MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Learning-Based Approaches to Speech Enhancement and Separation

Le Roux, J.; Vincent, E.; Erdogan, H.

TR2016-113 September 2016

Abstract

Being able to isolate a target speech signal from background signals is of direct importance for telephony, hands-free communication and audio surveillance, and it is also critical as a pre-processing step in applications such as voice activity detection, automatic speaker identification, and most importantly automatic speech recognition (ASR) in challenging environments. While speech enhancement and separation methods originally did not rely on training, there has recently been an explosion in the use of machine learning based methods that exploit large amounts of training data. This tutorial will present a broad overview of these methods, analyzing the insights that can be gained from the pre-deep-learning era of graphical modeling and NMF approaches, then diving into an in-depth presentation of recent deep learning approaches encompassing single-channel methods, multi-channel methods, and new directions.

2016 Interspeech Tutorials

Learning-Based Approaches to Speech Enhancement and Separation

Jonathan Le Roux, Mitsubishi Electric Research Labs
Emmanuel Vincent, Inria
Hakan Erdogan, Sabancı University & Microsoft Research

General overview

Speaker: Emmanuel Vincent

Speech separation and enhancement

What is speech enhancement?

- extract target speech signals from a recording,
- remove other speech sounds and noises (separation),
- remove reverberation (dereverberation).

In this tutorial, focus on separation.

What is it used for?

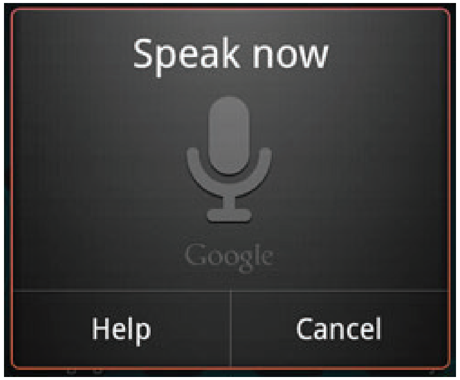
- listen to separated sources,
- remix them,
- retrieve information.

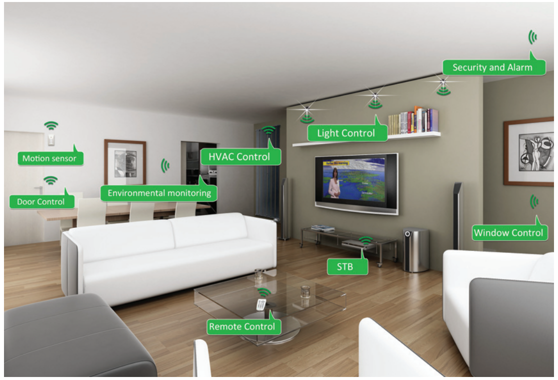
Applications to spoken communication

Enhance speech for

- mobile phones,
- hands-free phones,
- hearing aids. . .

Applications to human machine interfaces



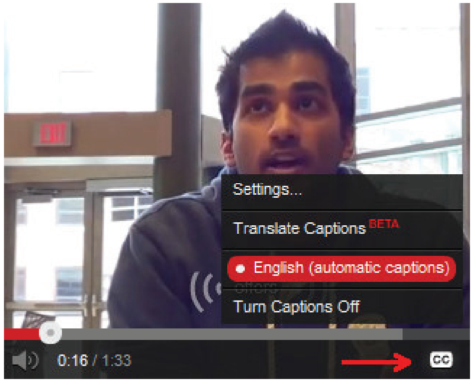


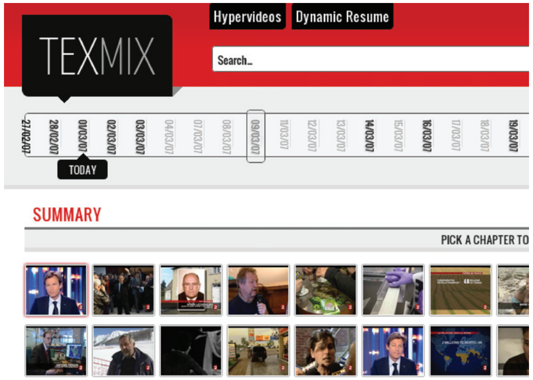
Voice command for

- personal assistants,
- home automation,
- robots. . .

Includes speech recognition, speaker recognition, paralinguistics. . .

Applications to spoken documents





Retrieve

- TV/radio/movie contents,
- personal videos...

Includes language recognition, speech recognition, speaker diarization, keyword spotting...

Time-frequency domain processing

Separation originally formulated as a linear inverse problem:

$$\mathbf{x}(t) = \sum_{\tau=0}^{\infty} \mathbf{A}(\tau)\,\mathbf{s}(t-\tau)$$

$\mathbf{x}(t)$: $I \times 1$ mixture signal
$\mathbf{s}(t)$: $J \times 1$ point source signals
$\mathbf{A}(\tau)$: $I \times J$ matrix of impulse responses
t: discrete time

Replaced by the more general formulation:

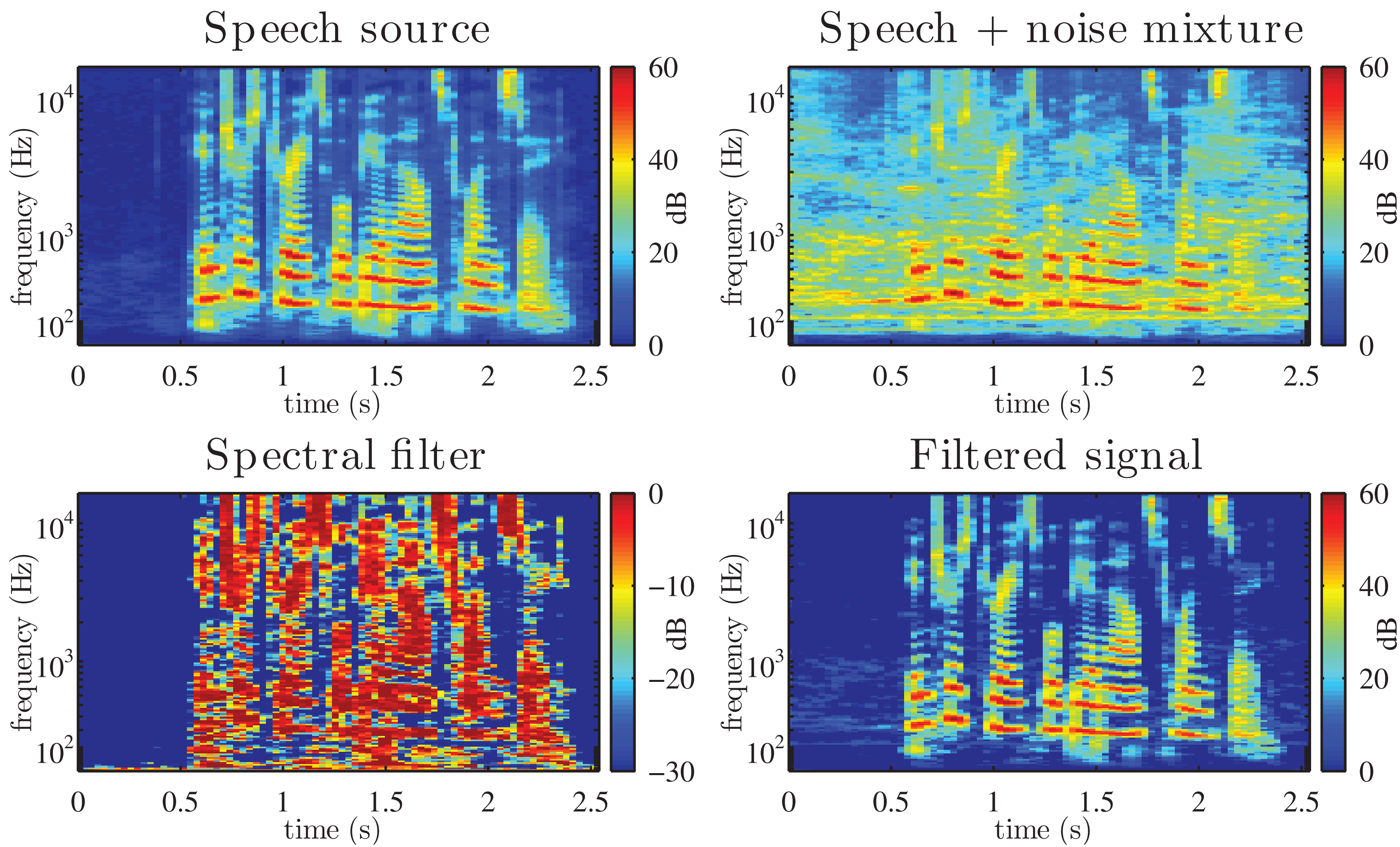
$$\mathbf{x}(n, f) = \sum_{j=1}^{J} \mathbf{c}_j(n, f)$$

$\mathbf{c}_j(n, f)$: $I \times 1$ spatial image of source j (can be diffuse)
n: time frame
f: frequency bin

Goal: filter the signal $\mathbf{x}(n, f)$ into the different sources in each time-frequency bin.

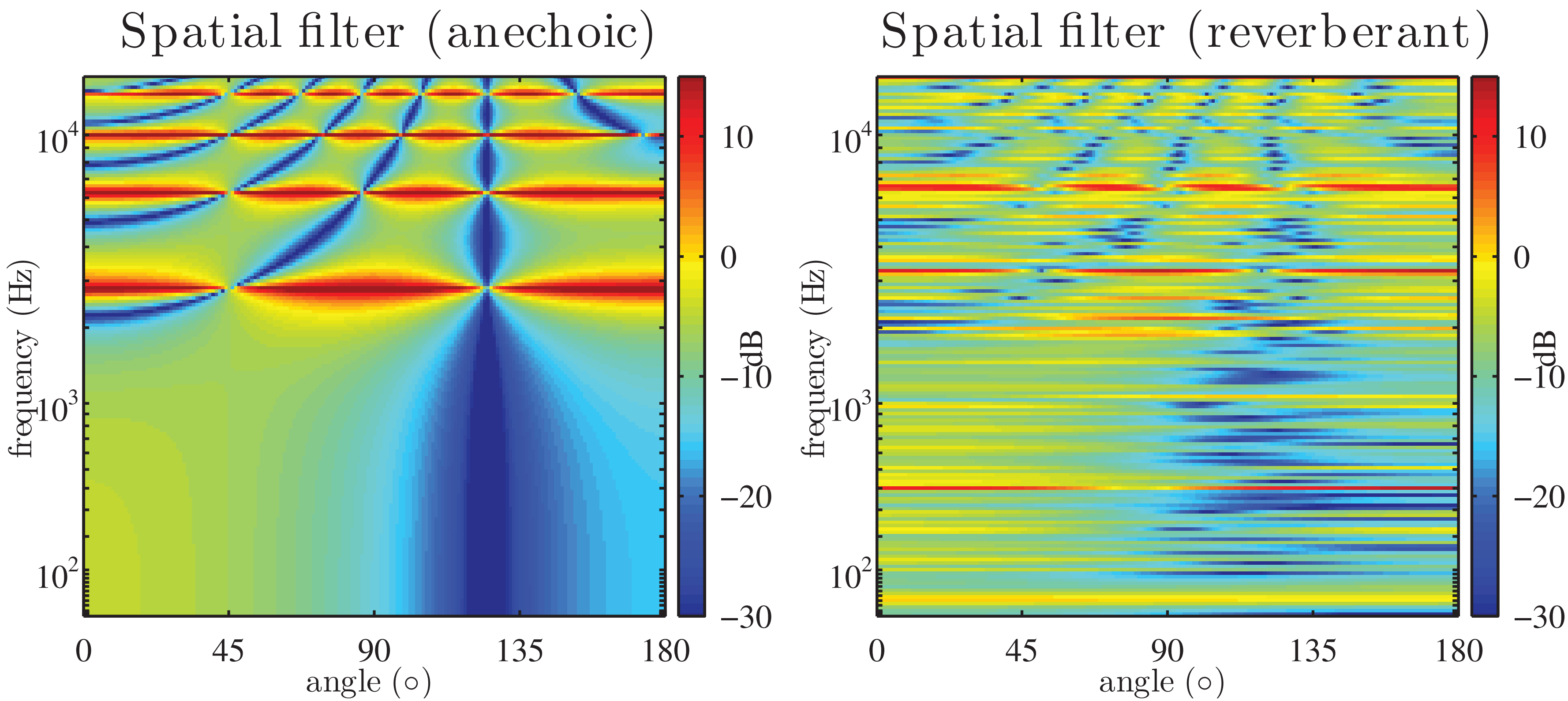
Spectral filtering

Single-channel: spectral filtering achieved via time-frequency masking.



Spatial filtering

Multichannel: combination of spatial and spectral filtering.



What is a good filter?

Spectral and spatial filtering can

- reduce other speech sounds and noises...
- but affect the target speech signal too!

Tradeoff between

- residual noise aka. interference
- speech distortion aka. artifacts

High speech distortion typically results in

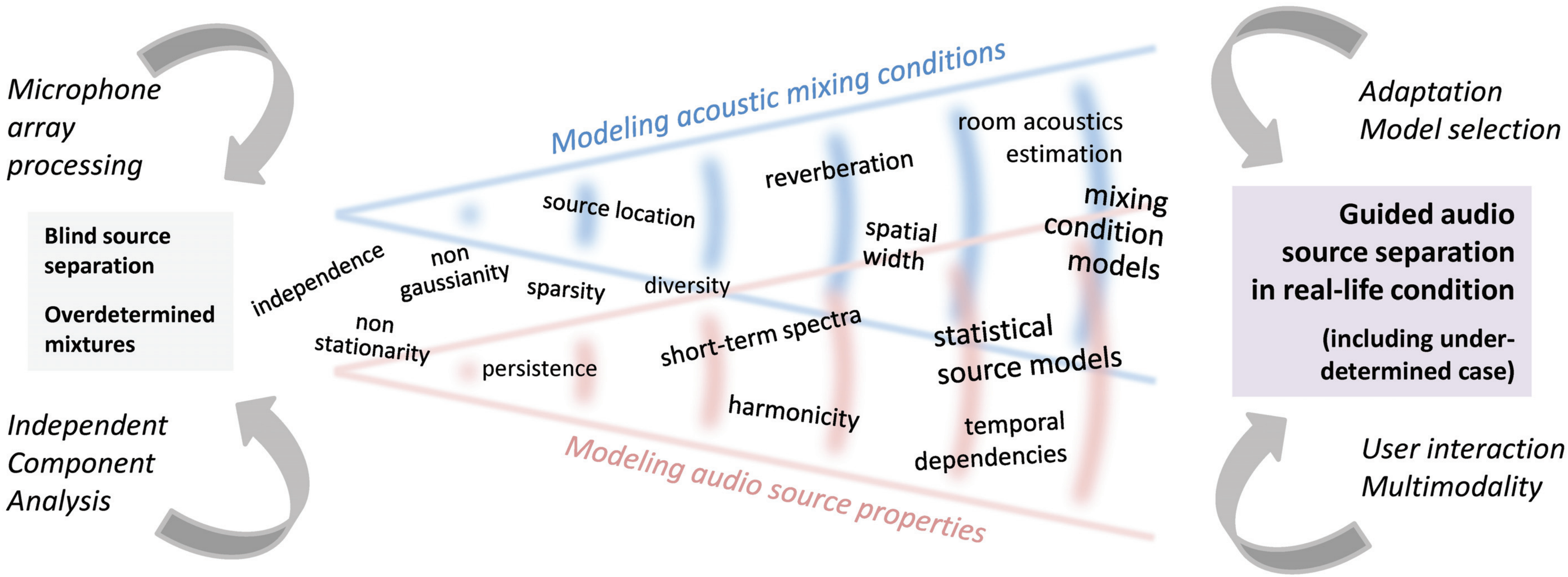
- low intelligibility,
- high error rate for speech recognition, speaker recognition...

How to estimate a good filter?

Two general approaches:

- model-based:
  - design a suitable source model (based on expertise),
  - learn/estimate the model parameters,
  - derive a filter,
- single-step: directly estimate the separation filter.

Increasingly complex models



Test paradigms

Separation methods often categorized according to the amount of information about the test data:

- blind: no information (inapplicable to audio),
- weakly guided: general information about the context of use, for instance "the sources are speech",
- strongly guided: specific information about the processed signal: speaker position, speaker identity, presence of a specific noise...
- informed: highly precise information encoded and transmitted along with the audio (kind of audio coding).

Training paradigms

In this tutorial, categorization according to existence and nature of training data:

- learning-free: no training, all parameters fixed by an expert or estimated from test data,
- unsupervised source modeling: train a model for each source from unannotated isolated signals of that source type,
- supervised source modeling: train a model for each source from isolated signals of that source type annotated with, e.g., speech transcripts, noise environment labels...
- separation based training: train a single-step filter or jointly train models for all sources from mixture signals given the underlying true source signals.

Last three categories are learning-based.

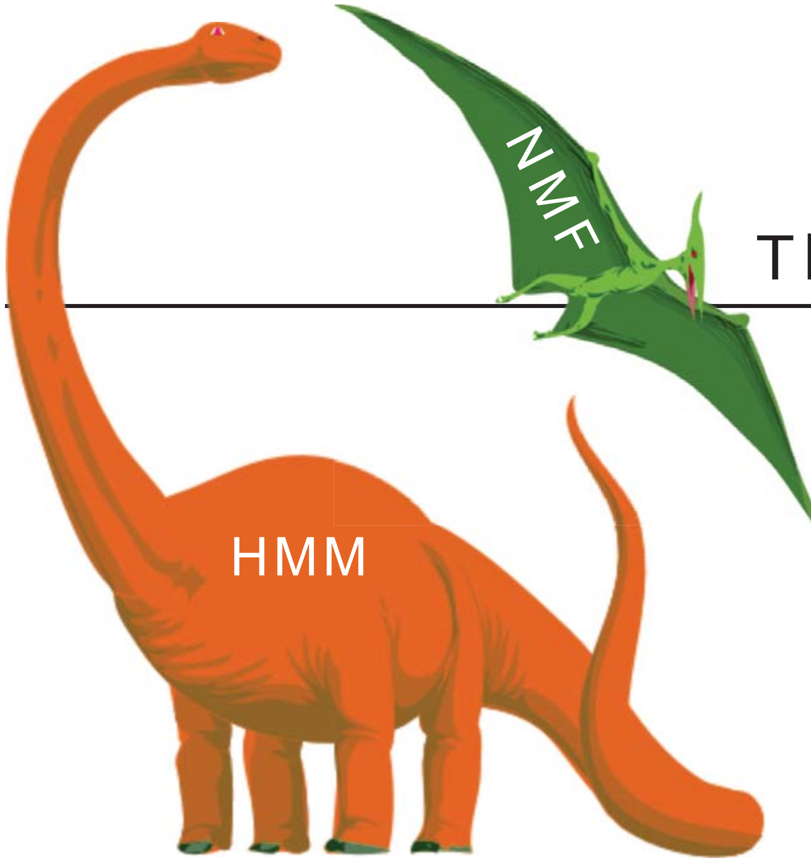
Why learning-based separation?

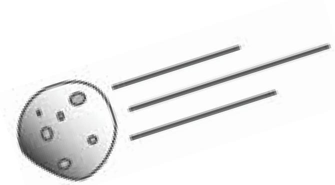
Compared to learning-free separation, learning-based separation can

- estimate parameter values more accurately (because data are not corrupted by interfering sources and noise),
- exploit larger amounts of data to design and learn more complex models.

Tutorial outline

- Graphical models, NMF, and shallow networks
- Deep learning approaches to single-channel separation
- Deep learning approaches to multichannel separation
- New directions in deep learning approaches
- Wrap-up, perspectives





The pre-deep-learning era

Speaker: Jonathan Le Roux

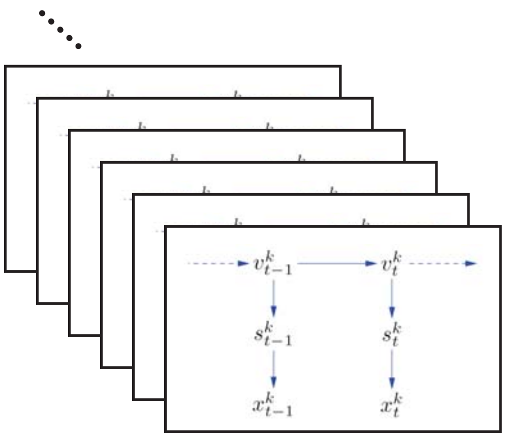
Interspeech 2016 Tutorial

Data-driven approaches to speech enhancement and separation

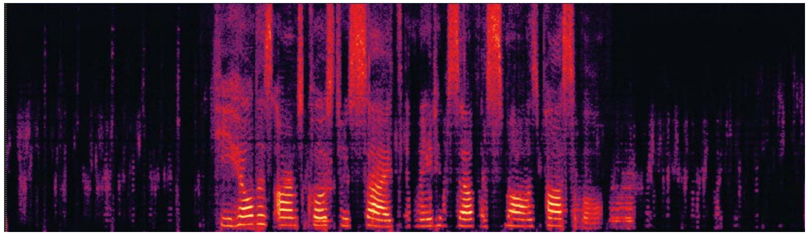
Generative vs discriminative

- Generative model-based methods
  - Training data used for
    - Getting models of each source (type), independently
  - Examples
    - Probabilistic models (GMMs/HMMs)
    - NMF (some overlap with probabilistic models)
- Discriminative methods: discriminative models, as well as discriminative training of generative models
  - Training data used for
    - Learning how to obtain the source estimates from the mixture
  - Examples
    - Some discriminative versions of the above
    - Classification-based objectives, early attempts using SVMs

Model-based Source Separation



Traffic Noise
Engine Noise
Speech Babble
Airport Noise
Car Noise
Music
Speech



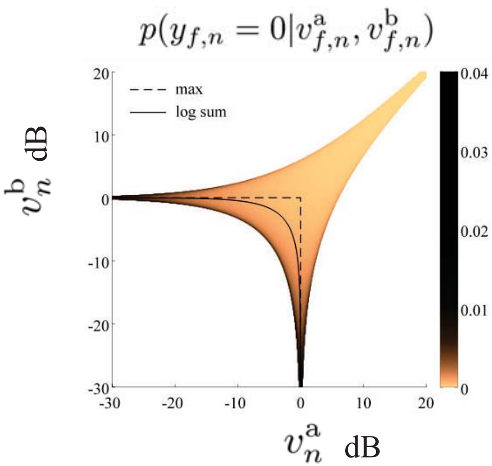
He held his arms close to…

Signal Models

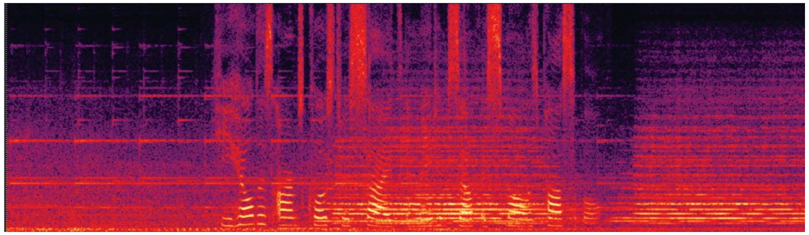
Inference

Predictions

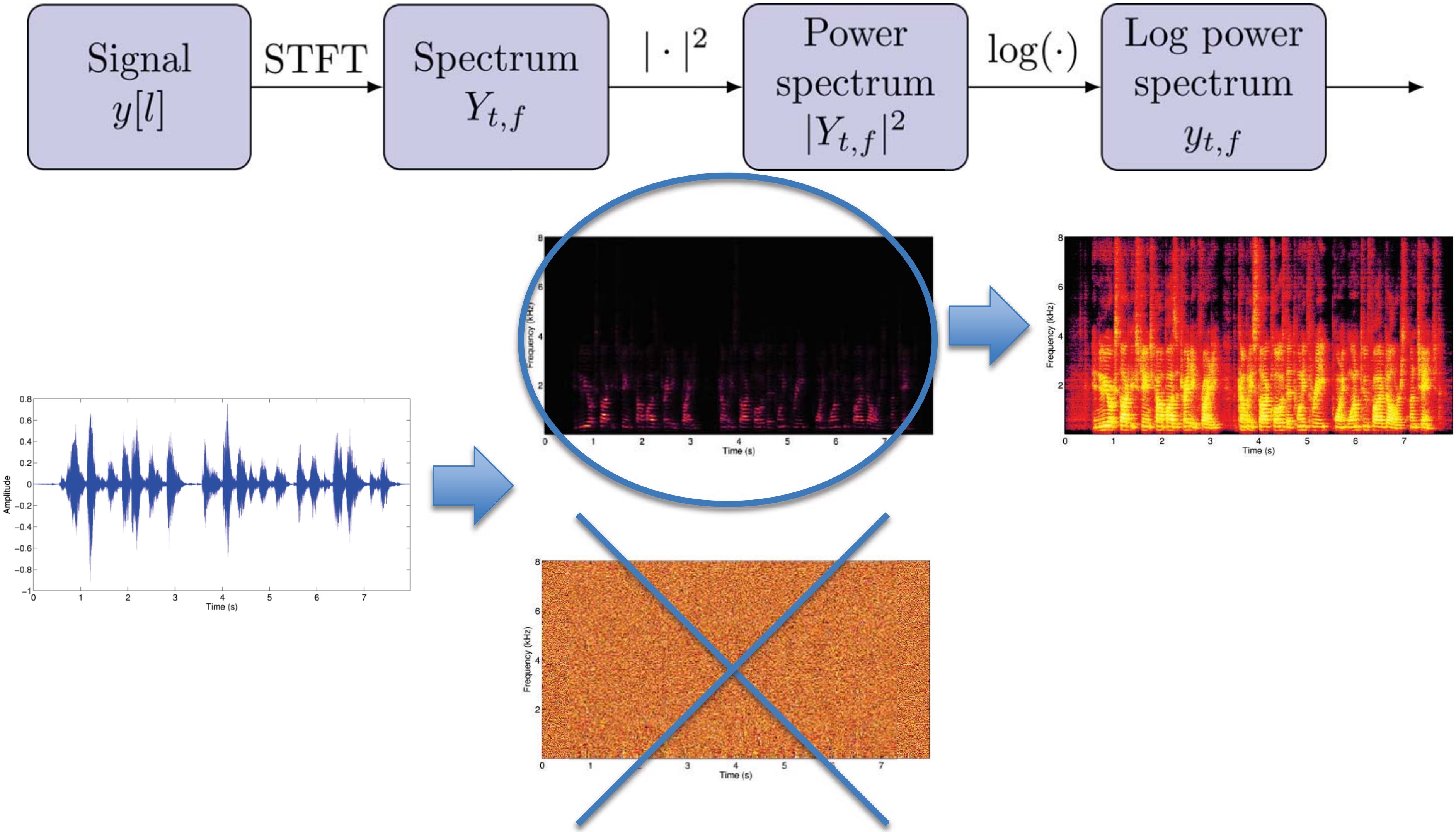
Interaction Models



Data



Signal modeling domain



Signal modeling domain

Signal $y[l]$ $\xrightarrow{\text{STFT}}$ Spectrum $Y_{t,f}$ $\xrightarrow{|\cdot|^2}$ Power spectrum $|Y_{t,f}|^2$ $\xrightarrow{\log(\cdot)}$ Log power spectrum $y_{t,f}$ $\rightarrow$

