal{P}_{\mathsf{cons}}(\mathbf{S})\right)$



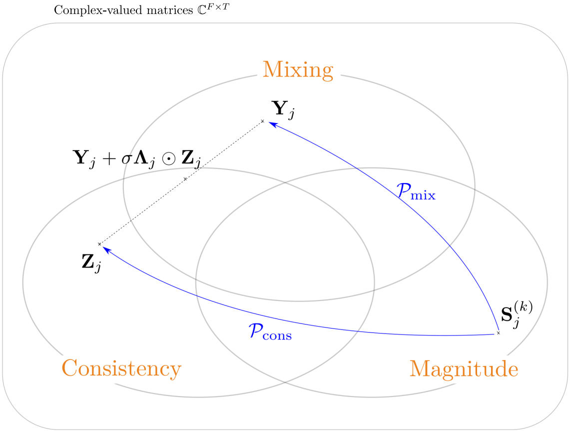
Algorithm derivation example: illustration

Compact update rule using the projectors: $\mathcal{P}_{\mathsf{mag}}\left(\mathcal{P}_{\mathsf{mix}}(\mathbf{S}) + \sigma\boldsymbol{\Lambda} \odot \mathcal{P}_{\mathsf{cons}}(\mathbf{S})\right)$



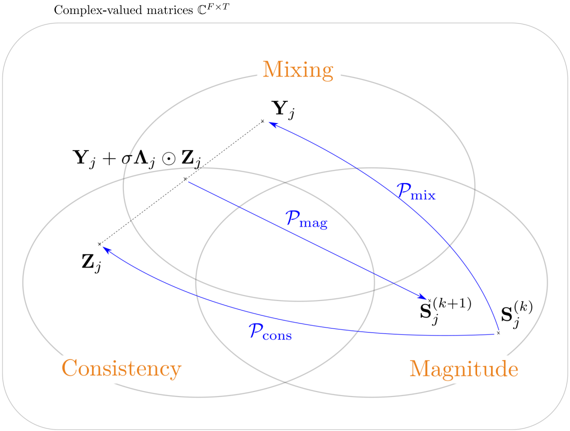
Algorithm derivation example: illustration

Compact update rule using the projectors: $\mathcal{P}_{\mathsf{mag}}\left(\mathcal{P}_{\mathsf{mix}}(\mathbf{S}) + \sigma\mathbf{\Lambda} \odot \mathcal{P}_{\mathsf{cons}}(\mathbf{S})\right)$



Algorithm derivation example: illustration

Compact update rule using the projectors: $\mathcal{P}_{\text{mag}}\left(\mathcal{P}_{\text{mix}}(\mathbf{S}) + \sigma\boldsymbol{\Lambda} \odot \mathcal{P}_{\text{cons}}(\mathbf{S})\right)$



Other algorithms

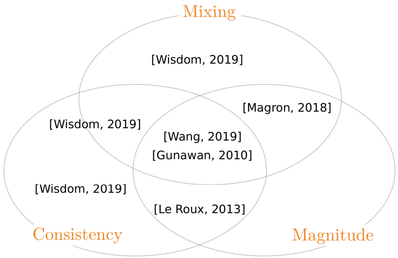
- Check the paper for all problem formulations / update schemes...
- ... and the supplementary material for all the mathematical derivation.

Other algorithms

- ▷ Check the paper for all problem formulations / update schemes...
- ▷ ... and the supplementary material for all the mathematical derivation.

`MISI*`	$\mathcal{P}_{\text{mix}}(\mathcal{P}_{\text{mag}}(\mathcal{P}_{\text{cons}}(\mathbf{S})))$
`Mix+Incons`	$\frac{1}{1+\sigma\mathbf{\Lambda}} \odot (\mathcal{P}_{\text{mix}}(\mathbf{S}) + \sigma\mathbf{\Lambda} \odot \mathcal{P}_{\text{cons}}(\mathbf{S}))$
`Mix+Incons_hardMag`	$\mathcal{P}_{\text{mag}}(\mathcal{P}_{\text{mix}}(\mathbf{S}) + \sigma\mathbf{\Lambda} \odot \mathcal{P}_{\text{cons}}(\mathbf{S}))$
`Incons_hardMix`	$\mathcal{P}_{\text{mix}}(\mathcal{P}_{\text{cons}}(\mathbf{S}))$
`Mag+Incons_hardMix`	$\mathcal{P}_{\text{mix}}\left(\frac{1}{1+\sigma}(\mathcal{P}_{\text{mag}}(\mathbf{S}) + \sigma\mathcal{P}_{\text{cons}}(\mathbf{S}))\right)$