Challenging **Signal representation** Convenient

excitation/filter: convolutive — multiplicative — additive

dynamic range: bad — good

strong source overlap — sparsity ⇒ less source overlap

spectral patterns/harmonics identifiable

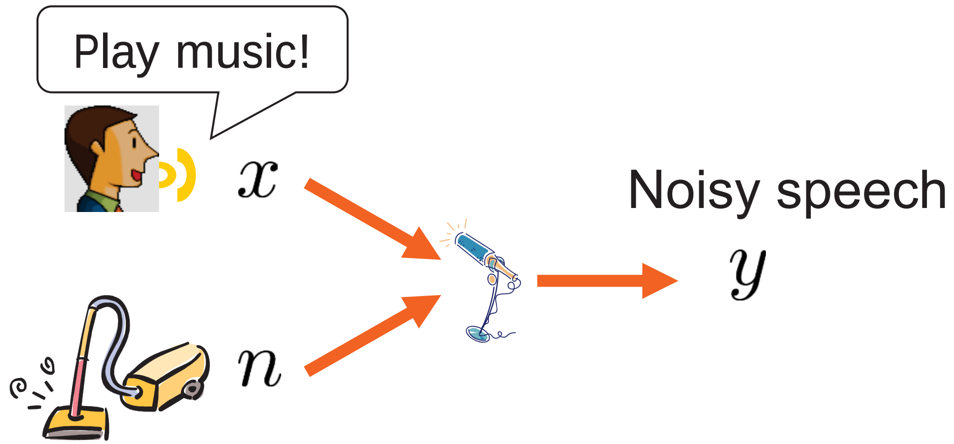
ignore phase (for better or worse)

Simple **Interaction model** Intricate

$$y[l] = x[l] + n[l] \qquad \cdots$$

$$y_{t,f} = \log\Big(\exp(x_{t,f}) + \exp(n_{t,f}) + 2\exp\big(\frac{x_{t,f}+n_{t,f}}{2}\big)\cos(\phi_{t,f})\Big)$$

Modeling signal interaction



Feature domain: / **Interaction model:**

Feature domain	Interaction model
Time	$y[l] = x[l] + n[l]$
Complex spectrum	$Y_{t,f} = X_{t,f} + N_{t,f}$
Power spectrum	$\|Y_{t,f}\|^2 = \|X_{t,f}\|^2 + \|N_{t,f}\|^2 + 2\|X_{t,f}\|\|N_{t,f}\| \cos(\phi_{t,f})$
Log-power spectrum	$y_{t,f} = \log\left(e^{x_{t,f}} + e^{n_{t,f}} + 2e^{\frac{x_{t,f}+n_{t,f}}{2}} \cos(\phi_{t,f})\right)$

Phase difference

Approximations:

Power-sum	$\|Y_{t,f}\|^2 \approx \|X_{t,f}\|^2 + \|N_{t,f}\|^2$
Log-sum	$y_{t,f} \approx \log\left(e^{x_{t,f}} + e^{n_{t,f}}\right)$
Max model	$y_{t,f} \approx \max\left(x_{t,f}, n_{t,f}\right)$

Example: enhancement using GMM with log-sum

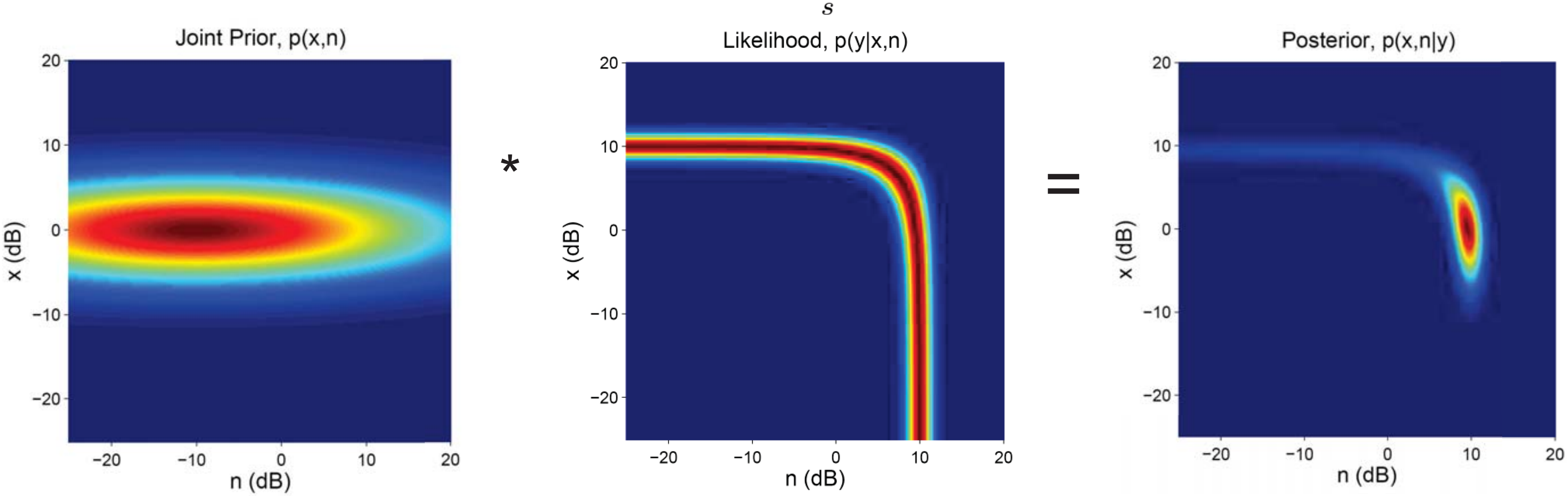
GMM clean speech model $p(\mathbf{x}|s^{\mathsf{x}}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathsf{x}|s^{\mathsf{x}}}, \boldsymbol{\Sigma}_{\mathsf{x}|s^{\mathsf{x}}}), \quad p(s^{\mathsf{x}})$

Single Gaussian noise model $p(\mathbf{n}) = \mathcal{N}(\mathbf{n}; \boldsymbol{\mu}_{\mathsf{n}}, \boldsymbol{\Sigma}_{\mathsf{n}})$

Log-sum approximation $p(y_{t,f}|x_{t,f}, n_{t,f}) = \mathcal{N}(y_{t,f}; \underbrace{\log(\mathrm{e}^{x_{t,f}} + \mathrm{e}^{n_{t,f}})}_{\text{non-linear}}, \psi_f)$

Still intractable!

- VTS: linearize at an expansion point $\tilde{\mathbf{z}}_s = [\tilde{\mathbf{x}}_s; \tilde{\mathbf{n}}_s]$ for speech state s^{x}
- Compute posterior distribution
- (Algonquin: iterate on $\tilde{\mathbf{z}}_s$ using posterior mean)
- Compute MMSE estimate: $\hat{\mathbf{x}} = \sum_s p(s|\mathbf{y}; (\tilde{\mathbf{z}}_{s'})_{s'}) \boldsymbol{\mu}_{\mathsf{x}|\mathbf{y},s;\tilde{\mathbf{z}}_s}$



Example: enhancement using GMM with log-sum

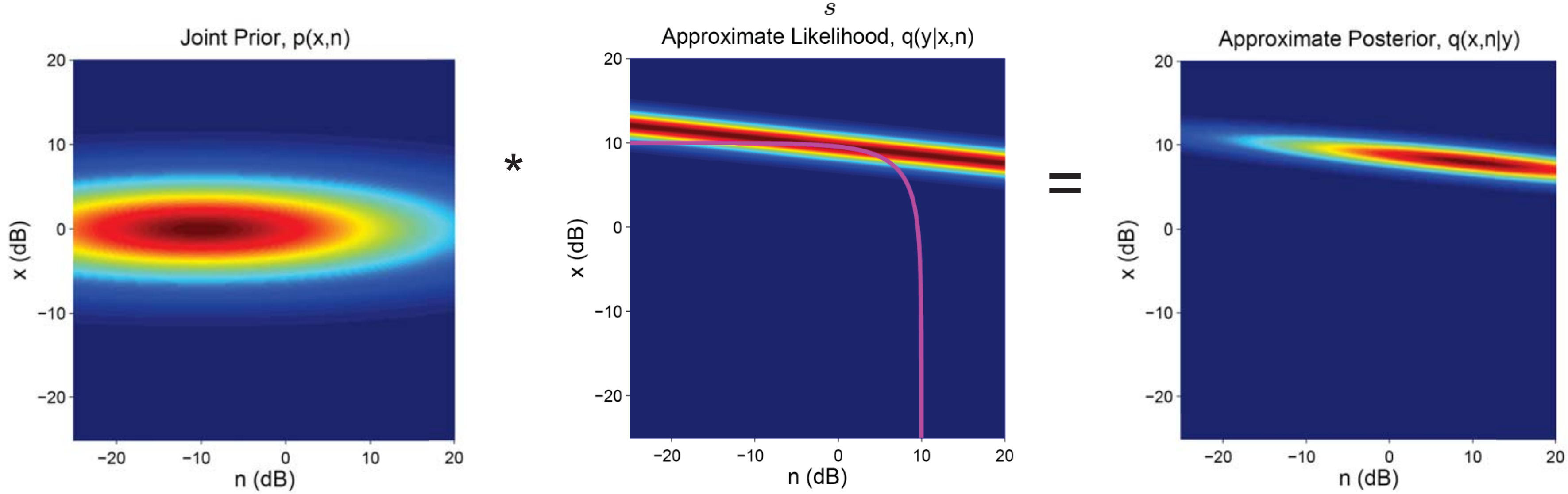
GMM clean speech model $p(\mathbf{x}|s^{\mathsf{x}}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathsf{x}|s^{\mathsf{x}}}, \boldsymbol{\Sigma}_{\mathsf{x}|s^{\mathsf{x}}}), \quad p(s^{\mathsf{x}})$

Single Gaussian noise model $p(\mathbf{n}) = \mathcal{N}(\mathbf{n}; \boldsymbol{\mu}_{\mathsf{n}}, \boldsymbol{\Sigma}_{\mathsf{n}})$

Log-sum approximation $p(y_{t,f}|x_{t,f}, n_{t,f}) = \mathcal{N}(y_{t,f}; \underbrace{\log(\mathrm{e}^{x_{t,f}} + \mathrm{e}^{n_{t,f}})}_{\text{non-linear}}, \psi_f)$

Still intractable!

- VTS: linearize at an expansion point $\tilde{\mathbf{z}}_s = [\tilde{\mathbf{x}}_s; \tilde{\mathbf{n}}_s]$ for speech state s^{x}
- Compute posterior distribution
- (Algonquin: iterate on $\tilde{\mathbf{z}}_s$ using posterior mean)
- Compute MMSE estimate: $\hat{\mathbf{x}} = \sum_s p(s|\mathbf{y}; (\tilde{\mathbf{z}}_{s'})_{s'}) \boldsymbol{\mu}_{\mathsf{x}|\mathbf{y},s;\tilde{\mathbf{z}}_s}$



Example: enhancement using GMM with log-sum

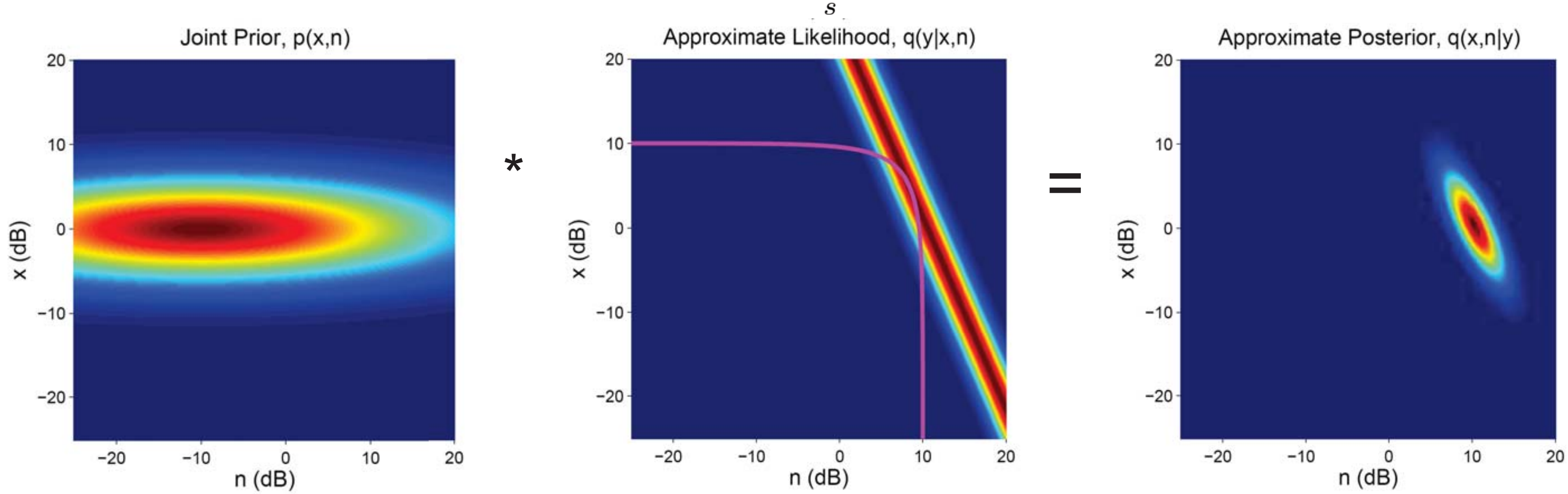
GMM clean speech model $p(\mathbf{x}|s^{\mathsf{x}}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathsf{x}|s^{\mathsf{x}}}, \boldsymbol{\Sigma}_{\mathsf{x}|s^{\mathsf{x}}}), \quad p(s^{\mathsf{x}})$

Single Gaussian noise model $p(\mathbf{n}) = \mathcal{N}(\mathbf{n}; \boldsymbol{\mu}_{\mathsf{n}}, \boldsymbol{\Sigma}_{\mathsf{n}})$

Log-sum approximation $p(y_{t,f}|x_{t,f}, n_{t,f}) = \mathcal{N}(y_{t,f}; \underbrace{\log(\mathrm{e}^{x_{t,f}} + \mathrm{e}^{n_{t,f}})}_{\text{non-linear}}, \psi_f)$

Still intractable!

- VTS: linearize at an expansion point $\tilde{\mathbf{z}}_s = [\tilde{\mathbf{x}}_s; \tilde{\mathbf{n}}_s]$ for speech state s^{x}
- Compute posterior distribution
- (Algonquin: iterate on $\tilde{\mathbf{z}}_s$ using posterior mean)
- Compute MMSE estimate: $\hat{\mathbf{x}} = \sum_s p(s|\mathbf{y}; (\tilde{\mathbf{z}}_{s'})_{s'}) \boldsymbol{\mu}_{\mathsf{x}|\mathbf{y},s;\tilde{\mathbf{z}}_s}$



Example: enhancement using GMM with log-sum

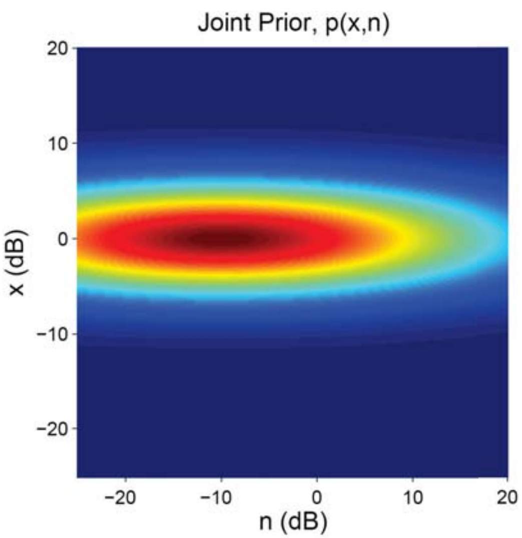
GMM clean speech model $p(\mathbf{x}|s^{\mathsf{x}}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathsf{x}|s^{\mathsf{x}}}, \boldsymbol{\Sigma}_{\mathsf{x}|s^{\mathsf{x}}}), \quad p(s^{\mathsf{x}})$

Single Gaussian noise model $p(\mathbf{n}) = \mathcal{N}(\mathbf{n}; \boldsymbol{\mu}_{\mathsf{n}}, \boldsymbol{\Sigma}_{\mathsf{n}})$

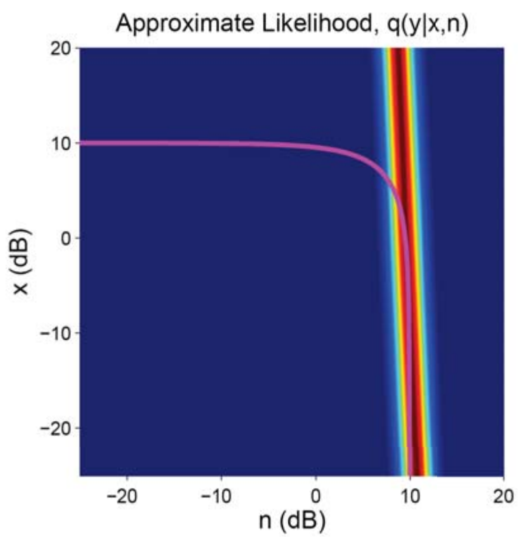
Log-sum approximation $p(y_{t,f}|x_{t,f}, n_{t,f}) = \mathcal{N}(y_{t,f}; \underbrace{\log(\mathrm{e}^{x_{t,f}} + \mathrm{e}^{n_{t,f}})}_{\text{non-linear}}, \psi_f)$

Still intractable!

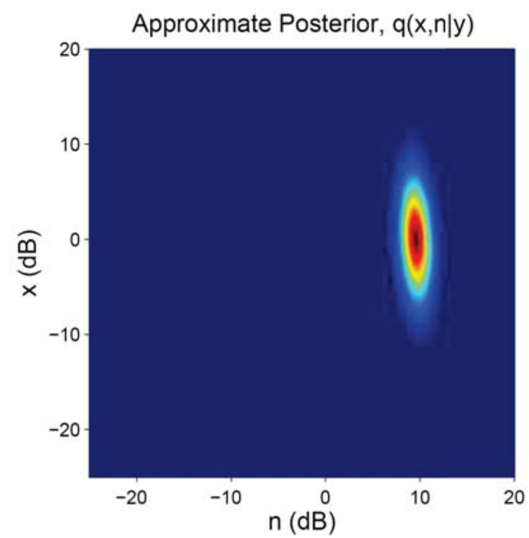
- VTS: linearize at an expansion point $\tilde{\mathbf{z}}_s = [\tilde{\mathbf{x}}_s; \tilde{\mathbf{n}}_s]$ for speech state s^{x}
- Compute posterior distribution
- (Algonquin: iterate on $\tilde{\mathbf{z}}_s$ using posterior mean)
- Compute MMSE estimate: $\hat{\mathbf{x}} = \sum_s p(s|\mathbf{y}; (\tilde{\mathbf{z}}_{s'})_{s'}) \boldsymbol{\mu}_{\mathsf{x}|\mathbf{y},s;\tilde{\mathbf{z}}_s}$



*



=



A tale of approximations

- Limit to class of models that we can begin to do maths on
  - Exponential families, conjugate priors, etc.
- Limiting assumptions
  - diagonal covariance in GMMs, conditional independence in HMMs
- Further approximate to derive useful quantities
  - VTS, max model
- … but still crazy derivations (no salvation from automatic diff.)
- Scales badly → Approximate inference algorithms, computational tricks
  - "Search" (not "inference"!)
    - Viterbi beam search across time
    - Hierarchical search for best Gaussian (cf. decision tree in ASR)
  - Band quantization to limit number of Gaussians to compute
- Nonetheless powerful, amenable to interpretation & extension

Factorial HMMs with max model

- K source models, with discrete states $s_t^k \in [1:N]$
- mask states $d_{f,t} \in [1:K]$ indicate dominant source
- inferring them jointly → **exponentially intractable!**

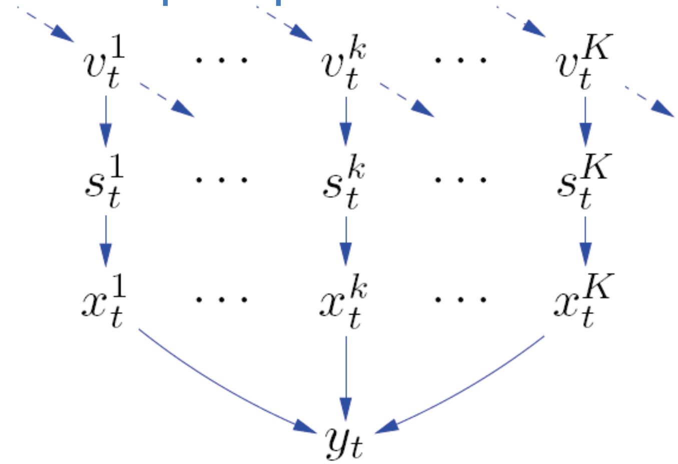
Posterior is a bi-partite graph:

- Given source states → infer masks in linear time
- Given mask states → infer sources in linear time

So variational EM can alternate between masks and source states [Rennie et al., 2008]

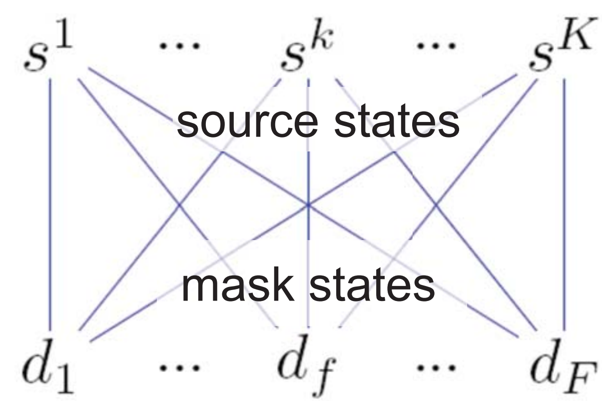
Gives amazing super-human results on a constrained problem (closed speaker set).

multiple speaker model

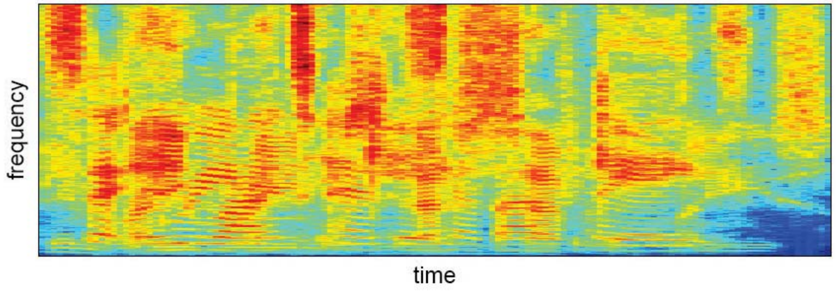


mixture $y_{t,f} = \max_k x_{t,f}^k$ sources

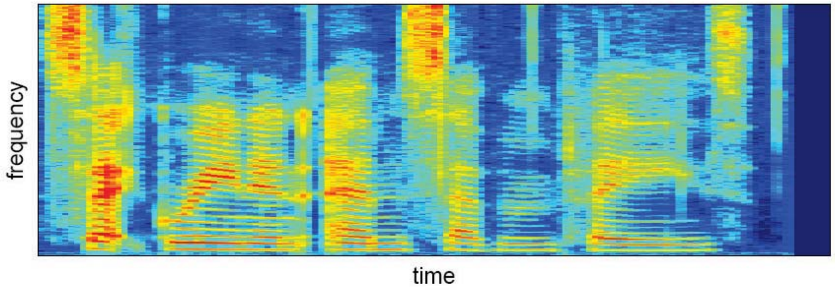
posterior state model



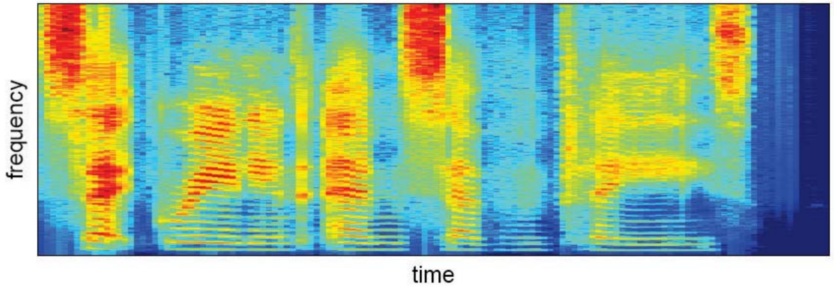
4 Speaker Separation



4 speaker mixture

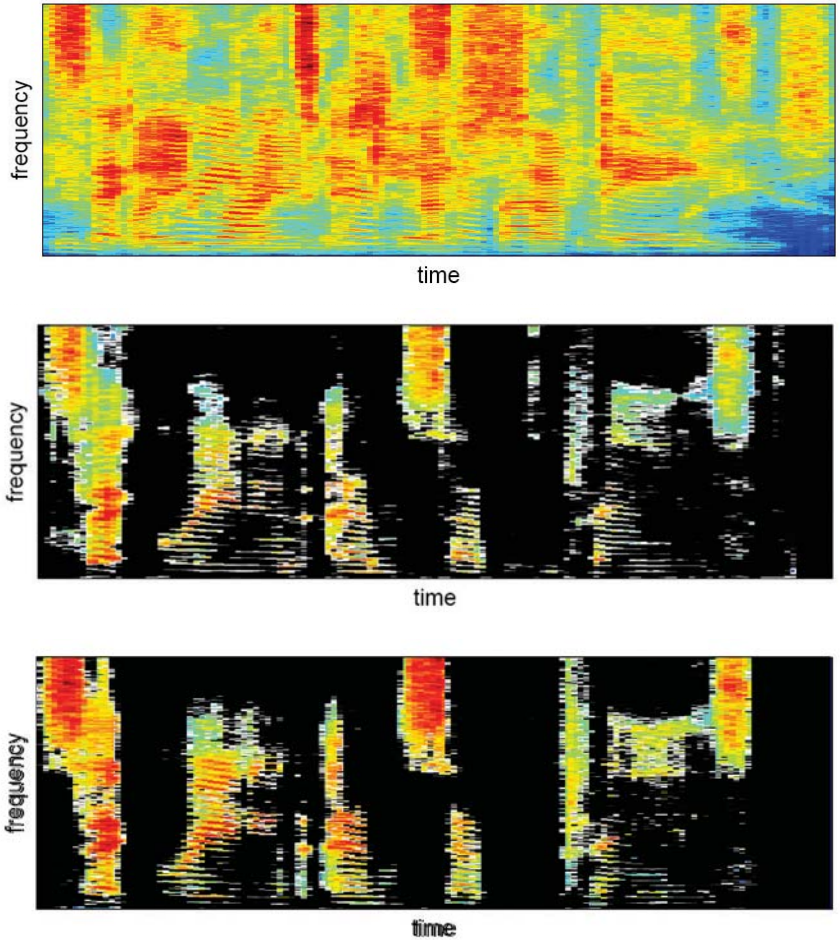


speaker #4 original



speaker #4 estimated

4 Speaker Separation



4 speaker mixture

speaker #4 mask

speaker #4 estimated mask

IBM’s 4-Speaker Separation Demo

Exact inference/marginals: over 1 trillion masks to compute

Approximate Inference: Only 1024 masks

PLACE WHITE AT D ZERO SOON 0 dB

PLACE RED IN H 3 NOW -7 dB

LAY BLUE AT P ZERO NOW -7 dB

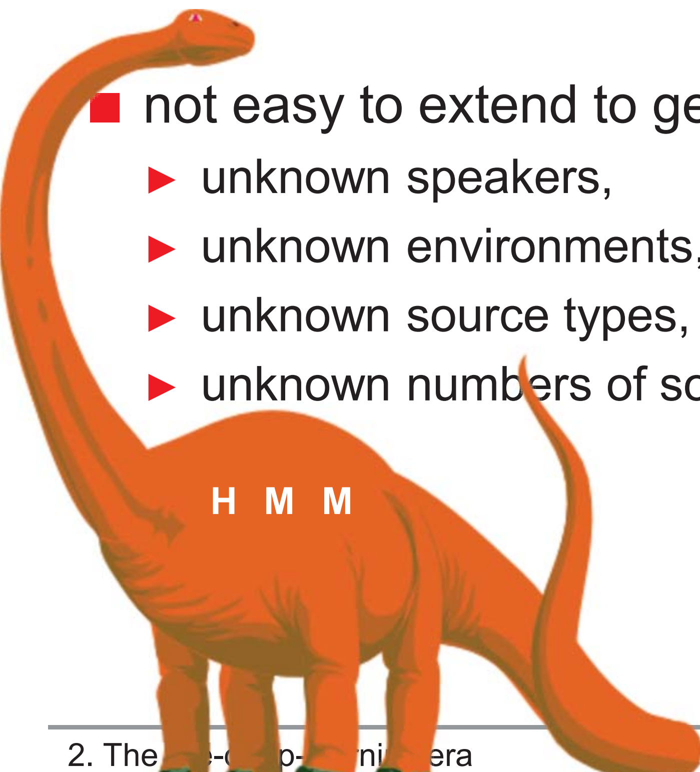
PLACE GREEN WITH B 8 SOON -7 dB

Impressive but ultimately limited

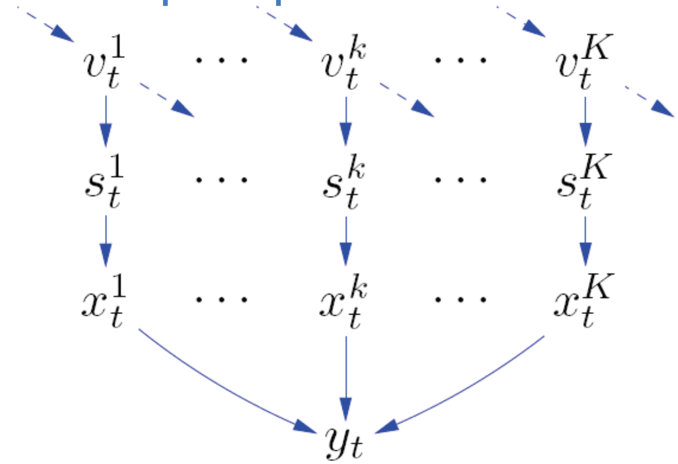
However, variational EM is:

- still too slow to be practical
- highly dependent on initialization
- has narrow model assumptions (e.g., diagonal covariance gaussians)

- not easy to extend to general conditions:
  - unknown speakers,
  - unknown environments,
  - unknown source types,
  - unknown numbers of sources.

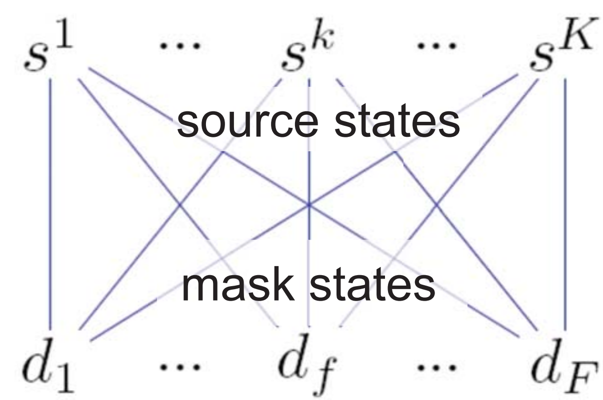


multiple speaker model



mixture $y_{t,f} = \max_k x_{t,f}^k$ sources

posterior state model



Non-negative Matrix Factorization (NMF)

- Factorize matrix $\mathbf{M} \geq 0$ into product $\mathbf{M} \approx \mathbf{WH}, \mathbf{W}, \mathbf{H} \geq 0$
- In audio, typically applied to (e.g., power) spectrogram

$\mathbf{M} \in \mathbb{R}^{F \times T}$ $\approx$ $\mathbf{W} \in \mathbb{R}^{F \times R}$ $*$ $\mathbf{H} \in \mathbb{R}^{R \times T}$

- Hope: decomposition into meaningful building blocks that are representative of the source or source type
  - Phonemes in speech, notes in music, etc.
  - Constraints such as sparsity added to help
- Obtained by minimizing some cost function

$$\overline{\mathbf{W}}, \overline{\mathbf{H}} = \arg\min_{\widetilde{\mathbf{W}}, \mathbf{H}} D(\mathbf{M} \mid \widetilde{\mathbf{W}}\mathbf{H}) + \mu |\mathbf{H}|_1$$

unit-norm

sparsity term

critical unless $R \ll F, T$

Relatively simple optimization procedure

- Typically minimizing some beta-divergence:

$$D_\beta(x|y) \stackrel{\text{def}}{=} \begin{cases} \frac{1}{\beta(\beta-1)}\left(x^\beta - y^\beta - \beta y^{\beta-1}(x-y)\right) & \text{if } \beta \in \mathbb{R}\backslash\{0,1\} \\ x(\log x - \log y) + (y - x) & \text{if } \beta = 1 \\ \frac{x}{y} - \log\frac{x}{y} - 1 & \text{if } \beta = 0 \end{cases}$$

$\beta = 2$: Euclidean dist.
Kullback-Leibler (KL) div.
Itakura-Saito dist.

- "Simple" iterative update equations

$$\mathbf{H} \leftarrow \mathbf{H} \otimes \frac{\mathbf{W}^\intercal(\mathbf{M} \otimes (\mathbf{WH})^{\beta-2})}{\mathbf{W}^\intercal(\mathbf{WH})^{\beta-1}}$$

$$\mathbf{W} \leftarrow \mathbf{W} \otimes \frac{((\mathbf{WH})^{\beta-2} \otimes \mathbf{M})\mathbf{H}^\intercal}{(\mathbf{WH})^{\beta-1}\mathbf{H}^\intercal}$$

- Easy to derive heuristically, e.g.,

$$\mathbf{H} \Leftarrow \mathbf{H} \circ \frac{[\nabla_{\mathbf{H}}\mathcal{E}]_-}{[\nabla_{\mathbf{H}}\mathcal{E}]_+},$$ where $$\nabla_{\mathbf{H}}\mathcal{E} = [\nabla_{\mathbf{W}}\mathcal{E}]_+ - [\nabla_{\mathbf{H}}\mathcal{E}]_-$$

Relatively simple optimization procedure

- Typically minimizing some beta-divergence:

$$D_\beta(x|y) \stackrel{\text{def}}{=} \begin{cases} \frac{1}{\beta(\beta-1)}\left(x^\beta - y^\beta - \beta y^{\beta-1}(x-y)\right) & \text{if } \beta \in \mathbb{R}\backslash\{0,1\} \\ x(\log x - \log y) + (y-x) & \text{if } \beta = 1 \\ \frac{x}{y} - \log\frac{x}{y} - 1 & \text{if } \beta = 0 \end{cases}$$

$\beta = 2$: Euclidean dist.
Kullback-Leibler (KL) div.
Itakura-Saito dist.

- "Simple" iterative update equations

$$\mathbf{H} \leftarrow \mathbf{H} \otimes \frac{\widetilde{\mathbf{W}}^\intercal(\mathbf{M} \otimes (\widetilde{\mathbf{W}\mathbf{H}})^{\beta-2})}{\widetilde{\mathbf{W}}^\intercal(\widetilde{\mathbf{W}\mathbf{H}})^{\beta-1} + \mu}$$

$$\mathbf{W} \leftarrow \widetilde{\mathbf{W}} \otimes \frac{((\widetilde{\mathbf{W}\mathbf{H}})^{\beta-2} \otimes \mathbf{M})\mathbf{H}^\intercal + \widetilde{\mathbf{W}} \otimes (\mathbf{1}\mathbf{1}^\intercal(\widetilde{\mathbf{W}} \otimes ((\widetilde{\mathbf{W}\mathbf{H}})^{\beta-1}\mathbf{H}^\intercal)))}{(\widetilde{\mathbf{W}\mathbf{H}})^{\beta-1}\mathbf{H}^\intercal + \widetilde{\mathbf{W}} \otimes (\mathbf{1}\mathbf{1}^\intercal(\widetilde{\mathbf{W}} \otimes (((\widetilde{\mathbf{W}\mathbf{H}})^{\beta-2} \otimes \mathbf{M})\mathbf{H}^\intercal)))}$$

- Easy to derive heuristically, e.g.,

$$\mathbf{H} \Leftarrow \mathbf{H} \circ \frac{[\nabla_\mathbf{H}\mathcal{E}]_-}{[\nabla_\mathbf{H}\mathcal{E}]_+}, \qquad \text{where} \qquad \nabla_\mathbf{H}\mathcal{E} = [\nabla_\mathbf{W}\mathcal{E}]_+ - [\nabla_\mathbf{H}\mathcal{E}]_-$$

NMF for speech separation

- One NMF model for each source type (e.g., speech v. noise)

$$\mathbf{M} \approx \sum_l \mathbf{S}^l \approx [\mathbf{W}^1 \cdots \mathbf{W}^L][\mathbf{H}^1; \cdots ; \mathbf{H}^L] = \mathbf{W}\mathbf{H}$$

  - So-called “supervised” setting: training data for all source types

Training: $\mathbf{T}^l \approx \overline{\mathbf{W}}^l \overline{\mathbf{H}}^l, \quad l = 1, \ldots, L$

bases activations

Obtain **bases** separately on each source type’s training data $\mathbf{T}^l$

Test: $\mathbf{M} \approx \sum_l \overline{\mathbf{W}}^l \hat{\mathbf{H}}^l = \overline{\mathbf{W}}\hat{\mathbf{H}}$

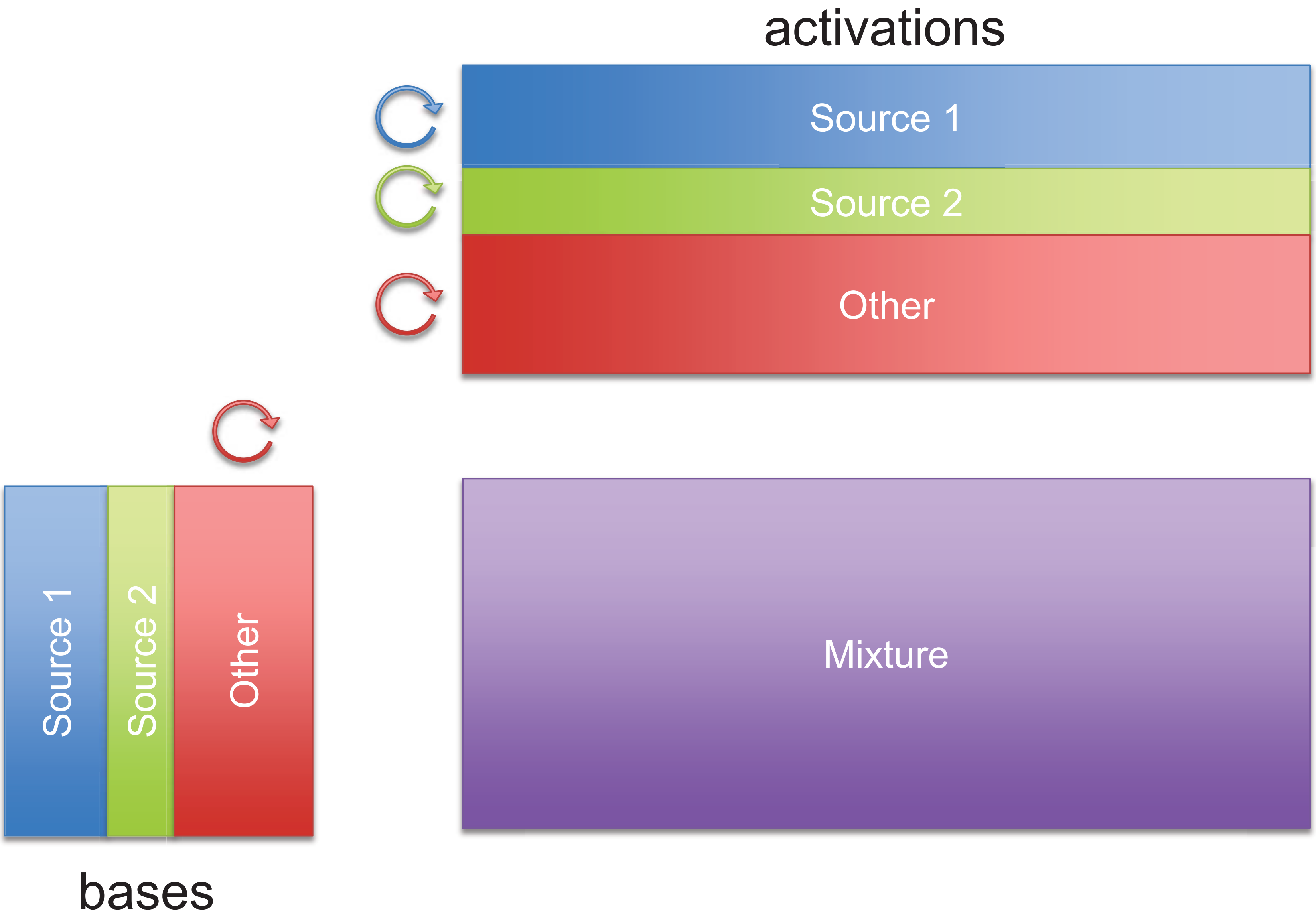
Obtain **activations** on mixture using trained bases

$$\hat{\mathbf{S}}^l = \frac{\overline{\mathbf{W}}^l \hat{\mathbf{H}}^l}{\sum_{l'} \overline{\mathbf{W}}^{l'} \hat{\mathbf{H}}^{l'}} \circ \mathbf{M}$$

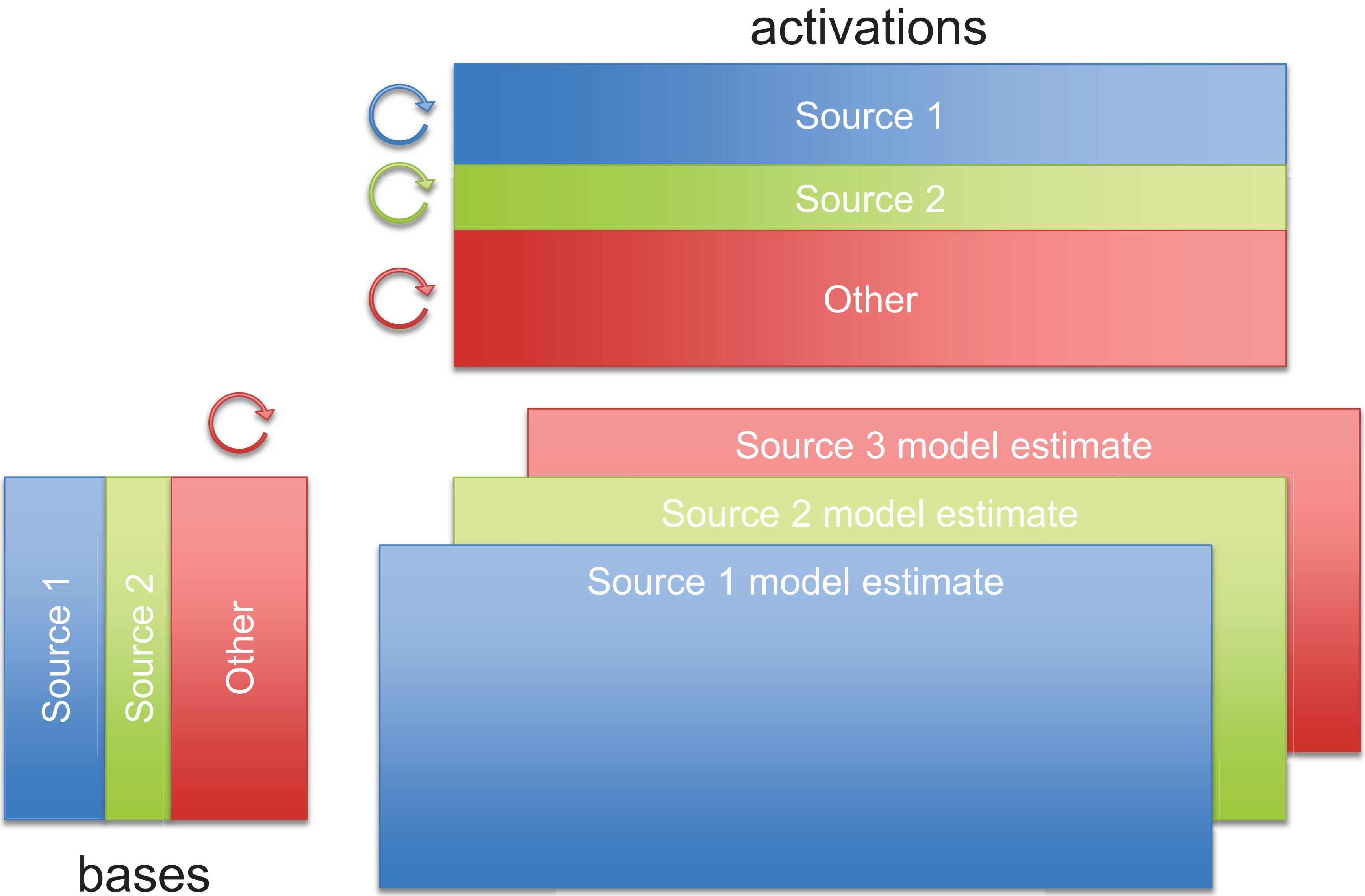
Reconstruct using Wiener filter-like mask

  - “Semi-supervised”: training data only for some source types, “garbage” model estimated at test time for the rest

Test time optimization



Test time optimization

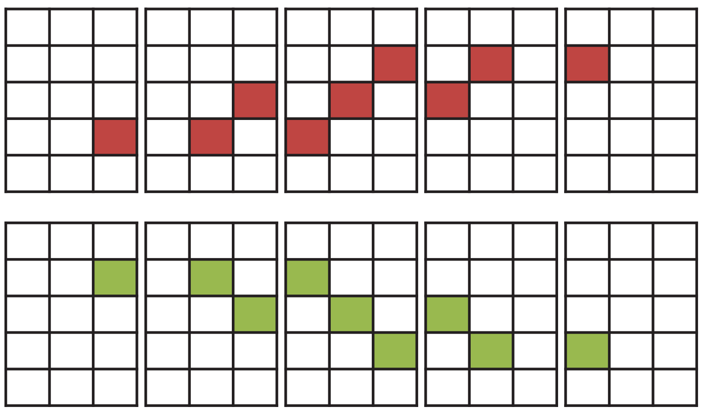


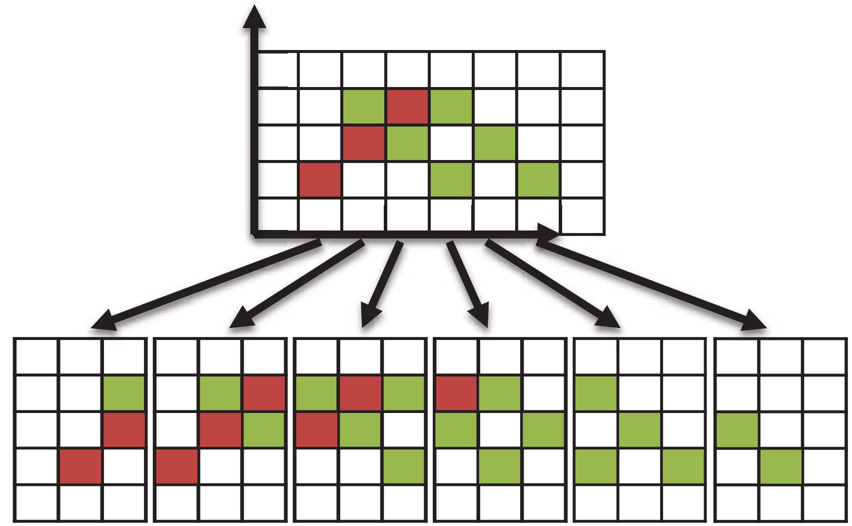
GMMs vs. NMF

- GMM arguably better suited to monophonic signals (e.g., speech)
  - Not great for music: too complex to represent all note combinations
  - Not great for mixtures: discrete state model, combinatorial optimization → exponential complexity in sources
- NMF great at handling polyphony
  - Popularity started with music
    - Especially good for instruments like piano whose spectrogram is approximately low rank
  - Not as clear for a single speaker because no polyphony, but may still be useful for mixtures: continuous state model → no explosion of complexity!
  - But components of one source and components of different sources interact in the same way → need additional constraints