* Multiple Input Spectrogram Inversion

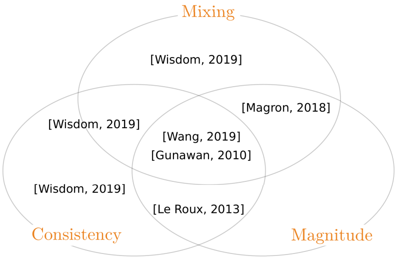


Other algorithms

- ▷ Check the paper for all problem formulations / update schemes...
- ▷ ... and the supplementary material for all the mathematical derivation.

`MISI*`	$\mathcal{P}_{\text{mix}}(\mathcal{P}_{\text{mag}}(\mathcal{P}_{\text{cons}}(\mathbf{S})))$
`Mix+Incons`	$\frac{1}{1+\sigma\mathbf{\Lambda}} \odot (\mathcal{P}_{\text{mix}}(\mathbf{S}) + \sigma\mathbf{\Lambda} \odot \mathcal{P}_{\text{cons}}(\mathbf{S}))$
`Mix+Incons_hardMag`	$\mathcal{P}_{\text{mag}}(\mathcal{P}_{\text{mix}}(\mathbf{S}) + \sigma\mathbf{\Lambda} \odot \mathcal{P}_{\text{cons}}(\mathbf{S}))$
`Incons_hardMix`	$\mathcal{P}_{\text{mix}}(\mathcal{P}_{\text{cons}}(\mathbf{S}))$
`Mag+Incons_hardMix`	$\mathcal{P}_{\text{mix}}\left(\frac{1}{1+\sigma}(\mathcal{P}_{\text{mag}}(\mathbf{S}) + \sigma\mathcal{P}_{\text{cons}}(\mathbf{S}))\right)$

* Multiple Input Spectrogram Inversion



Some problem formulations / algorithms are not reported: ill-posed (conflicting constraints), impractical (2 redundant soft penalties), updates that only affect magnitude...

Experiments

Protocol

Task: speech enhancement ($J = 2$)

- ▷ Clean speech (VoiceBank) + noise (DEMAND: living room, bus, and public square noises).
- ▷ Mixtures at various input SNR (iSNR): -10, 0, and 10 dB.
- ▷ 100 mixtures (50/50 for validation/test).

Magnitude estimation

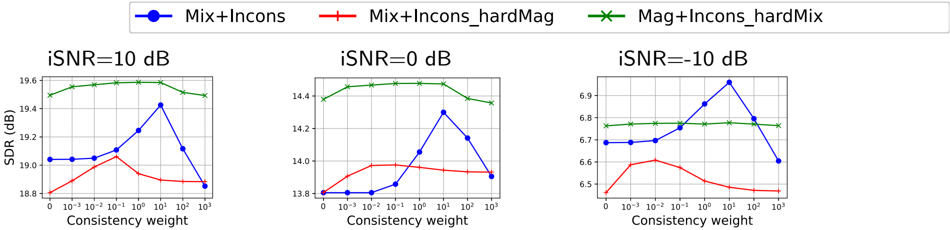
- ▷ Open-Unmix: a freely available BLSTM network (trained on different speakers and noises).
- ▷ In practice, magnitudes are estimated more accurately as the iSNR increases.

Methods

- ▷ Initialization with an amplitude mask (AM) = estimated magnitude + mixture's phase.
- ▷ MISI is a widely-used baseline algorithm.
- ▷ The consistency weight σ and number of iterations are tuned on the validation set.

Separation quality measured with the speech signal-to-distortion ratio (SDR).