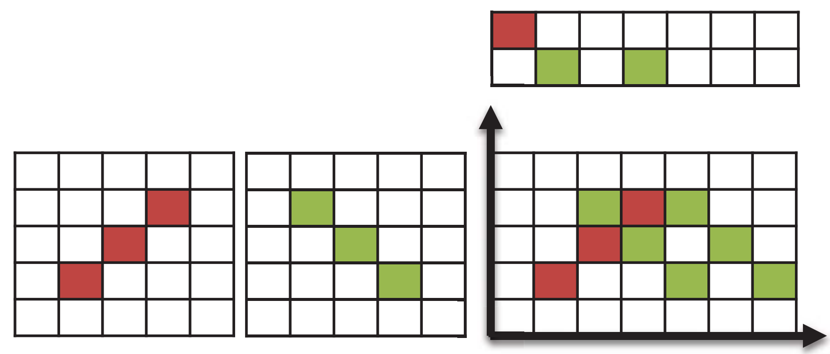
Handling dynamics and context

- Why do we need dynamics?
- Stacked/spliced frames





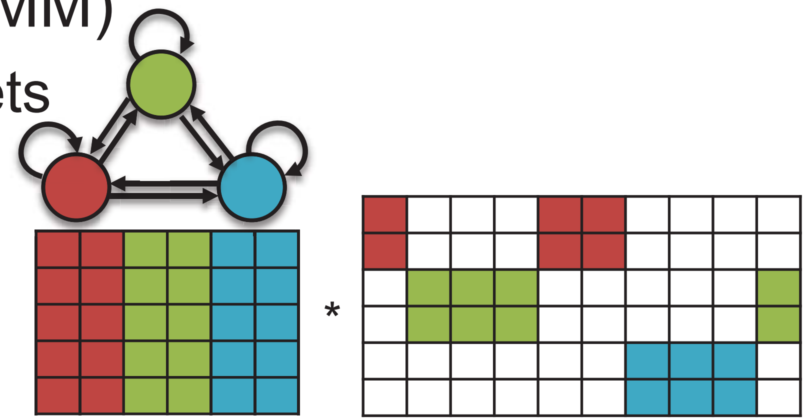
- Convolutive bases
  - OK for sounds with clear templates
  - Rationale less clear for speech



Handling dynamics and context

- Dynamic state models: $\mathbf{h}_t \sim p(\mathbf{h}_t | \mathbf{h}_{t-1})$
- Discrete: non-negative HMM (N-HMM)
  - transition between multiple basis sets
  - each set represents e.g. phoneme

$$\mathbf{h}_t \sim p(\mathbf{h}_t | s_t), \; s_t \sim p(s_t | s_{t-1})$$



- Continuous: non-negative dynamical system (NDS)

$$\mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1} \circ \boldsymbol{\epsilon}_t^{\mathrm{h}}, \quad \boldsymbol{\epsilon}_{r,t}^{\mathrm{h}} \sim \mathcal{G}(\alpha_r^{\mathrm{h}}, \theta_r^{\mathrm{h}})$$

$$\mathbf{v}_t = \mathbf{W}\mathbf{h}_t \circ \boldsymbol{\epsilon}_t^{\mathrm{v}}, \quad \boldsymbol{\epsilon}_{f,t}^{\mathrm{v}} \sim \mathcal{G}(\alpha_f^{\mathrm{v}}, \theta_f^{\mathrm{v}})$$

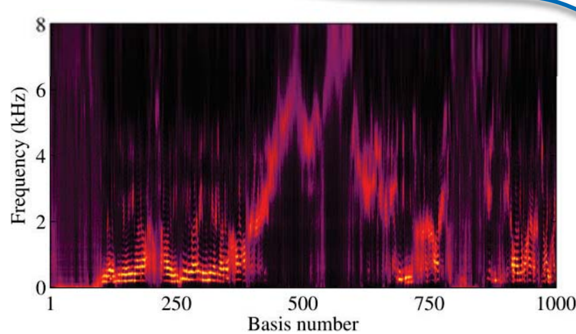
(Gamma)

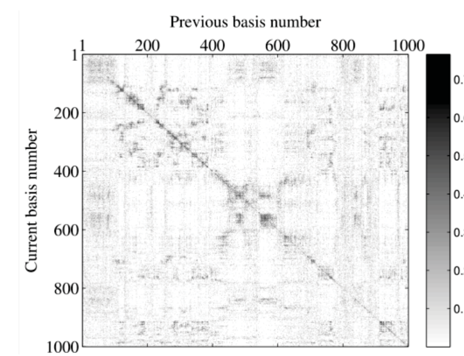
Activation

Power Spectrogram

Dictionary matrix

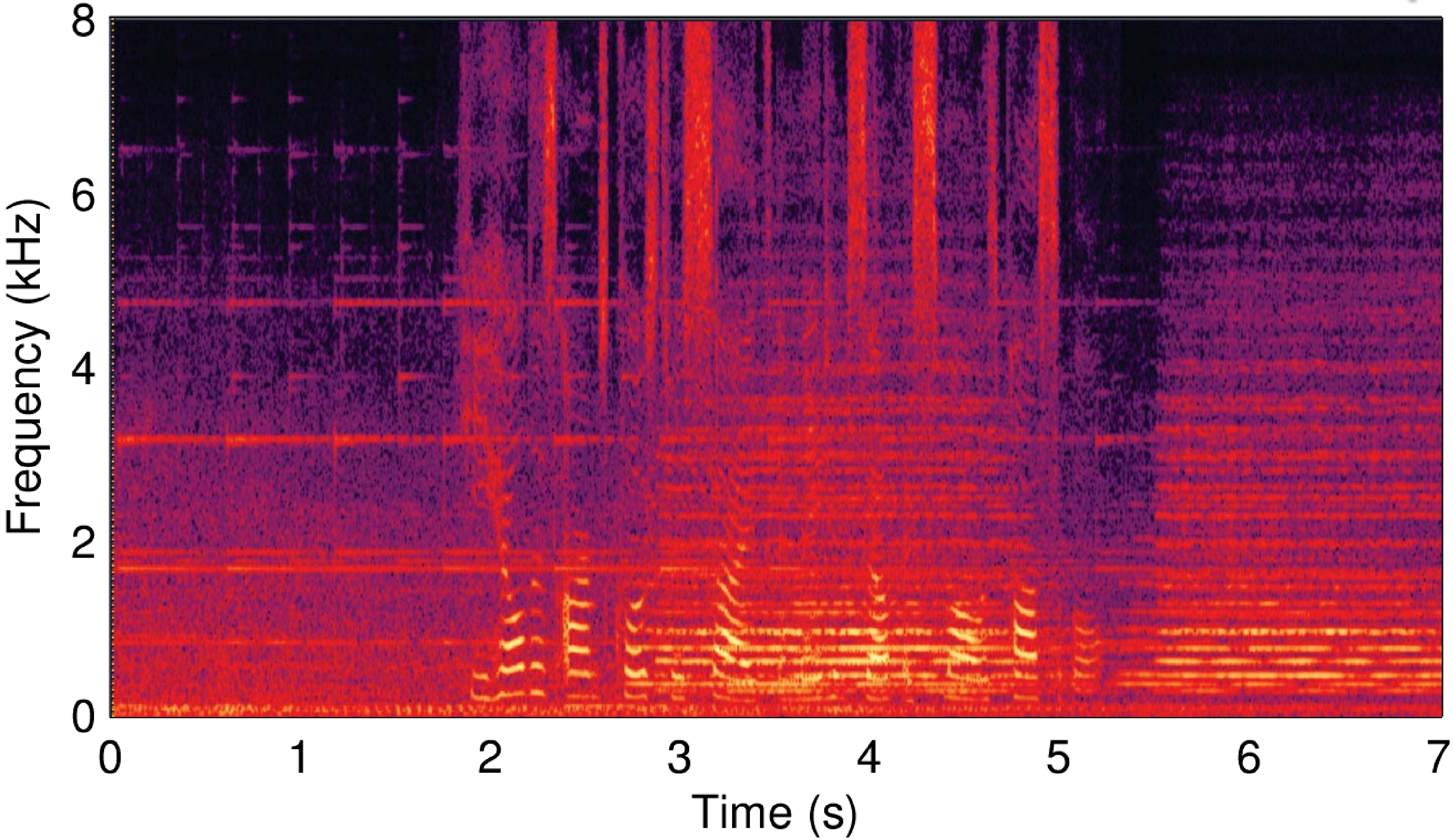
Transition matrix





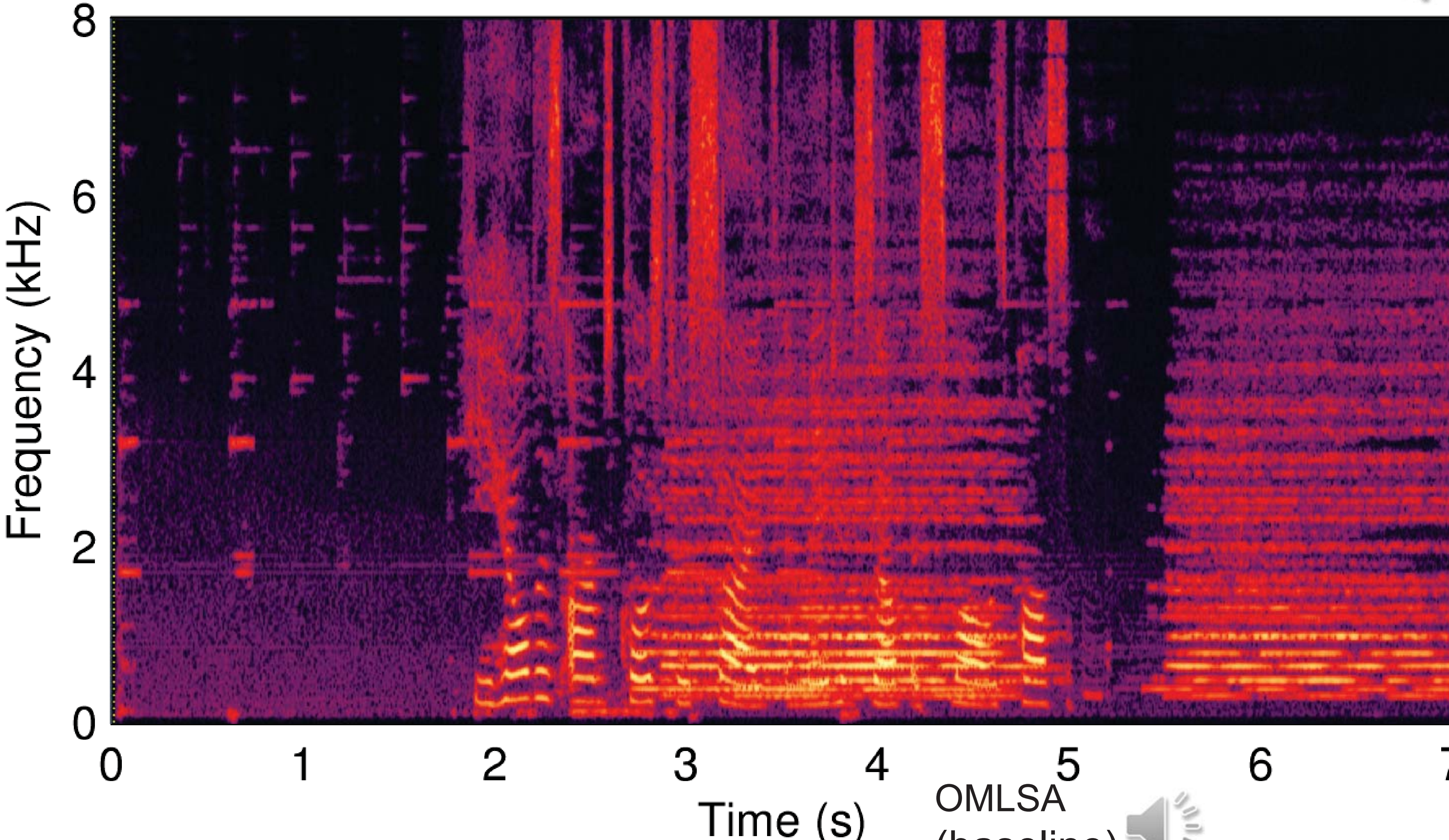
NDS enhancement example

No processing



NDS enhancement example

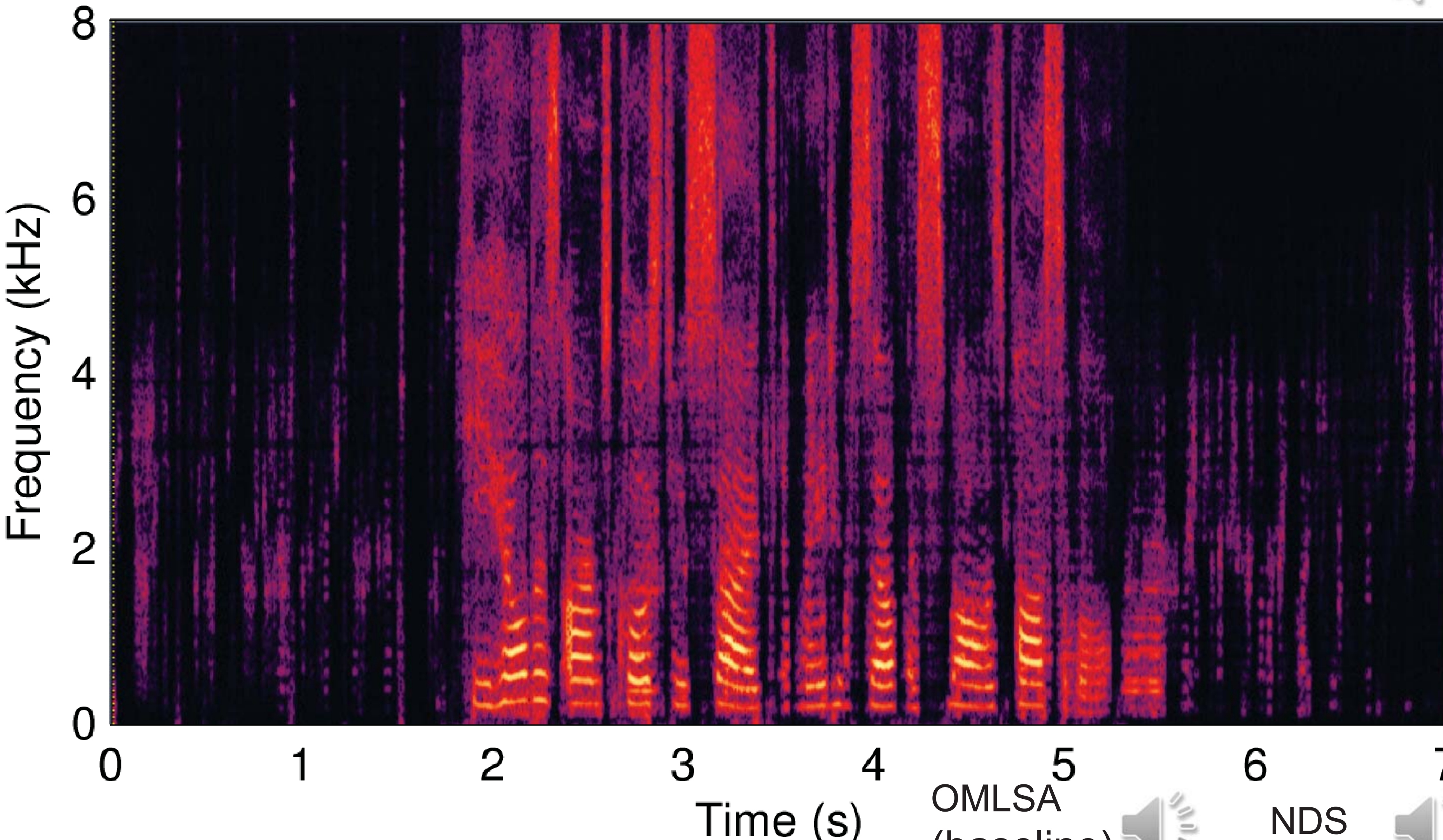
No processing



OMLSA
(baseline)

NDS enhancement example

No processing



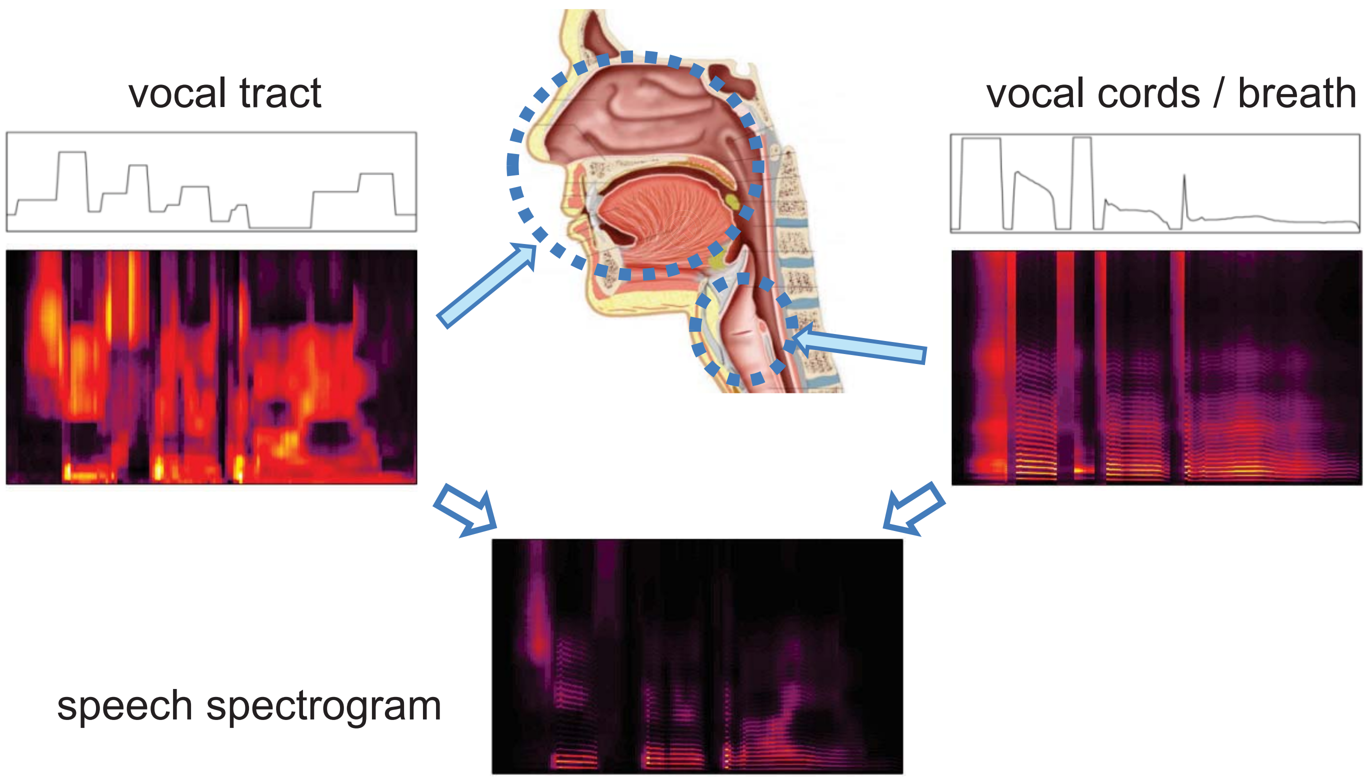
OMLSA (baseline)

NDS

Factorial extensions

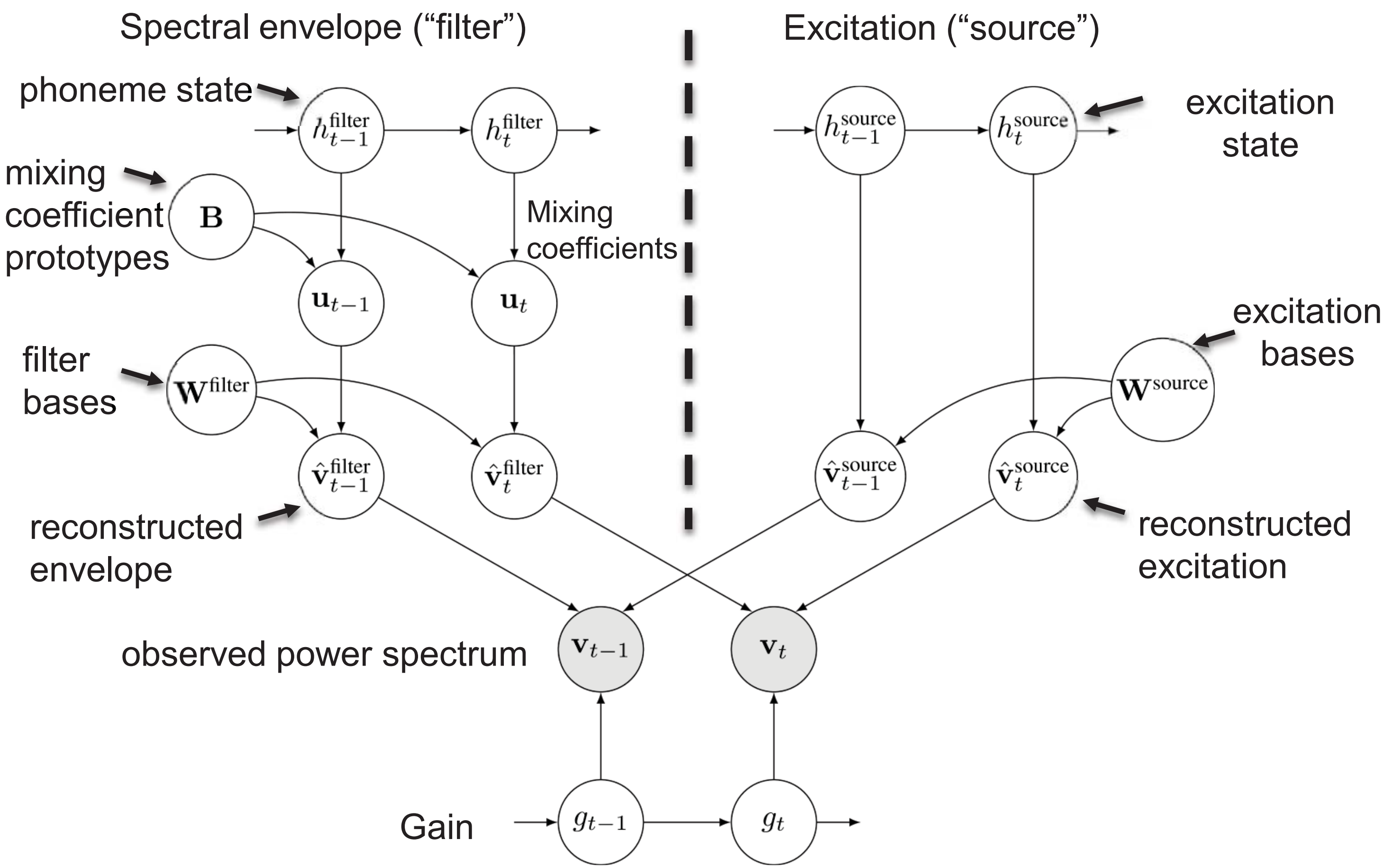
- Enforce further structure in spectrogram factorization
- Source-filter model of speech production
  - Can be as simple as $\mathbf{V}^{\text{speech}} \approx \mathbf{V}^{\text{filter}} \circ \mathbf{V}^{\text{source}}$

$$\approx (\mathbf{W}^{\text{filter}}\mathbf{H}^{\text{filter}}) \circ (\mathbf{W}^{\text{source}}\mathbf{H}^{\text{source}})$$



Factorial extensions

- Enforce further structure in spectrogram factorization
- Source-filter model of speech production
  - … or a bit more involved (SFNDS)



Many other extensions

- Use linguistic knowledge
  - Introduce language models via N-HMM framework
  - Iterate separation and ASR, using phoneme-dependent models
- Deal with unknown speakers
  - Universal speech model using group sparsity to represent a new speaker using a small number of known speaker models
- Online separation with unseen speaker/noise
  - Incrementally update noise model and speech+noise activations
- Handle phase information
  - Complex NMF: $V_{f,t} \approx \sum_r W_f^r H_t^r e^{j\phi^r_{f,t}}$
  - Time domain spectrogram factorization: optimize time-domain signal and NMF factorization to match its STFT
- Obtain better basis sets
  - Exemplar-based NMF: sample data frames as bases

What's "wrong" with "generative" methods?

- **The example of NMF-based separation:**

Training: $$\overline{\mathbf{W}}^l, \overline{\mathbf{H}}^l = \arg\min_{\mathbf{W}^l \mathbf{H}^l} D_{\beta_1}(\mathbf{T}^l \mid \mathbf{W}^l \mathbf{H}^l) + \mu |\mathbf{H}^l|_1$$

Bases optimal for activations obtained on sources!

Mismatch!

At test time, activations obtained on a mixture!

Test: $$\hat{\mathbf{H}} = \arg\min_{\mathbf{H}} D_{\beta_1}(\mathbf{M} \mid \overline{\mathbf{W}}\mathbf{H}) + \mu |\mathbf{H}|_1$$

$$\hat{\mathbf{S}}^l = \frac{\overline{\mathbf{W}}^l \hat{\mathbf{H}}^l}{\sum_{l'} \overline{\mathbf{W}}^{l'} \hat{\mathbf{H}}^{l'}} \circ \mathbf{M}$$

Wiener filter for reconstruction
Not part of objective!

Why is it so hard to train them discriminatively?

- Bi-level optimization

$$\arg\min_{\mathbf{W}} \mathcal{D}(\mathbf{S}^l | \hat{\mathbf{S}}^l_{\mathbf{W}}(\mathbf{M}))$$

where $$\hat{\mathbf{S}}^l_{\mathbf{W}}(\mathbf{M}) = \frac{\mathbf{W}^l \hat{\mathbf{H}}^l}{\sum_{l'} \mathbf{W}^{l'} \hat{\mathbf{H}}^{l'}} \circ \mathbf{M}$$

and $$\hat{\mathbf{H}} = \arg\min_{\mathbf{H}} D_{\beta_1}(\mathbf{M} \mid \mathbf{W}\mathbf{H}) + \mu |\mathbf{H}|_1$$

- A way to break the bi-level issue:
  - Allow the bases to be different
  - Added benefit: leads to a more general model

Discriminative NMF

Goal: maximize SDR for target source 1 (just say it!)

Training 1: $\overline{\mathbf{W}}^l, \overline{\mathbf{H}}^l = \underset{\mathbf{W}^l\mathbf{H}^l}{\arg\min}\, D_{\beta_1}(\mathbf{T}^l \mid \mathbf{W}^l\mathbf{H}^l) + \mu|\mathbf{H}^l|_1$ Analysis $\mathbf{W}$

Training 2: on training mixtures $m = s^1 + \cdots + s^L$,

$$\hat{\mathbf{H}} = \underset{\mathbf{H}}{\arg\min}\, D_{\beta_1}(\mathbf{M} \mid \overline{\mathbf{W}}\mathbf{H}) + \mu|\mathbf{H}|_1$$

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\arg\min}\, D_2\left(\mathbf{S}^1 \mid \hat{\mathbf{S}}^1_{\mathbf{W}}(\mathbf{M})\right)$$

Reconstruction $\mathbf{W}$

where $\hat{\mathbf{S}}^1_{\mathbf{W}}(\mathbf{M}) = \frac{\mathbf{W}^1\hat{\mathbf{H}}^1}{\sum_l \mathbf{W}^l\hat{\mathbf{H}}^l} \circ \mathbf{M}$

Same procedure

Test: $\hat{\mathbf{H}} = \underset{\mathbf{H}}{\arg\min}\, D_{\beta_1}(\mathbf{M} \mid \overline{\mathbf{W}}\mathbf{H}) + \mu|\mathbf{H}|_1$

$$\hat{\mathbf{S}}^1_{\hat{\mathbf{W}}}(\mathbf{M}) = \frac{\hat{\mathbf{W}}^1\hat{\mathbf{H}}^1}{\sum_l \hat{\mathbf{W}}^l\hat{\mathbf{H}}^l} \circ \mathbf{M}$$

First classification-based approaches

- Time-frequency (T-F) mask, binary or continuous, used as intermediate goal for decades (cf. Wiener filter, CASA, etc.)
- Generative model approach: use classifier to predict a mask
  - Bayesian classifier of SNR using GMMs on AM features
- Discriminative approaches
  - Optimize classification accuracy on mask
    - SVMs on various features
  - Optimize SNR
    - MLPs on pitch-based features (1 hidden layer)
- Shallow methods
  - Limited capacity, scale poorly, don't generalize well
  - Processing each channel separately
    - More context → more information, but harder to extract
  - Difficult to extend to joint inference of whole spectrograms

References 1 (Probabilistic models)

- A. P. Varga and R. K. Moore, “Hidden Markov model decomposition of speech and noise,“ in Proc. ICASSP, 1990
- P. J. Moreno, B. Raj, and R. M. Stern, “A vector Taylor series approach for environment-independent speech recognition," in Proc. ICASSP, 1996
- B. J. Frey, L. Deng, A. Acero, and T. T. Kristjansson, “ALGONQUIN: Iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition,” in Proc. Eurospeech, 2001
- J. R. Hershey, S. J. Rennie, and J. Le Roux, “Factorial models for noise robust speech recognition,” 2012
- T. Kristjansson and J. R. Hershey, “High resolution signal reconstruction,” in Proc. ASRU, 2003
- S. T. Roweis, “Factorial models and refiltering for speech separation and denoising,” in Proc. Interspeech, 2003
- T. T. Kristjansson, H. Attias, and J. R. Hershey, “Single microphone source separation using high resolution signal reconstruction,” in Proc. ICASSP, May 2004
- M. Fujimoto, K. Ishizuka, and T. Nakatani, “Study of integration of statistical model-based voice activity detection and noise suppression," in Proc. Interspeech, 2008
- J. Le Roux and J. R. Hershey, “Indirect model-based speech enhancement,” in Proc. ICASSP, Mar. 2012
- H. Attias, J. C. Platt, A. Acero, and L. Deng, “Speech denoising and dereverberation using probabilistic models,” in NIPS, 2001
- L. Benaroya, F. Bimbot, and R. Gribonval, “Audio source separation with a single sensor,” IEEE TASLP, 2006
- S. J. Rennie, P. Fousek, and P. L. Dognin, “Factorial hidden restricted Boltzmann machines for noise robust speech recognition,” in Proc. ICASSP, Mar. 2012

References 2 (Probabilistic models)

- Y. Ephraim, D. Malah, and B.-H. Juang, “On the application of hidden Markov models for enhancing noisy speech,” IEEE TASSP, 37 (12), 1989
- Y. Ephraim, “A Bayesian estimation approach for speech enhancement using hidden Markov models,” IEEE TSP, 40(4), 1992
- H. Sameti, H. Sheikhzadeh, L. Deng, and R. L. Brennan, “HMM-based strategies for enhancement of speech signals embedded in nonstationary noise,” IEEE TASP, 1998
- S. T. Roweis, “One microphone source separation," in NIPS, 2000
- J. R. Hershey and M. Casey, “Audio-visual sound separation via hidden Markov models,” in NIPS, 2001
- J. R. Hershey, T. Kristjansson, S. Rennie, and P. A. Olsen, “Single channel speech separation using factorial dynamics,” in NIPS, 2006
- S. J. Rennie, T. T. Kristjansson, P. A. Olsen, and R. Gopinath, “Dynamic noise adaptation,” in Proc. ICASSP, 2006

References 3 (NMF)

- D.D. Lee and H.S. Seung, “Learning the parts of objects by non-negative matrix factorization,” Nature, 1999.
- P. Smaragdis and J. C. Brown, “Non-negative matrix factorization for polyphonic music transcription,” in Proc. WASPAA, 2003
- P. Smaragdis, B. Raj, and M. Shashanka, “Supervised and semi-supervised separation of sounds from single-channel mixtures,” in Proc. ICA, 2007
- C. Joder, F. Weninger, F. Eyben, D. Virette, and B. Schuller, “Real-time speech separation by semi-supervised nonnegative matrix factorization,” in Proc. LVA/ICA, 2012
- N. Mohammadiha, P. Smaragdis, and A. Leijon, “Supervised and unsupervised speech enhancement using nonnegative matrix factorization,” IEEE TASLP, 21(5), 2013
- J. Le Roux, F. Weninger, J.R. Hershey, “Sparse NMF -- half-baked or well done?," MERL Tech. Rep. TR2015-023, 2015
- P. Sprechmann, A. M. Bronstein, and G. Sapiro, “Supervised non-negative matrix factorization for audio source separation,” in Excursions in Harmonic Analysis, Birkhaeuser, 2015

References 4 (NMF)

- P. Smaragdis, “Convolutive speech bases and their application to supervised speech separation,” IEEE TASLP, 15(1), 2007
- A. Ozerov, C. Fevotte, and M. Charbit, “Factorial scaled hidden Markov model for polyphonic audio representation and source separation,” in Proc. WASPAA, 2009
- G. Mysore, P. Smaragdis, and B. Raj, “Non-negative hidden Markov modeling of audio with application to source separation,” Proc. LVA, 2010
- M. Nakano, J. Le Roux, H. Kameoka, T. Nakamura, N. Ono and S. Sagayama, "Bayesian Nonparametric Spectrogram Modeling Based on Infinite Factorial Infinite Hidden Markov Model," in Proc. WASPAA, 2011
- G. Mysore and M. Sahani, “Variational inference in non-negative factorial hidden Markov models for efficient audio source separation,” in Proc. ICML, 2012
- C. Fevotte, J. Le Roux, and J. R. Hershey, “Non-negative dynamical system with application to speech and audio," in Proc. ICASSP, 2013
- P. Smaragdis, C. Fevotte, G. Mysore, N. Mohammadiha, M. Hoffman, “Static and Dynamic Source Separation Using Nonnegative Factorizations: A unified view,” IEEE SPM, 31(3), 2014

References 5 (NMF)

- A. Ozerov, E. Vincent, and F. Bimbot, “A general flexible framework for the handling of prior information in audio source separation,” IEEE TASLP, 20(4), 2012
- U. Simsekli, J. Le Roux, and J. R. Hershey, “Hierarchical and coupled non-negative dynamical systems with application to audio modeling,” in Proc. WASPAA, 2013
- J.-L. Durrieu, G. Richard, B. David, and C. Fevotte, “Source-filter model for unsupervised main melody extraction from polyphonic audio signals,” IEEE TASLP, 18, 2010
- U. Simsekli, J. Le Roux, and J. R. Hershey, “Non-negative source-filter dynamical system for speech enhancement,” in Proc. ICASSP, 2014
- D. Bouvier, N. Obin, M. Liuni, and A. Roebel, “A source/filter model with adaptive constraints for NMF-based speech separation,” in Proc. ICASSP, 2016

- B. Raj, R. Singh, and T. Virtanen, “Phoneme-dependent NMF for speech enhancement in monaural mixtures,” in Proc. Interspeech, 2011
- G. Mysore and P. Smaragdis, “A non-negative approach to language informed speech separation,” in Proc. LVA, 2012

References 6 (NMF)

- D. L. Sun and G. Mysore, "Universal speech models for speaker independent single channel source separation," in Proc. ICASSP, 2013
- C. Joder, F. Weninger, F. Eyben, D. Virette, and B. Schuller, "Real-time speech separation by semi-supervised nonnegative matrix factorization," in Proc. LVA, 2012
- Z. Duan, G. Mysore, and P. Smaragdis, "Speech enhancement by online non-negative spectrogram decomposition in non-stationary noise environments," in Proc. Interspeech, 2012
- F. G. Germain and G. Mysore, "Speaker and noise independent online single-channel speech enhancement," in Proc. ICASSP, 2015

- H. Kameoka, N. Ono, K. Kashino, and S. Sagayama, "Complex NMF: A new sparse representation for acoustic signals," in Proc. ICASSP, 2009
- H. Kameoka, "Multi-resolution signal decomposition with time-domain spectrogram factorization," in Proc. ICASSP, 2015

- J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," IEEE TASLP, 19(7), 2011
- P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Supervised non-Euclidean sparse NMF via bilevel optimization with applications to speech enhancement," in Proc. HSCMA, 2014
- F. Weninger, J. Le Roux, J. R. Hershey, and S. Watanabe, "Discriminative NMF and its application to single-channel source separation," in Proc. Interspeech, 2014

References 7 (Classification-based approaches)

- G. Hu, “Monaural speech organization and segregation,” Ph.D. dissertation, The Ohio State University, 2006
- Z. Jin and D.L. Wang, “A supervised learning approach to monaural segregation of reverberant speech,” Proc. ICASSP 2007
- G. Kim, Y. Lu, Y. Hu, and P. C. Loizou, “An algorithm that improves speech intelligibility in noise for normal-hearing listeners,” JASA, 2009
- K. Han and D. Wang, “An SVM based classification approach to speech separation,” in Proc. ICASSP, 2011
- K. Han and D. Wang, “On generalization of classification based speech separation,” in Proc. ICASSP, 2012
- K. Hu and D. Wang, “SVM-based separation of unvoiced-voiced speech in cochannel conditions,” in Proc. ICASSP, 2012

Deep learning approaches to single-channel speech enhancement/separation

Speaker: Hakan Erdogan

Interspeech 2016 Tutorial

Data-driven approaches to speech enhancement and separation

Deep learning

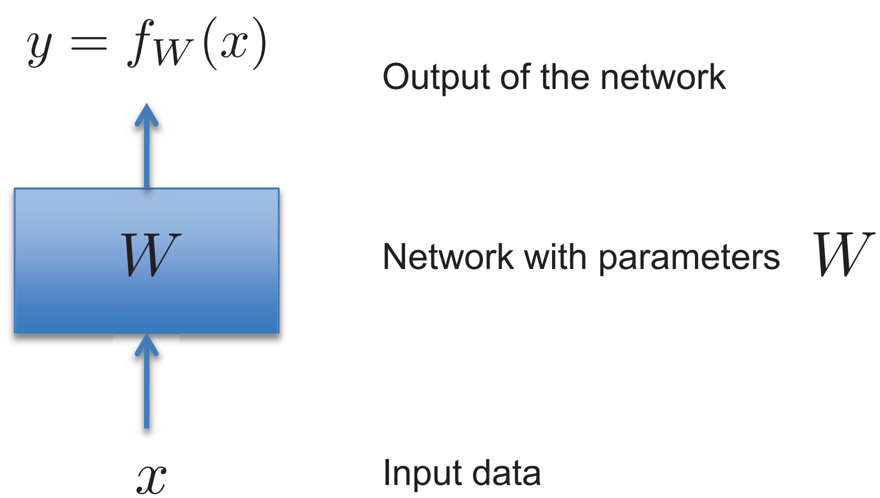
- What is deep learning?
- Hot research topic of interest while basic idea not so new
- Inspired from how brain processes data
- A computational machine with multi-layered architecture with an input x and output $y = f_W(x)$
- For a given input x, we would like to have a desired output t
  - Given many training pairs (x, t), We want to make $y = f_W(x)$ as close as possible to t for future x values
- The machine has parameters W that can be learned from data using automatic differentiation (back-propagation)
- What made deep learning explode recently?
  - More data
  - More computational power (GPUs) for learning parameters
  - Better theory, more manpower for research

Deep learning - continued

- Recent successes of deep learning (DL) in:
  - Speech recognition and spoken dialogue systems (siri, google voice) (2011-)
  - Object recognition in images (imagenet challenge) (2012-)
  - Handwriting recognition, machine translation, sentiment analysis
  - Atari game playing (2014) (DL+reinforcement learning) Video
  - Image captioning (2015) web page
  - Alphago (2016) news page
- Will it help get us to “true” Artificial Intelligence? What’s next?
- Companies
  - Google (bought Deepmind, Geoff Hinton’s company)
  - IBM, Microsoft, Apple, Amazon, Facebook
  - Openai (Elon Musk)
- Some important researchers
  - Geoff Hinton (U Toronto, Google), Yann LeCun (NYU, Facebook), Yoshua Bengio (U Montreal), Andrew Ng (Stanford, Baidu), Jurgen Schmidhuber (IDSIA)

Neural net – a function approximator

- Let $\mathcal{D} = \{(x_i, t_i) : i = 1, \ldots, N\}$ be a training data set of input and target pairs.
- We define a loss function $\mathcal{L}(W, \mathcal{D})$ to minimize with respect to W of network parameters.
- The goal of the minimization is to make the network outputs $f_W(x)$ get closer to targets t for future unseen inputs.



+ lots of data to train from

Multi-layer feed-forward network

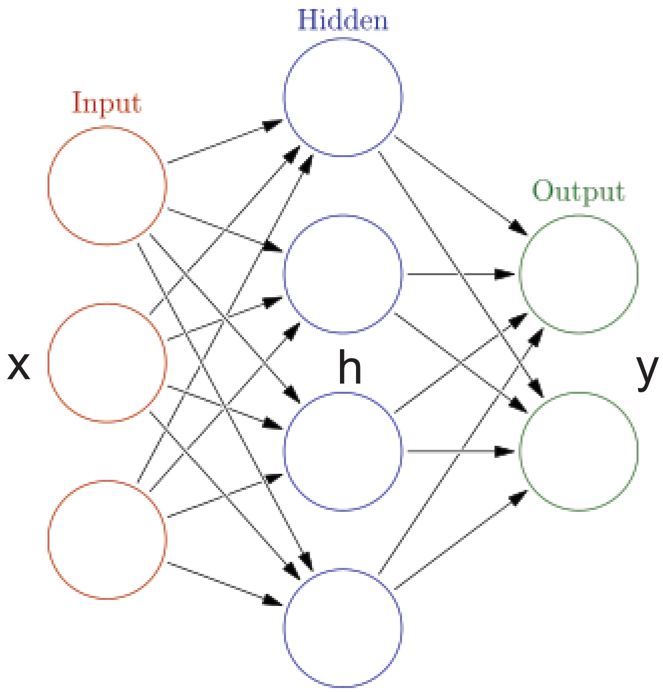


Image taken from wikipedia

Hidden layers 1 through L-1

$$\boldsymbol{h}^{(1)} = \sigma(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})$$
$$\boldsymbol{h}^{(k)} = \sigma(\boldsymbol{W}^{(k)}\boldsymbol{h}^{(k-1)} + \boldsymbol{b}^{(k)})$$
$$\boldsymbol{y} = \sigma(\boldsymbol{W}^{(L)}\boldsymbol{h}^{(L-1)} + \boldsymbol{b}^{(L)})$$

- First operation at a layer:
  - An affine transform: a matrix times input vector plus a bias vector
- Second operation at a layer:
  - An elementwise nonlinearity
    - Usually a sigmoid function
    - Tanh
    - Rectified linear unit
    - Others

Inputs, outputs, targets…

- Inputs (*x*) for a learning problem
  - Raw signal values (image pixel intensities, signal values)
  - Features extracted from data
  - Machine learning people prefer raw data -> no hand-engineering required
- Targets (*t*) are desired outputs known during training the system (learning targets)
  - Class identities, integral data (for classification)
  - Another vector (for regression), any N-dimensional tensor data
  - Sequences of classes/vectors/tensors
  - Other structured targets may also be possible

Loss functions

- Network training criterion:

$$\hat{W} = \arg\min_{W} \mathcal{L}(W, \mathcal{D})$$
$$\mathcal{L}(W, \mathcal{D}) = \sum_i D(f_W(x_i), t_i) + \lambda \mathcal{R}(W)$$

  where $D(.,.)$ is a distortion or divergence measure between a network output and a target, $\mathcal{R}(.)$ is a regularization function and λ is a parameter

- $\mathcal{L}$ is the loss function to minimize for training

- Example divergence measures

  - Cross-entropy loss for classification problems
  - Hinge loss for binary classification
  - Least-squares (or mean-squared error) loss for regression
  - Generalized KL divergence
  - Other application specific losses

Training a network

- How to solve the training problem?
  - Use (stochastic) gradient descent to minimize the loss function
  - Use **back-propagation** to calculate gradients
  - Stochastic gradient descent (SGD) works much faster than full batch gradient
    - Many variants exist: momentum, RMSPROP, RPROP, ADAM etc.
  - In SGD, we iteratively update parameters using mini-batches of training data
    - choose a mini-batch of data, calculate gradients on that data only and update parameters in the direction of the negative gradient with a step size
    - Step size in the update which is called the learning rate is an important parameter in SGD
  - Other optimization methods such as Hessian-free optimization exist and can sometimes achieve better results, but slower to run

Back-propagation - 1

- Consider a mini batch of data $\mathcal{B} = \{(x_i, t_i) : i = 1, \ldots, N_b\}$ and a loss function $\mathcal{L}(W, \mathcal{B}) = \sum_i D(f_W(x_i), t_i)$. Consider a scalar parameter w somewhere in the network.
- SGD update rule is given as

$$w := w - \eta \frac{\partial \mathcal{L}}{\partial w}.$$

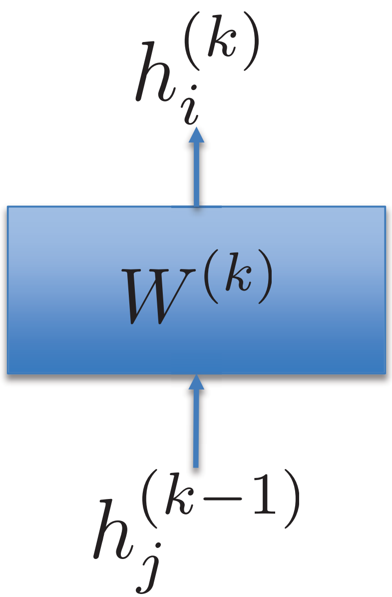
- Finding the gradient of the loss function with respect to w is an exercise in using the chain rule from calculus.
- For a feedforward network, lets say we know the gradient with respect to the outputs of a layer k

$$v_i^{(k)} = \frac{\partial \mathcal{L}}{\partial h_i^{(k)}}$$

Back-propagation - 2

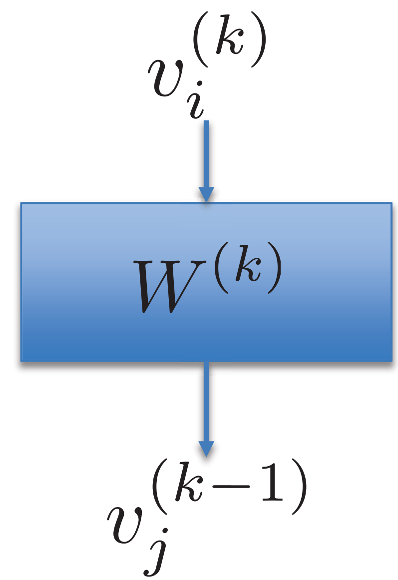
- Now our goal is to determine the gradient with respect to weights in layer k and also the gradient with respect to the previous layer outputs (ignoring biases for simplicity)

- Define linear layer outputs as $z_i^{(k)}$. We can calculate the gradient with respect to the weights as follows:



Forward pass

$$
\begin{aligned}
z_i^{(k)} &= \sum_j w_{ij}^{(k)} h_j^{(k-1)} \\
h_i^{(k)} &= \sigma(z_i^{(k)}) \\
\frac{\partial \mathcal{L}}{\partial w_{ij}^{(k)}} &= \frac{\partial \mathcal{L}}{\partial h_i^{(k)}} \frac{\partial h_i^{(k)}}{\partial z_i^{(k)}} \frac{\partial z_i^{(k)}}{\partial w_{ij}^{(k)}} \\
\frac{\partial \mathcal{L}}{\partial w_{ij}^{(k)}} &= v_i^{(k)} \sigma'(z_i^{(k)}) h_j^{(k-1)}
\end{aligned}
$$



Backward pass

Back-propagation - 3

- The derivatives with respect to the previous layer outputs can be calculated using the chain rule:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial h_j^{(k-1)}} &= \sum_i \frac{\partial \mathcal{L}}{\partial h_i^{(k)}} \frac{\partial h_i^{(k)}}{\partial z_i^{(k)}} \frac{\partial z_i^{(k)}}{\partial h_j^{(k-1)}} \\
\frac{\partial \mathcal{L}}{\partial h_j^{(k-1)}} &= \sum_i v_i^{(k)} \sigma'(z_i^{(k)}) w_{ij}.
\end{aligned}$$

- Note that, for the sigmoid function, we have $\sigma'(z_i^{(k)}) = h_i^{(k)}(1 - h_i^{(k)})$.
- So, finally, we get these two gradients:

$$\begin{aligned}
v_j^{(k-1)} &= \frac{\partial \mathcal{L}}{\partial h_j^{(k-1)}} = \sum_i v_i^{(k)} \sigma'(z_i^{(k)}) w_{ij} \\
\frac{\partial \mathcal{L}}{\partial w_{ij}^{(k)}} &= v_i^{(k)} \sigma'(z_i^{(k)}) h_j^{(k-1)}
\end{aligned}$$

Back-propagation - 4

In summary, the back-propagation for a feed forward network can be conducted as follows:

- First, calculate the gradient of the mini-batch loss function $\mathcal{L}(W, \mathcal{B})$ with respect to the outputs of the network $y = f_W(x)$ to obtain v_L
- Then, for each layer
  - First, element-wise multiply the incoming derivative v_i's with the derivative of the activation function at that layer to obtain a vector d
  - Second, form outer product of this vector with the inputs h to obtain the gradient dh^T for the weights of that layer
  - Third, multiply the vector d with the transpose of the weight matrix to obtain the gradient $W^T d$ with respect to the inputs of that layer
- Now, repeat the same for all the lower layers

Generalization: Computational networks

- Consider a directed acyclic graph (DAG) of computational “layers”
- Each layer needs to provide a forward pass and backward pass routine
- In the backward pass, one should calculate the gradient wrt the learnable parameters (weights) in the layer and also the gradient wrt the inputs of the layer given the gradient wrt the outputs of the layer
- If outputs of a layer is fed into multiple layers, then the gradients coming from each (in the backward pass) should be summed
- All gradients can be computed with a single forward pass followed by a backward pass through the whole computational network
- Many toolkits (cntk, theano, tensorflow, torch, caffe, chainer) use this kind of ideas to build and train networks

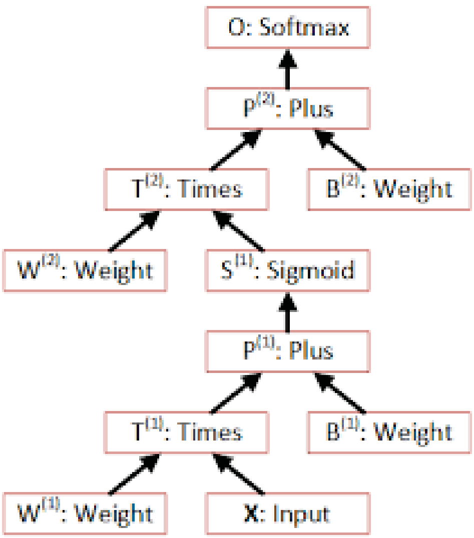


Image taken from CNTK toolkit documentation

Recent tricks of the trade for training

- How to make the networks train better?
  - Better: faster to converge, achieving better validation error
  - Some tricks aim to avoid gradient vanishing
  - Others use randomization techniques to improve generalization
- Older tricks (from 2010): unsupervised (RBM, deep autoencoder) layer by layer initialization
- Better nonlinearities (such as rectified linear units)
- Dropout training
- Maxout
- Batch normalization
- Residue networks
- Ladder networks
- Highway networks

Are neural networks slow?