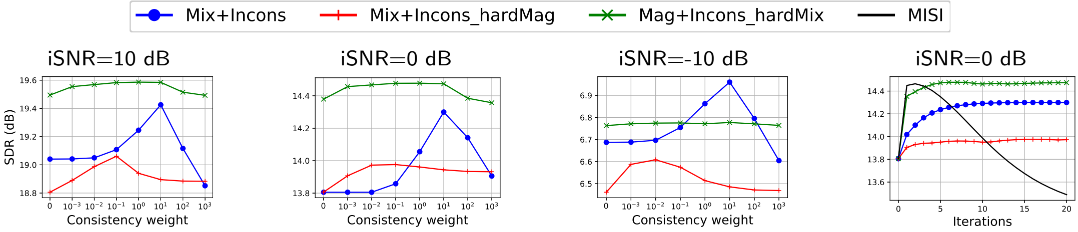
Validation results



Consistency weight

- ▷ SDR peak: adjusting σ maximizes the performance.
- ▷ Our general framework $>$ particular cases ($\sigma = 0$ or $+\infty$) corresponding to existing algorithms.

Validation results



Consistency weight

- ▷ SDR peak: adjusting σ maximizes the performance.
- ▷ Our general framework $>$ particular cases ($\sigma = 0$ or $+\infty$) corresponding to existing algorithms.

Iterations

- ▷ MISI reaches its peak performance after very few iterations.
- ▷ Alternative algorithms are more stable / easier to tune.
- ▷ For a fair comparison, use an algorithm-specific number of iterations (often overlooked).

Test results

	iSNR$= 10$ dB	iSNR$= 0$ dB	iSNR$= -10$ dB
AM	18.7	13.5	7.7
MISI	**19.6**	**14.1**	**7.7**

Test results

	iSNR$= 10$ dB	iSNR$= 0$ dB	iSNR$= -10$ dB
`AM`	18.7	13.5	7.7
`MISI`	**19.6**	**14.1**	7.7
`Mag+Incons_hardMix`	**19.6**	**14.1**	7.7

▷ `Mag+Incons_hardMix`: interesting alternative to MISI (same performance, stable over iterations).

Test results

	iSNR$= 10$ dB	iSNR$= 0$ dB	iSNR$= -10$ dB
`AM`	18.7	13.5	7.7
`MISI`	**19.6**	**14.1**	7.7
`Mag+Incons_hardMix`	**19.6**	**14.1**	7.7
`Incons_hardMix`	**19.6**	13.9	7.5

- ▷ `Mag+Incons_hardMix`: interesting alternative to MISI (same performance, stable over iterations).
- ▷ `Incons_hardMix`: the performance degrades as the iSNR decreases.

Test results

	iSNR= 10 dB	iSNR= 0 dB	iSNR= -10 dB
`AM`	18.7	13.5	7.7
`MISI`	**19.6**	**14.1**	7.7
`Mag+Incons_hardMix`	**19.6**	**14.1**	7.7
`Incons_hardMix`	**19.6**	13.9	7.5
`Mix+Incons_hardMag`	18.7	13.8	**7.9**

- `Mag+Incons_hardMix`: interesting alternative to MISI (same performance, stable over iterations).
- `Incons_hardMix`: the performance degrades as the iSNR decreases.
- `Mix+Incons_hardMag` $>$ MISI at low iSNR, but not at high iSNR ($\neq$ from previous studies: optimized number of iterations and different magnitude estimation technique).

Test results

	iSNR= 10 dB	iSNR= 0 dB	iSNR= −10 dB
AM	18.7	13.5	7.7
MISI	**19.6**	**14.1**	**7.7**
Mag+Incons_hardMix	**19.6**	**14.1**	7.7
Incons_hardMix	**19.6**	13.9	7.5
Mix+Incons_hardMag	18.7	13.8	7.9
Mix+Incons	19.3	13.7	**8.1**

- Mag+Incons_hardMix: interesting alternative to MISI (same performance, stable over iterations).
- Incons_hardMix: the performance degrades as the iSNR decreases.
- Mix+Incons_hardMag $>$ MISI at low iSNR, but not at high iSNR ($\neq$ from previous studies: optimized number of iterations and different magnitude estimation technique).
- Mix+Incons: mitigates the SDR drop at high iSNR + boosts the performance at low iSNR.

Conclusion

Main contribution

A general framework for deriving spectrogram inversion algorithms for source separation.

- ▷ Encompasses many existing techniques from the literature.
- ▷ Some novel algorithms are interesting alternatives.

♫ `https://github.com/magronp/spectrogram-inversion`

Future research / work in progress:

- ▷ Unfold these algorithms into neural networks for time-domain separation.
- ▷ Combine them with deep phase priors.
- ▷ Application to music / speech separation.