- Training can be very slow due to large amounts of training data
  - But training is done offline, so it is potentially OK!
  - Trained models can be reused for many problems

- But, inference is very fast, we just feed-forward the input data!
- Usually no lag or latency in processing, appropriate for online processing

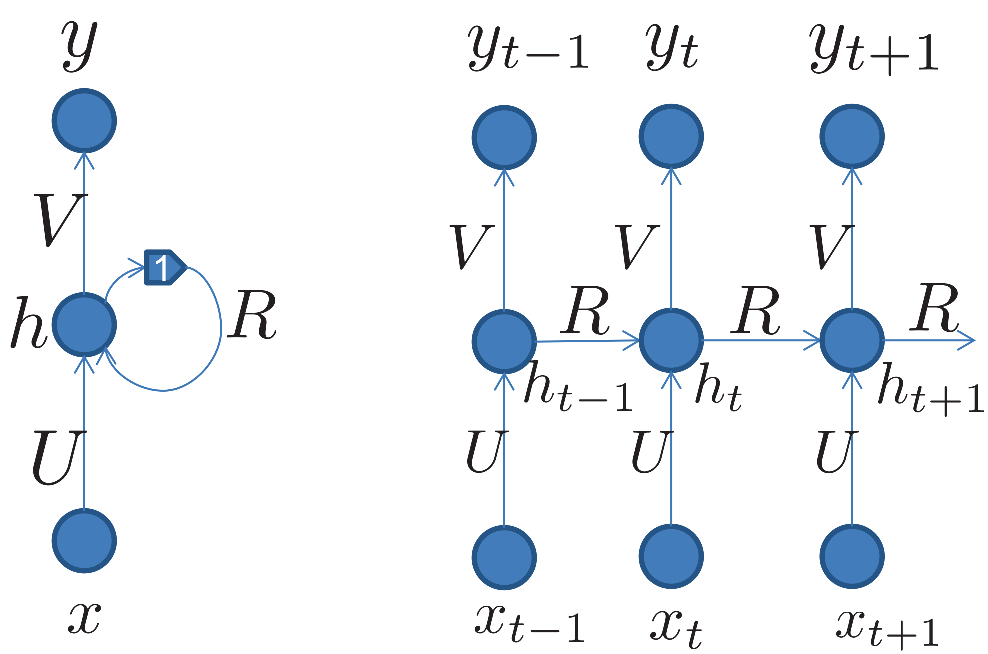
- So final answer: No they are not slow, in contrast they are very FAST to apply

Why need toolkits?

- We require toolkits to define **computational networks** and learn their parameters (**W**) from training data
- We want the toolkits to be **flexible**, so that researcher's can define loss functions, **novel architectures**, saving and reloading networks should be easy
- **Build networks from low level functions**, automatically differentiate using back-propagation
- The code should run efficiently on a GPU without much effort
- NVIDIA has CUDA library, also CUDNN, CUBLAS, CUSPARSE etc., but these are low level libraries
- Need higher level toolkits
- More discussion about toolkits in the appendix

Recurrent neural network (RNN)

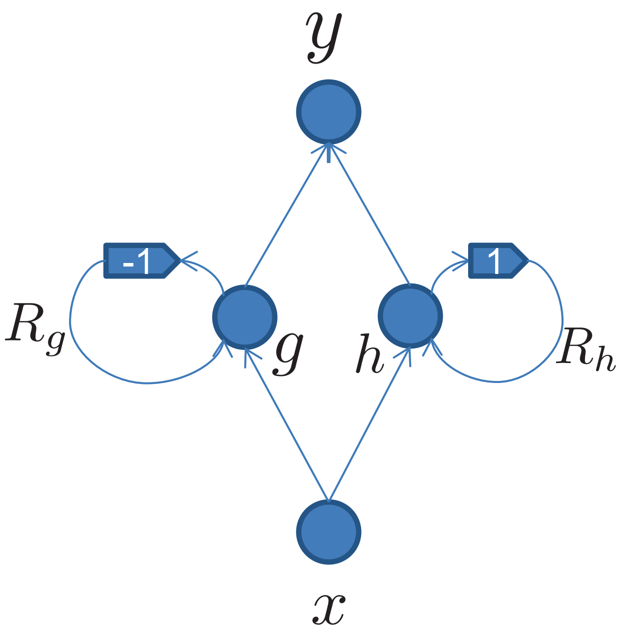
- Recurrent networks process sequential data, have memory

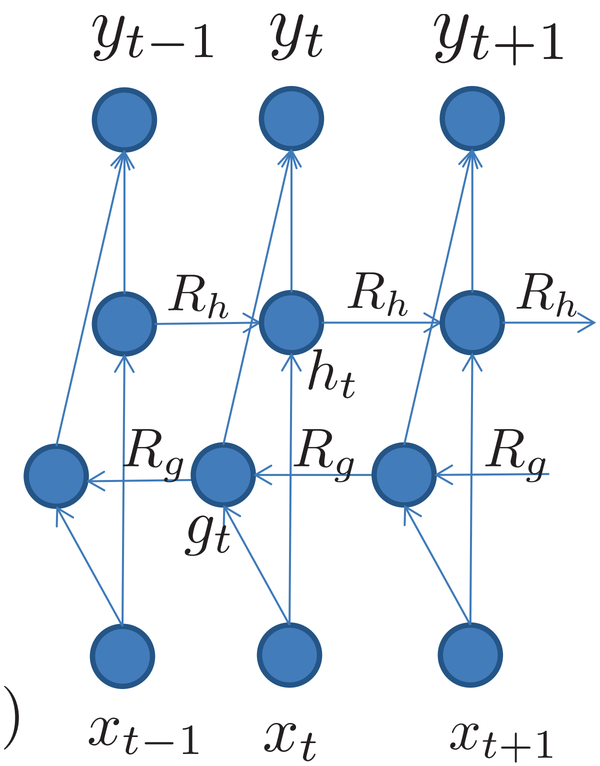


$$
\begin{aligned}
h_t &= \sigma(Rh_{t-1} + Ux_t) \\
y_t &= \sigma(Vh_t)
\end{aligned}
$$

Bi-directional RNN

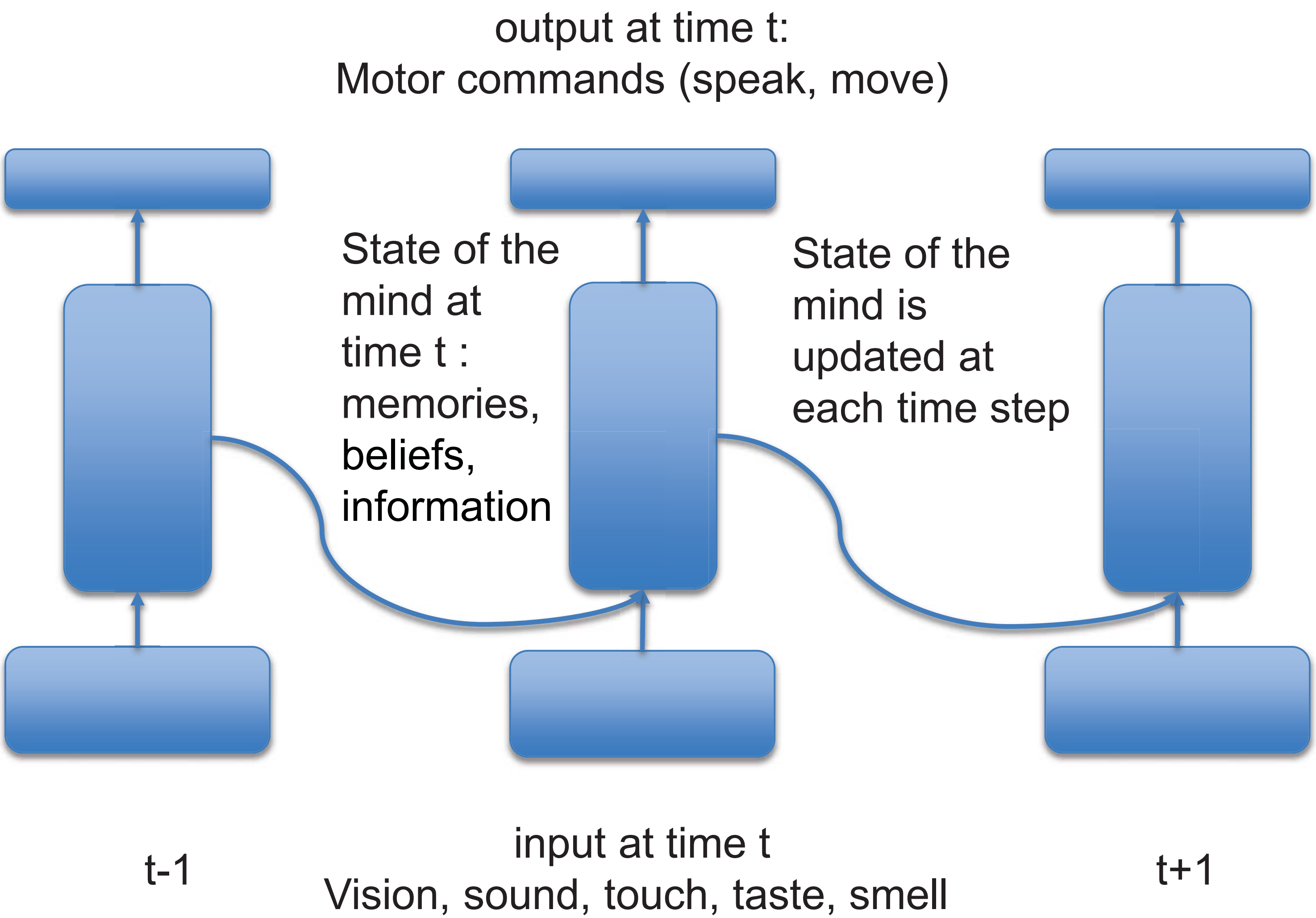
- Bi-directional RNNs have forward and backward memory





$$
\begin{aligned}
h_t &= \sigma(R_h h_{t-1} + U_h x_t) \\
g_t &= \sigma(R_g g_{t+1} + U_g x_t) \\
y_t &= \sigma(V_h h_t + V_g g_t)
\end{aligned}
$$

A simple model for human brain as an RNN



Enhancement/separation problem formulation

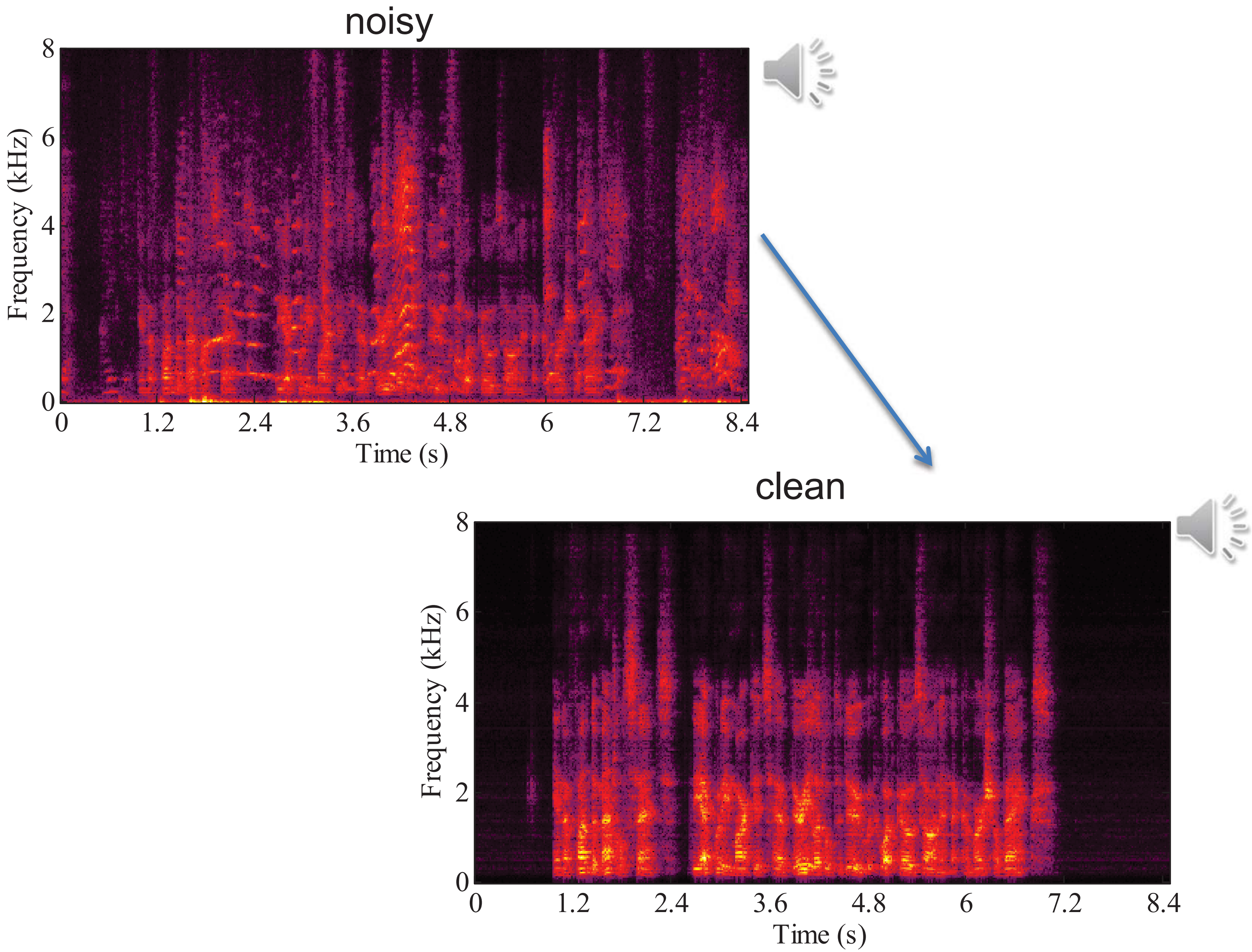
$$y(\tau) = s(\tau) + n(\tau)$$

- Short-time Fourier transform (STFT) domain
- ***y*,*s*,*n***=mixed signal, speech, noise STFTs (complex)

$$y_{t,f} = s_{t,f} + n_{t,f}$$

- Often made assumption: $|y_{t,f}| \approx |s_{t,f}| + |n_{t,f}|$
- Problem: Given mixed signal's STFT **y**, estimate speech STFT **s**
- Assume availability of *training data* for each source type (speech and other), mixtures of them can be formed at various SNRs -> Use machine learning

Ideally we want



The magnitude and the phase

$$s = |s| \exp\{j\theta_s\}$$

- Should we try to estimate both $|s|$ and θ_s ?
- Under complex Gaussian assumptions for speech and noise and some other constraints, MMSE optimal estimate for θ_s is equal to θ_y [Ephraim&Malah 1984, Cohen&Berdugo 2001]
- So, try to estimate the magnitude only and use the mixed signal phase as the phase estimate
- Can we do better than that? May be we can, but in this talk, we restrict ourselves to magnitude prediction

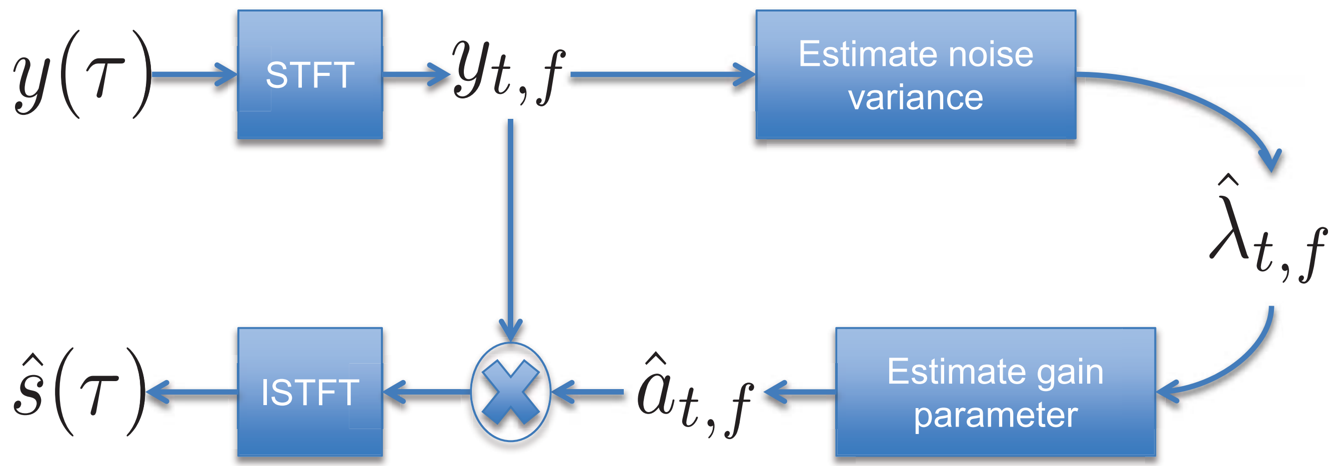
Mask prediction

- Estimate speech as $\hat{s} = \hat{a} \otimes y$ where $\hat{a}$ is a real filter (mostly restricted to [0,1])
  - uses the mixed signal's scaled magnitude and its exact phase
- Boils down to estimation of $\hat{a}_{t,f}$ at each time-frequency bin
- Binary mask: $\hat{a}$ is 0 or 1
  - Assign each TF-bin to one source only
- Ratio mask or soft mask: $\hat{a}$ is between 0 and 1
  - Distribute each TF-bin partially to each source

Learning-free speech enhancement

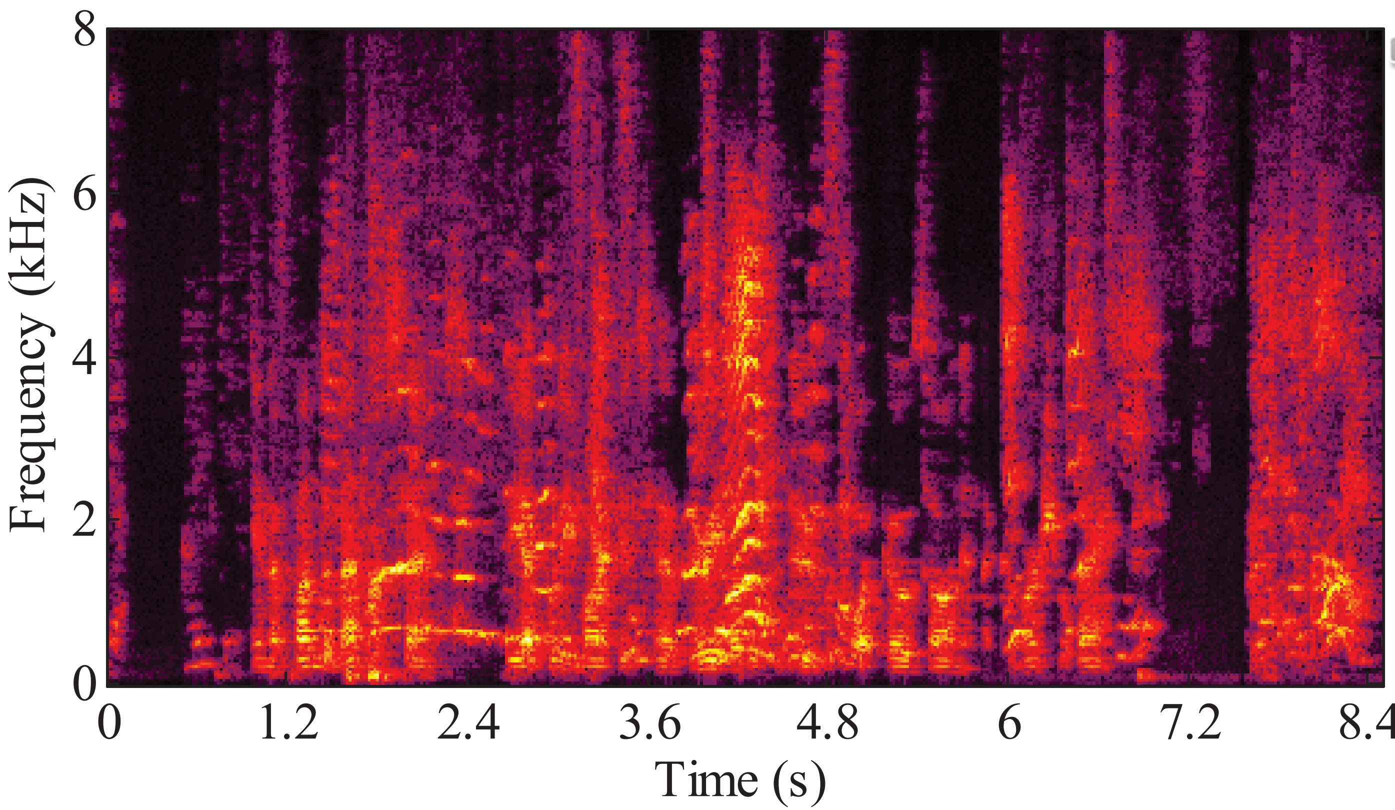
- Spectral subtraction, Wiener filter, MMSE-STSA, LSA, OMLSA
- No machine learning (no prior training data)
- Estimate parameters from utterance at hand
- Assumptions: speech and noise STFTs are independent, both complex Gaussian with mean zero
- Noise is stationary (or slowly varying)
- Minimize MSE in complex, magnitude or log-magnitude domains
- Estimate noise variance from noisy data (using minimum energy averaging, speech presence probability, etc.) which leads to estimation of a gain parameter
- Phase is not estimated but taken from noisy data since it is the MMSE-optimal choice [Ephraim&Malah 1984]

Learning-free methods



- Uncorrelatedness, Gaussianity, stationarity assumptions may not be realistic
- Also, need to estimate some parameters from only a single observation

Performance of OMLSA in non-stationary noise



OMLSA algorithm (best conventional one in our trials)

Machine learning methods

- Focus on the term “source separation” rather than “speech enhancement”
- Age of big data: we have a lot of data and CPU/GPU power to process them
- Machine learning techniques
  - Model-based (already covered)
    - NMF and its variants
    - Other methods
  - Neural networks
    - Deep feed-forward neural nets or MLPs
    - Recurrent NN (RNN)
    - LSTM-RNN

Speech enhancement using Neural Nets: Early days

- Earlier studies [1988-1998] on speech enhancement using neural networks
  - Speech enhancement using time-domain signal segments as inputs [Tamura&Waibel 1988, Tamura 1989, Tamura&Nakamura 1990]
  - Transform domain noise reduction network (enhancing features for speech recognition, concatenate with a recognition network) [Sorensen 1991, Dawson&Sridharan 1992, Moon&Hwang 1993, Wan 1995]
  - Log-spectral domain gain estimation (single t-f bin) [Xie&Compernolle 1994]
  - Other time-domain speech enhancement papers for various applications [Dahl&Claessen 1996, Le&Mason 1996]
  - A survey paper summarizing earlier papers [Wan&Nelson 1998]

What was missing back then?

- Earlier papers used smaller neural networks trained from small amounts of training data
- Deep learning studies did not exist, so efficient training techniques were not present
- Time-domain enhancement performed worse than transform domain enhancement
- Neural networks were in the decline, scientific community believed that other model based methods were superior to neural networks
- There were not widespread databases for consistent comparison of different approaches

Types of recent neural network methods

- Revived interest on speech enhancement/separation using deep neural networks with following goals
  - ***Binary*** mask estimation from noisy spectra using DNNs and other classifiers, essentially a ***classification*** problem [Wang&Wang 2013]
  - ***Soft-mask*** estimation, or directly estimating source spectra from noisy spectra, essentially a ***regression*** problem [Xu&Du&Dai&Lee 2014, Huang&Kim&Johnson&Smaragdis 2014, Weninger&Hershey&LeRoux&Schuller 2014, Wang&Narayanan&Wang 2014]
  - Use DNN as a classifier to check for validity of source estimates while solving a ***constrained optimization problem (****our first trial of DNNs for this problem – details to follow****)*** [Grais&Sen&Erdogan 2014]

Using a DNN as a source verifier

- Summary of our paper: Grais, Sen and Erdogan, "Deep neural networks for single channel source separation," *ICASSP 2014*.
- Inspired from using NMF as a model for each source, we thought we could use a DNN as a model for each source, or better yet, a "discriminative" model, a classifier of sources
- We train a single DNN to classify sources
- Idea: We have three conditions to satisfy to solve the problem:
  - The source estimates should be compatible with their own models (should be classified as source one or two using the trained DNN)
  - The source estimates should sum to the mixed signal
  - The source estimates should be nonnegative
- In the paper, we solve an optimization problem that aims to achieve these conditions together

Formulation

- Consider the problem of source separation where $y = \alpha_1 x_1 + \alpha_2 x_2$ where x_1 and x_2 are normalized sources
- Consider a verifier DNN with two outputs $f_1(x)$ and $f_2(x)$ as indicators for each source and define

$$\begin{aligned} E_1(x) &= (1 - f_1(x))^2 + f_2^2(x) \\ E_2(x) &= f_1^2(x) + (1 - f_2(x))^2 \end{aligned}$$

- Define the set of parameters to be estimated as $\theta = [x_1, x_2, \alpha_1, \alpha_2]$
- Define an objective function to be minimized for θ as

$$E(x_1, x_2, y, \alpha_1, \alpha_2) = E_1(x_1) + E_2(x_2) + \lambda ||\alpha_1 x_1 + \alpha_2 x_2 - y||^2 + \beta \sum_i \min(\theta_i, 0)^2$$

- After solving the optimization problem at test time, obtain source estimates using an adaptive Wiener filter

$$\hat{s}_1 = \frac{(\alpha_1 x_1)^2}{(\alpha_1 x_1)^2 + (\alpha_2 x_2)^2} \otimes y$$

Results

SMR dB	NMF SDR	NMF SIR	NMF SNR	DNN $L=1$ SDR	DNN $L=1$ SIR	DNN $L=1$ SNR	DNN $L=3$ SDR	DNN $L=3$ SIR	DNN $L=3$ SNR
-5	1.79	5.01	3.15	2.81	7.03	3.96	**3.09**	**7.40**	**4.28**
0	4.51	8.41	5.52	5.46	9.92	6.24	**5.73**	**10.16**	**6.52**
5	7.99	12.36	8.62	8.74	**13.39**	9.24	**8.96**	13.33	**9.45**

SMR dB	NMF SDR	NMF SIR	NMF SNR	DNN $L=1$ SDR	DNN $L=1$ SIR	DNN $L=1$ SNR	DNN $L=3$ SDR	DNN $L=3$ SIR	DNN $L=3$ SNR
-5	5.52	15.75	6.30	6.31	**18.48**	7.11	**6.67**	18.30	**7.43**
0	3.51	12.65	4.88	4.23	**16.03**	5.60	**4.45**	15.90	**5.88**
5	0.93	9.03	3.35	1.79	12.94	3.96	**1.97**	**13.09**	**4.17**

speech - piano music separation performance, L is the stacked frame count

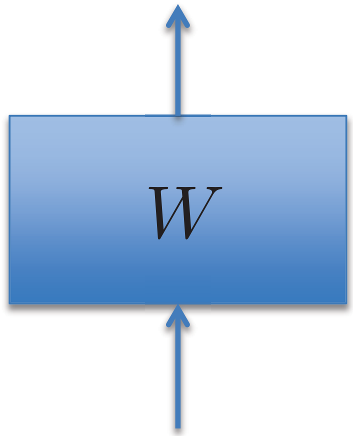
Pros and cons of this approach

- Pros:
  - Does not require training from mixtures, only train a DNN to classify single sources
  - DNN is a better model than an NMF model due to its power of representation and discriminative nature
- Cons:
  - Requires solving an optimization problem at test time: quite slow
  - Requires good initialization of source estimates from an NMF model
- Conclusion:
  - Straightforward method with feed-forward inference seems to work better, which we address next

Neural Net for speech enhancement simplified

$$\hat{s}_w(y) \;/\; \hat{a}_w(y)$$

Enhanced spectrogram, or mask



network

$$y$$

Noisy data

+ lots of data to train from

What should be the input to the network?

- All references seem to agree that using features similar to log-magnitude-spectra is a good type of input to the network
- It was found that using log-mel-filterbank features with 100 Mel filters gave the best result [Weninger&Hershey&LeRoux&Schuller 2014]
- In ASR typically less filters are used, but in enhancement, it looks like a larger number like 100 is necessary
- For DNN: concatenate features from neighboring frames for contextual information (splicing, super-frames, sliding-window)
- For RNN: use single-frame features, it handles context directly

Network training loss functions

- Divergence measures that can be used in the network training loss function

$$D(\hat{s}_w(y), |s|) = \sum_{tf} D([\hat{s}_w(y)]_{t,f}, |s_{t,f}|)$$

- Squared Euclidean distance (relates to SNR) [Huang&Kim&Johnson&Smaragdis 2014, Weninger&Hershey&LeRoux&Schuller 2014]

$$D([\hat{s}_w(y)]_{t,f}, |s_{t,f}|) = ([\hat{s}_w(y)]_{t,f} - |s_{t,f}|)^2$$

- Log-spectral distance (LSD): Predict log-mag-spectra and use squared Euclidean distance between logarithms (perceptual) [Xu&Du&Dai&Lee 2014]

$$D([\hat{l}_w(y)]_{tf}, \log|s_{tf}|) = ([\hat{l}_w(y)]_{t,f} - \log|s_{t,f}|)^2$$

Effects of various network losses

- Predicting log-spectra and using log-spectral distance tends to oversmooth the large values and has some global variance problems which needs to be post-corrected [Xu&Du&Dai&Lee 2014]
- Using squared error measure relates more to SNR, however it may not be perceptually optimal when speech power is low
- KL, IS divergences (Bregman, beta, alpha divergences and others) need to be investigated
- More investigation and comparison of different versions may be required

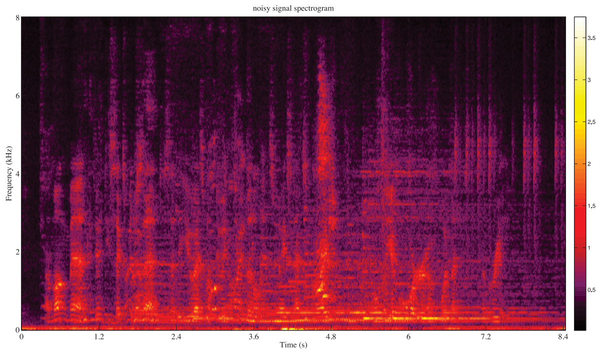
Predict mask or spectra?

Should the network predict a clean spectrogram or a mask?

- It can predict a magnitude spectrogram with a least-squares divergence function
  $D([\hat{s}_w(y)]_{t,f}, |s_{t,f}|) = ([\hat{s}_w(y)]_{t,f} - |s_{t,f}|)^2$
- Or, it can predict a mask
  - with a mask approximation (MA) least-squares loss
    $D([\hat{a}_w(y)]_{t,f}, a^*_{t,f}) = ([\hat{a}_w(y)]_{t,f} - a^*_{t,f})^2$, where a^* is an ideal mask
  - or for a binary mask, we can use binary-cross-entropy loss
    $D([\hat{a}_w(y)]_{t,f}, a^*_{t,f}) = -a^*_{t,f} \log[\hat{a}_w(y)]_{t,f} - (1 - a^*_{t,f}) \log(1 - [\hat{a}_w(y)]_{t,f})$
  - or with a magnitude spectrum approximation (MSA) loss
    $D([\hat{a}_w(y)]_{t,f}, |s_{t,f}|) = ([\hat{a}_w(y)]_{t,f}|y_{t,f}| - |s_{t,f}|)^2$

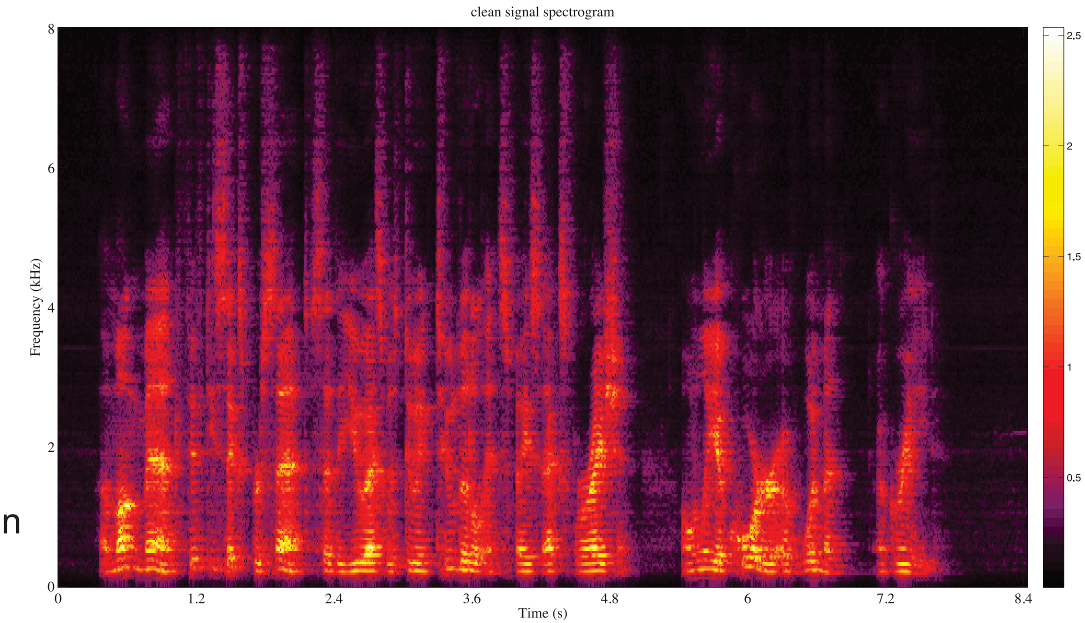
[Weninger&Hershey&LeRoux&Schuller 2014] found MSA is better than MA,
[Wang&Narayanan&Wang 2014] found MSA is better than LS loss

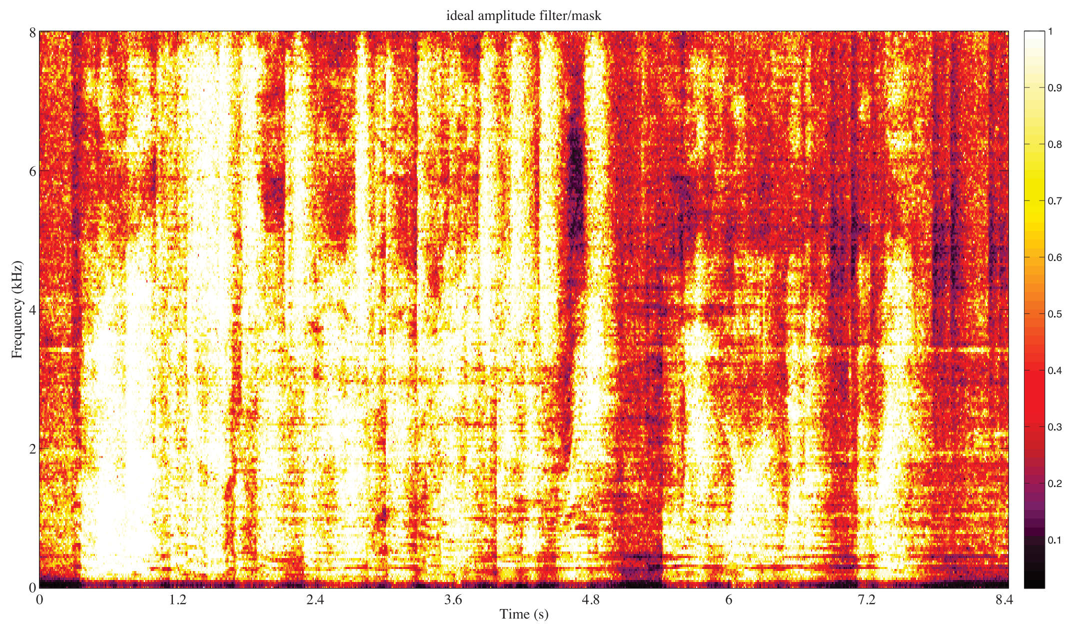
Mask versus spectra



noisy

clean





Ideal amplitude mask
limited to [0,1] range

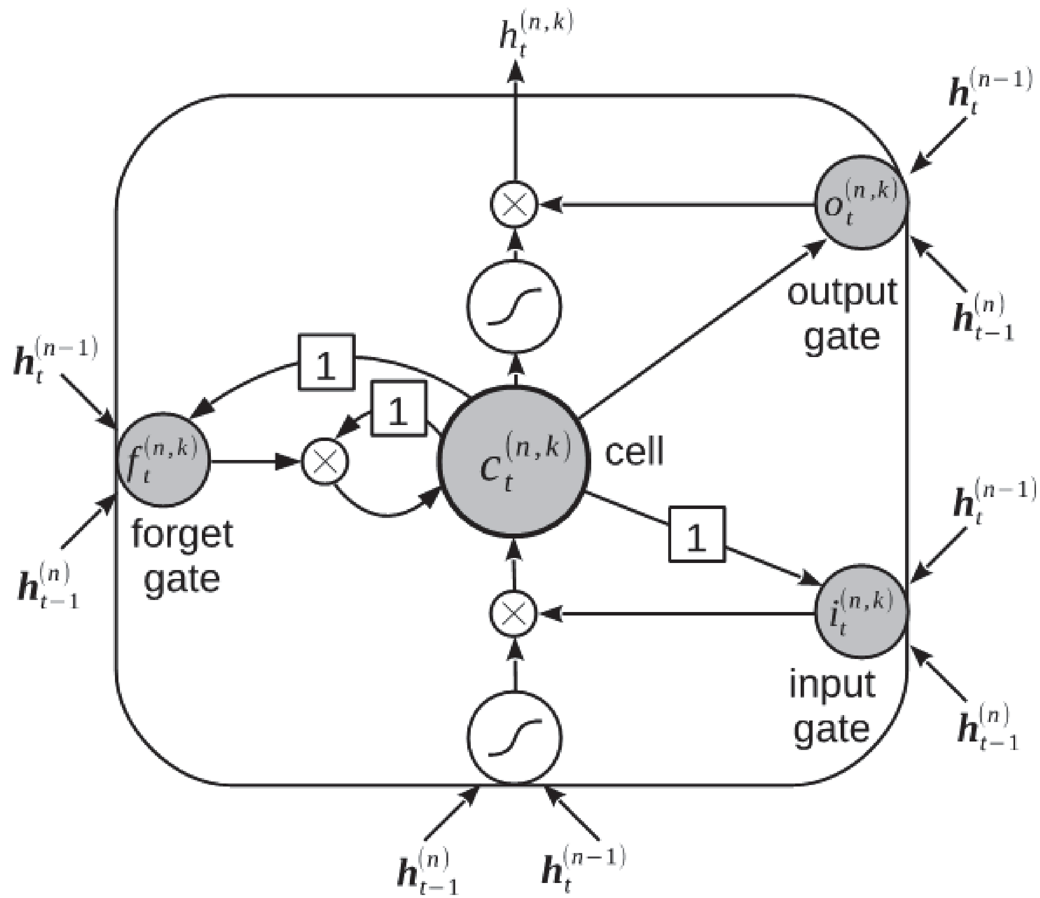
Why predict mask?

- Mask value can be restricted to be in the range [0,1] and we can use a logistic sigmoid output layer to predict it
- Direct spectral prediction may require using a linear or rectified linear output layer with an infinite range of output
- Prediction of the spectra may also yield over-smoothing effects in general due to the regression-to-the-mean effect (regardless of predicting log-spectra or not)
- When the signal is clean, predicting a mask of 1, can directly pass the clean signal to the output, giving “perfect reconstruction” without having to learn to produce the signal.

Problems with RNNs

- It was found that DNN and RNN performance for source separation are very close. [Huang&Kim&Johnson&Smaragdis 2014]
- Shouldn't RNN be better since it uses potentially longer context?
  - Yes, but hard to learn the parameters
- Weights learned with back propagation through time (BPTT)
- BPTT gradients get too small (or too large) as we back-propagate from t to t-T where T is large
- Network forgets previous "events" due to shrinkage of earlier hidden node activations as time progresses
- Need to "preserve" the earlier (t-T) hidden node activations to be able to use them in predictions at time t
- One solution: Long short-term memory (LSTM) RNNs [Hochreiter&Schmidhuber 1997]

LSTM memory cell



Taken from [Weninger et.al. 2014].

LSTM forward computations

$$
\begin{aligned}
\mathbf{h}_0^{(1,\ldots,N)} &:= \mathbf{0}, \mathbf{c}_0^{(1,\ldots,N)} := \mathbf{0}, \\
\mathbf{h}_t^{(0)} &:= \tilde{\mathbf{x}}_t, \\
\mathbf{f}_t^{(n)} &:= \sigma(\mathbf{W}^{f,(n)}[\mathbf{h}_t^{(n-1)}; \mathbf{h}_{t-1}^{(n)}; \mathbf{c}_{t-1}^{(n)}; 1]) \\
\mathbf{i}_t^{(n)} &:= \sigma(\mathbf{W}^{i,(n)}[\mathbf{h}_t^{(n-1)}; \mathbf{h}_{t-1}^{(n)}; \mathbf{c}_{t-1}^{(n)}; 1]) \\
\mathbf{c}_t^{(n)} &:= \mathbf{f}_t^{(n)} \otimes \mathbf{c}_{t-1}^{(n)} \\
&+ \mathbf{i}_t^{(n)} \otimes \tanh\left(\mathbf{W}^{c,(n)}[\mathbf{h}_t^{(n-1)}; \mathbf{h}_{t-1}^{(n)}; 1]\right), \\
\mathbf{o}_t^{(n)} &:= \sigma(\mathbf{W}^{o,(n)}[\mathbf{h}_t^{(n-1)}; \mathbf{h}_{t-1}^{(n)}; \mathbf{c}_t^{(n)}; 1]) \\
\mathbf{h}_t^{(n)} &:= \mathbf{o}_t^{(n)} \otimes \tanh(\mathbf{c}_t^{(n)}), \\
\tilde{\mathbf{y}}_t &:= \mathbf{W}^{(N+1)}[\mathbf{h}_t^{(N)}; 1].
\end{aligned}
$$

Results with LSTM networks

- [Weninger&Hershey&LeRoux&Schuller 2014] found that using a two layer LSTM network with a MSA objective gave ***much better*** results than an NMF baseline and also is better than using a DNN

Improvements on the baseline method

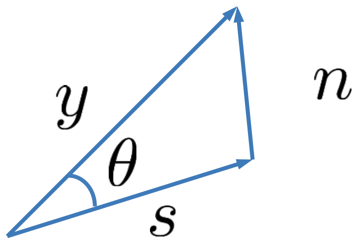
- How can we improve LSTM based, mask-predicting, magnitude spectrogram approximation (MSA) network introduced in [Weninger&Hershey&LeRoux&Schuller 2014] whose performance is quite good already
  - Use a *phase-sensitive* loss function
  - Use *ASR alignments* as additional evidence for enhancement, iteration of ASR and enhancement
  - Use *bidirectional* LSTM

Analyzing oracle masks

- Use the mixed signal phase θ_y and try to estimate only the magnitude of speech $|s|$
- How well can we perform (say in terms of SNR) if we estimated the magnitude "perfectly" ?
- But: "perfectly" depends on what you consider to be perfect!
- There are various options to consider!

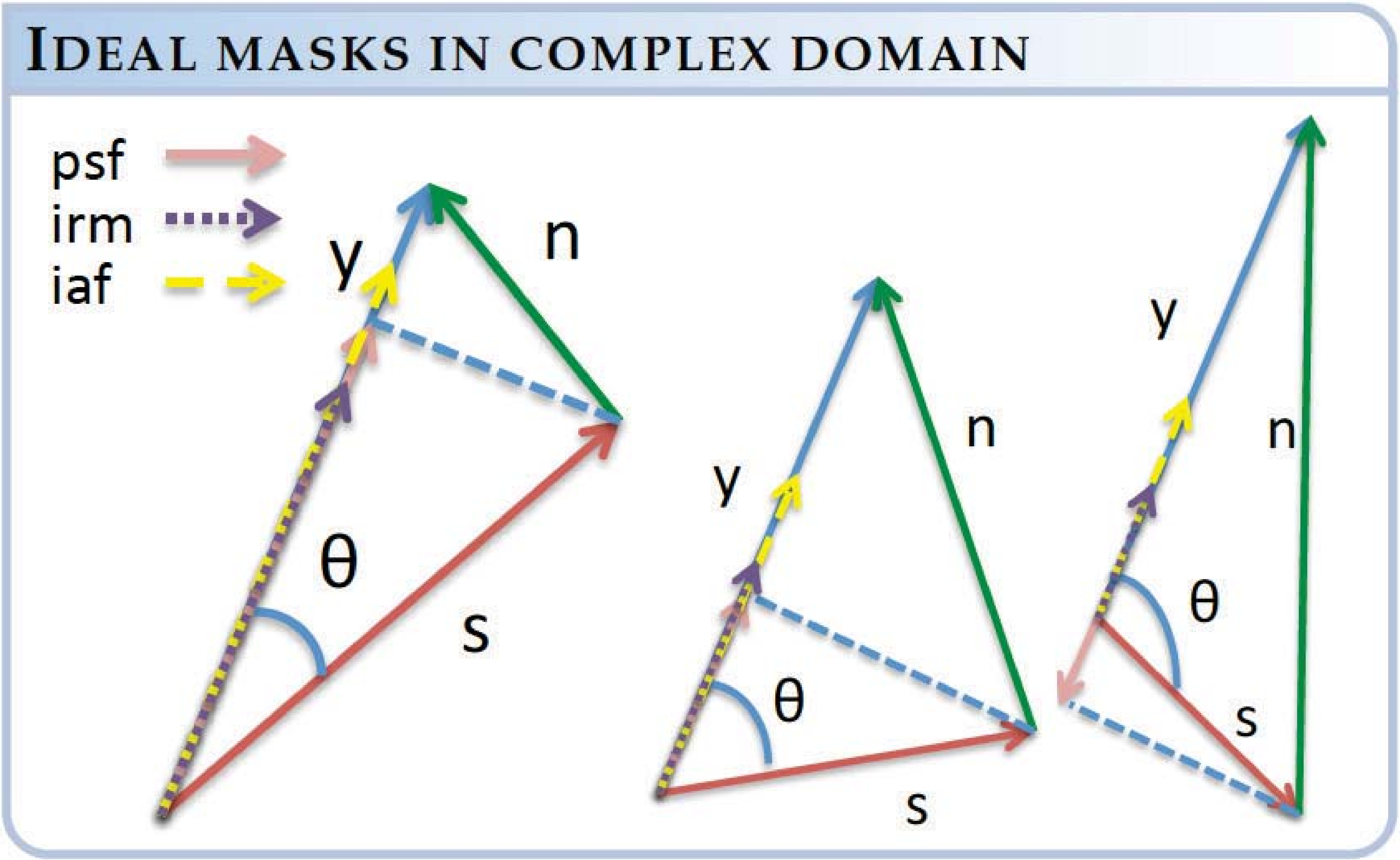
Oracle masks

$$y = s + n \qquad |y| \approx |s| + |n|$$

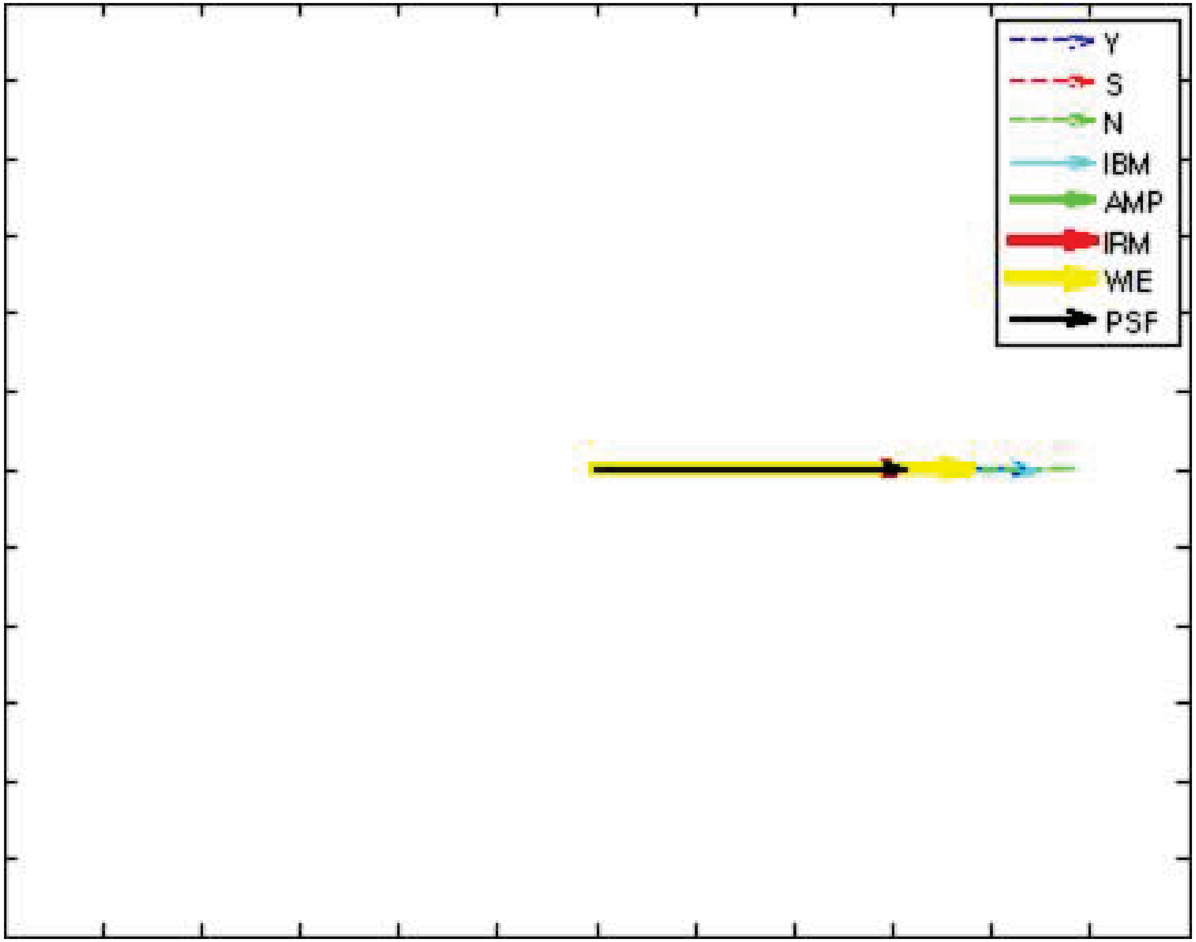


target mask/filter	formula	optimality principle
IBM:	$a^{\mathrm{ibm}} = \delta(\lvert s\rvert > \lvert n\rvert)$,	max SNR given $a \in \{0, 1\}$
IRM:	$a^{\mathrm{irm}} = \dfrac{\lvert s\rvert}{\lvert s\rvert + \lvert n\rvert}$,	max SNR given $\theta_{\mathrm{s}} = \theta_{\mathrm{n}}$
"Wiener like":	$a^{\mathrm{wf}} = \dfrac{\lvert s\rvert^2}{\lvert s\rvert^2 + \lvert n\rvert^2}$,	max SNR, expected power
ideal amplitude:	$a^{\mathrm{iaf}} = \lvert s\rvert / \lvert y\rvert$,	exact $\lvert\hat{s}\rvert$, max SNR $\theta_{\mathrm{s}} = \theta_{\mathrm{y}}$
phase-sensitive:	$a^{\mathrm{psf}} = \dfrac{\lvert s\rvert}{\lvert y\rvert} \cos(\theta)$,	max SNR given $a \in \mathbb{R}$
ideal complex:	$a^{\mathrm{icf}} = s/y$,	max SNR given $a \in \mathbb{C}$

Illustrating ideal masks



Oracle masks movie



Oracle mask results

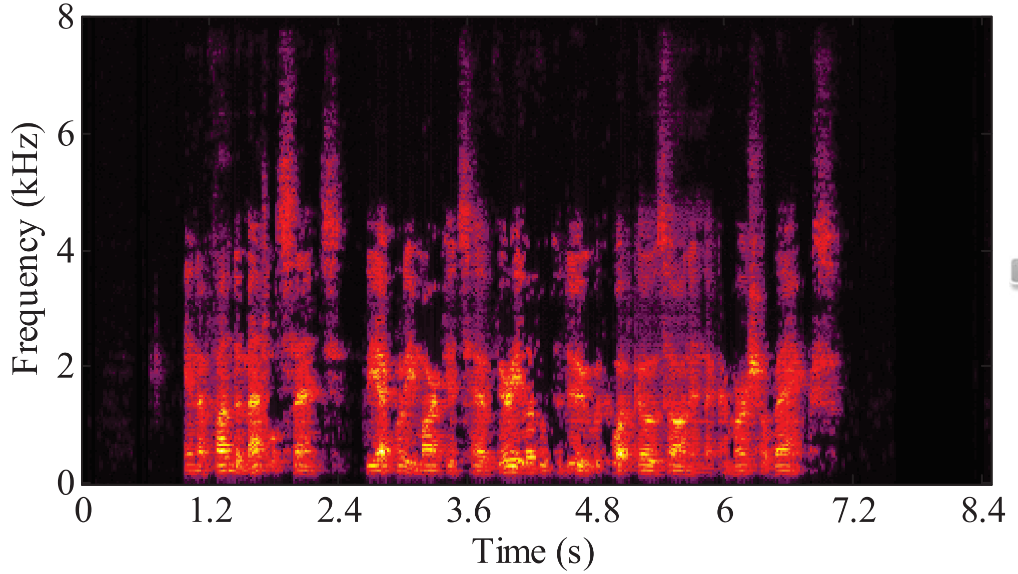
Ideal masks CHiME-2 dev set SDR (in dB)

dt	-6 dB	9 dB	Avg
IBM	14.56	20.89	17.59
IRM	14.13	20.69	17.29
"Wiener-like"	15.20	21.49	18.21
ideal amplitude	13.97	21.35	17.52
phase sensitive filter	17.74	24.09	**20.76**
truncated PSF	16.13	22.49	**19.17**

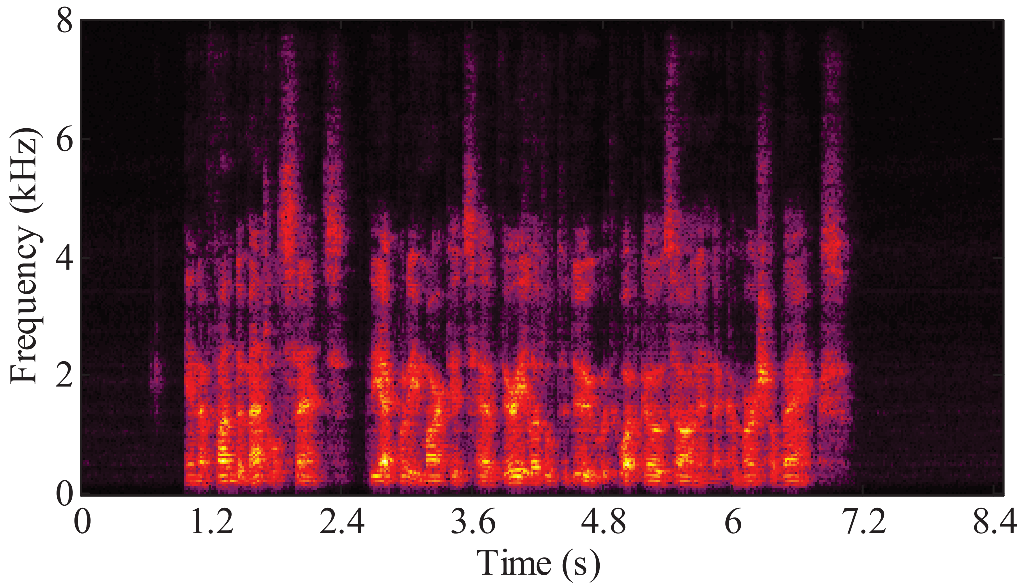
Phase-sensitive filter (PSF) ideal or truncated to [0,1] gives much higher SDR value than others!

Spectrograms obtained using oracle masks

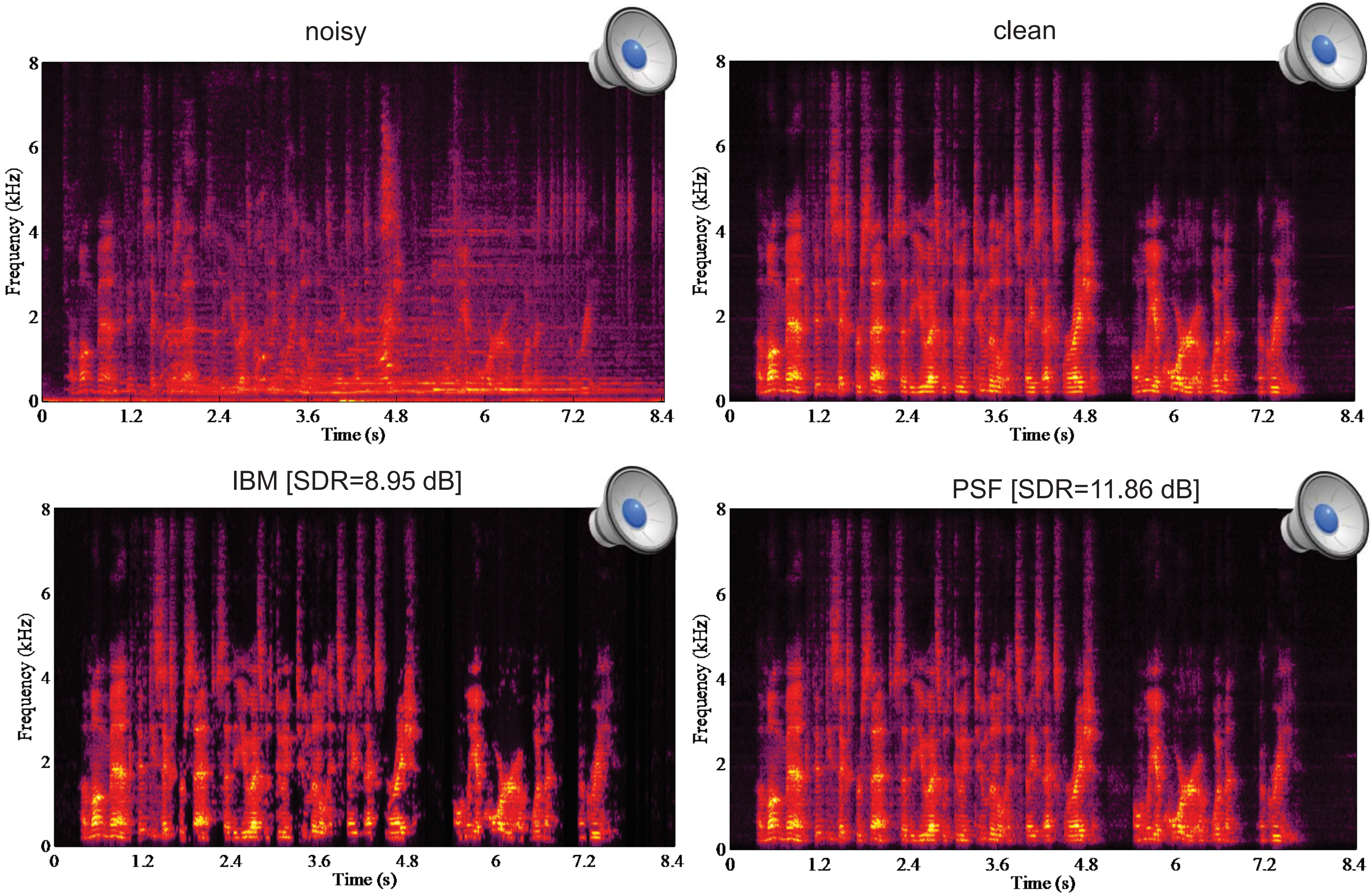
IBM



PSF



Spectrograms obtained using oracle masks



Phase-sensitive approximation loss

Inspired by the phase-sensitive ideal filter, we introduce a phase-sensitive approximation (PSA) divergence

$$D_{\mathsf{PSA}}([\hat{a}_w(y)]_{t,f}, s_{t,f}) = |[\hat{a}_w(y)]_{t,f} y_{t,f} - s_{t,f}|^2$$

which is equivalent to $D_{\mathsf{PSA}}([\hat{a}_w(y)]_{t,f}, s_{t,f}) = ([\hat{a}_w(y)]_{t,f}|y_{t,f}| - |s_{t,f}|\cos(\theta_{t,f}))^2$ where $\theta_{t,f}$ is the angle between $s_{t,f}$ and $y_{t,f}$.

- Using PSA, the network still only predicts a real mask and NOT the phase. When using PSA, the network learns to “shrink” its output masks by cosine of the angle during training which is known
- It also shrinks the masks during test time by implicitly guessing the cosine of the angle which is small when the network thinks the noise is high for that time-frequency bin
- In contrast, previous divergence functions
  - $D_{\mathsf{MA}}([\hat{a}_w(y)]_{t,f}, a^*_{t,f}) = ([\hat{a}_w(y)]_{t,f} - a^*_{t,f})^2$
  - $D_{\mathsf{MSA}}([\hat{a}_w(y)]_{t,f}, |s_{t,f}|) = ([\hat{a}_w(y)]_{t,f}|y_{t,f}| - |s_{t,f}|)^2$

Relation to time-domain loss

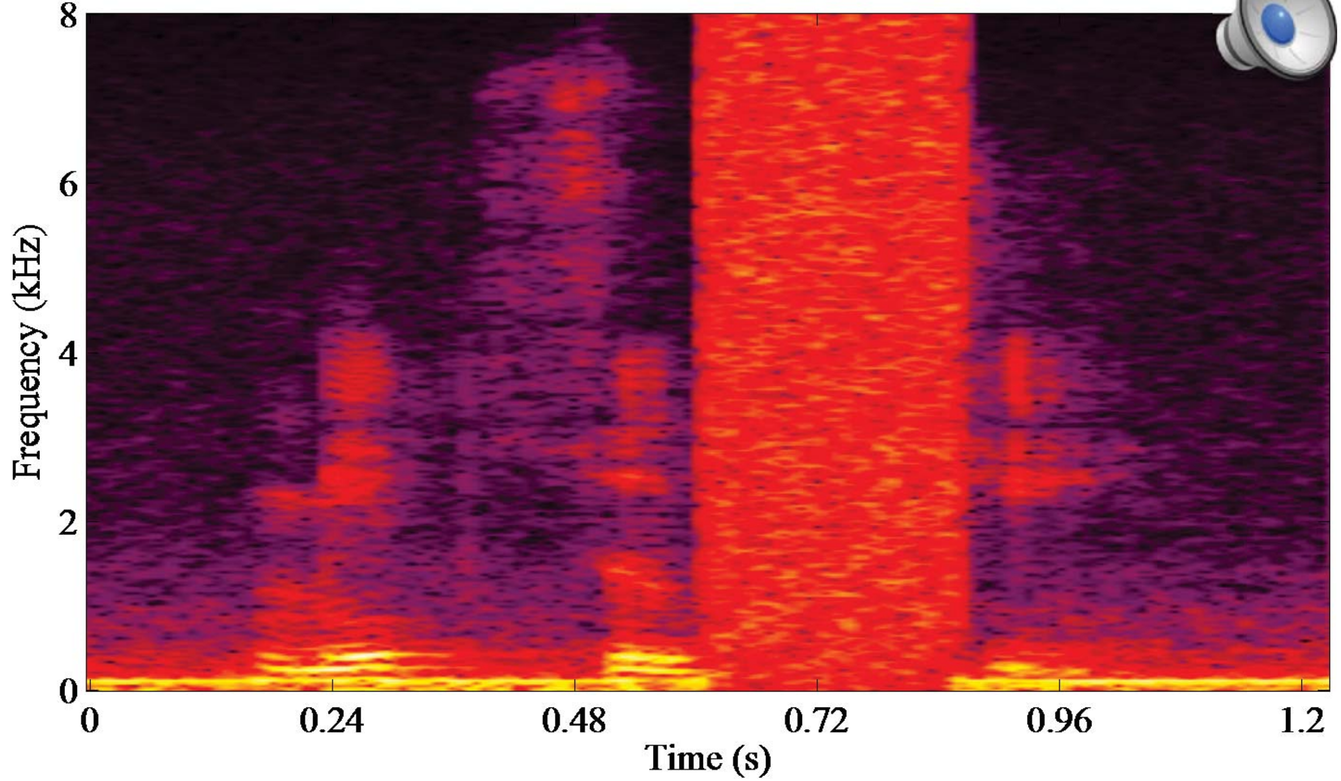
- Recently, [Wang & Wang 2015] introduced using time-domain least squares error in training neural networks for speech enhancement using DNNs
- They also use the noisy phase to reconstruct the time domain signal and calculate network's training loss as the squared error in the time domain
- Phase-sensitive loss function is equivalent to time-domain least squares loss function since they are both maximizing SNR
  - Time domain and frequency domain errors are equivalent
  - Due to Parseval's theorem

Using ASR to improve speech enhancement

- ASR has access to a language model which the LSTM network does not know about
- Additional information coming from an ASR system can help improve enhancement
- We use additional “alignment information” inputs, which are basically obtained from alignment of frames with the one-best ASR decoding result
- A simple first step to achieve integration of ASR and speech separation/enhancement

A concocted example

ASR will help here



Transcription: **Mit** **su** **bish** **i**

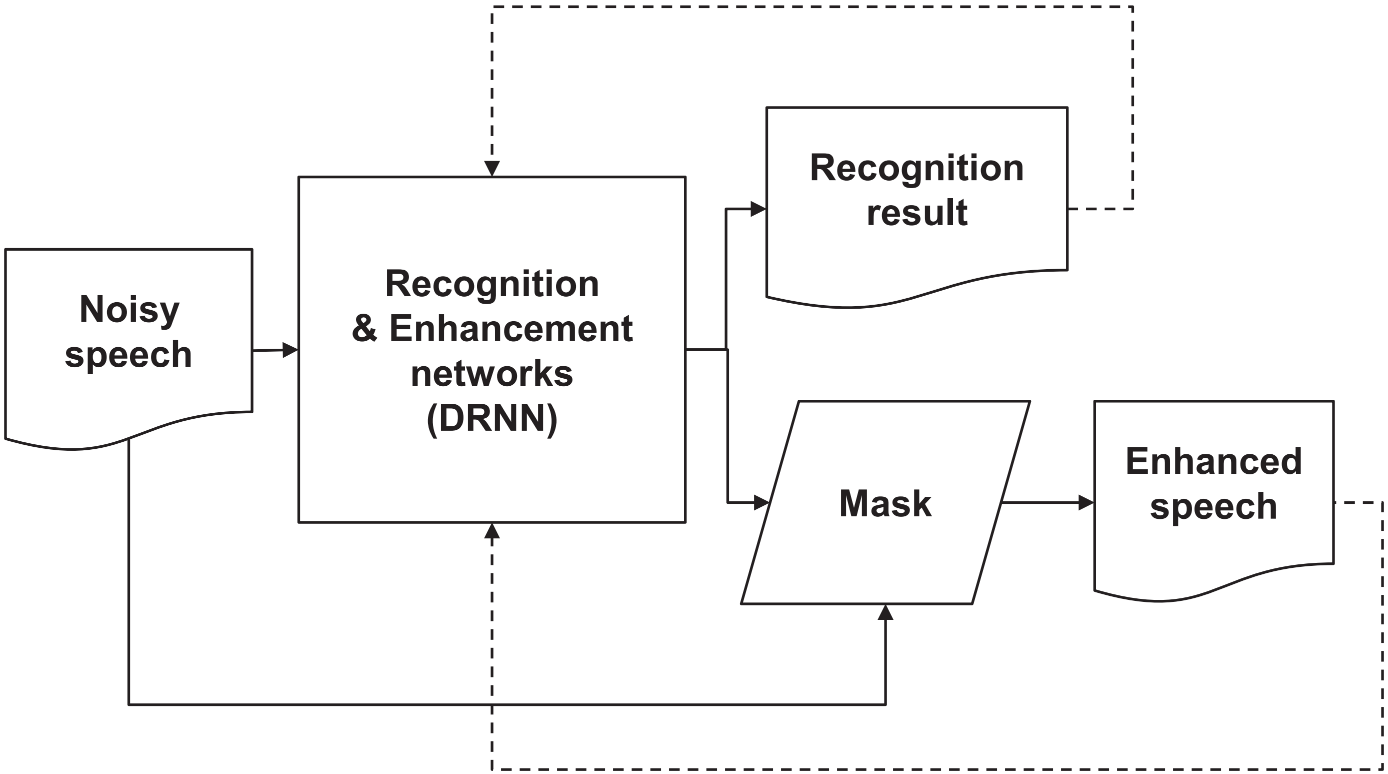
ASR alignment information – how to use

- One possible way: concatenate "one-hot state alignment vectors" to the noisy log-mel-filterbank input
- Another way: use "mean alignment vectors" for each state
- Mean alignment vector = "*average of the log-mel-filterbank features belonging to the ASR state*"
- We found ignorable difference in performance between different methods of providing the alignment information
- We concatenate "mean alignment vector" for the active state at the current frame to the noisy log-mel-filterbank input to obtain the results in this talk

Bidirectional LSTM

- Since we use ASR, no need to restrict ourselves to low-latency real-time techniques
- Bidirectional LSTM networks have the same algorithmic latency as an ASR system, so we can use them
- BLSTM uses contextual information from past and future events that help predict the correct output at each frame
- We use single frame inputs in LSTM and BLSTM
- Let BLSTM learn which past and future events are relevant for prediction at the current frame

Iterated enhancement & ASR



Enhance → ASR → Enhance → ASR

Neural network learning parameters/tricks

- Layer-by-layer supervised pre-training (aka. Microsoft style)
- Whenever possible, initialize from an earlier trained network
  - Initially train with mask approximation in Mel-domain, then switch to signal approximation in spectral domain
- Stochastic gradient with mini-batch size of 50 utterances
- Learning rate (per whole training data) 1e-6
- Momentum with momentum weight 0.9
- Sequence shuffling
- Normalize input data to mean 0 and variance 1
- Add Gaussian noise to input with stdev 0.1 for robustness
- Validation using monitoring of development set loss
- Wait 20 epochs before no more improvement on validation loss to stop training

SDR/SIR results on CHiME-2 eval/test set

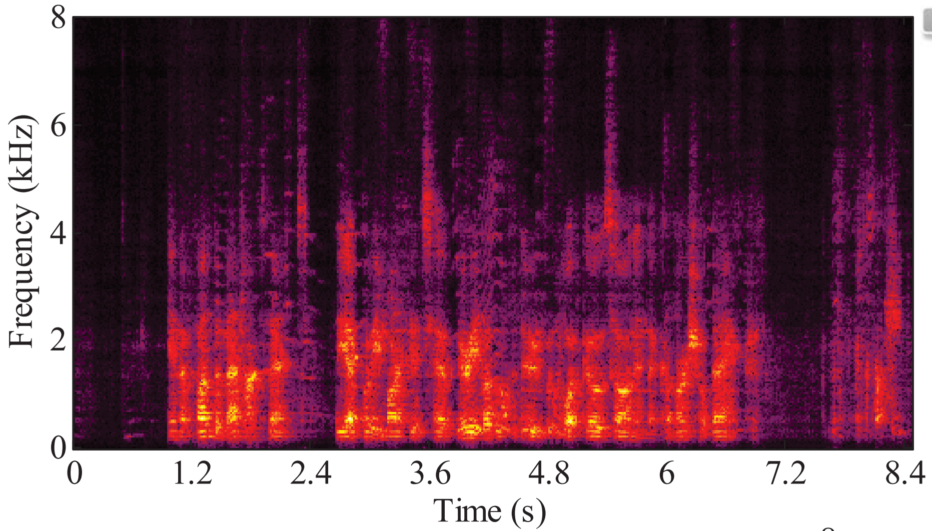
Average SDR and SIR in dB (higher is better)

Network	Cost	Input	Ave-SDR	Ave-SIR
LSTM 2x256	MSA	mfb	13.83	17.53
BLSTM 2x384	MSA	mfb	14.22	18.24
LSTM 2x256	PSA	mfb	14.14	19.20
BLSTM 2x384	PSA	mfb	14.51	19.78
BLSTM 2x384	PSA	mfb+align	**14.75**	**20.46**

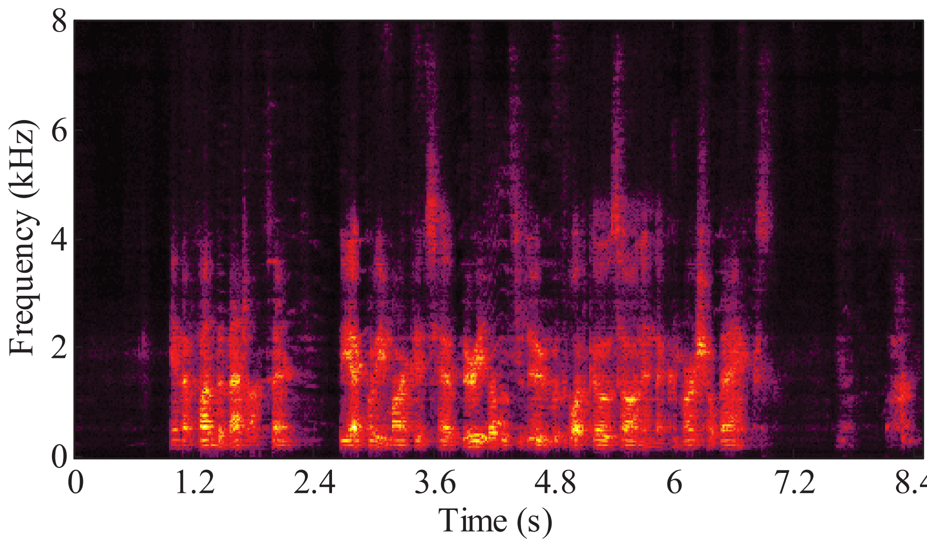
mfb=log-mel-filterbank, 100 dimensional
align=average log-mel-filterbank for the active state, 100 dimensional

Spectrograms

LSTM-MSA [SDR=8.26]



BLSTM-PSA-Align [SDR=10.51]



Recognition results with enhanced speech - CHiME-2

GMM-HMM system trained with mixed training data (multi-condition training)
and retrained with enhanced training data

Enh. method/WER%	**No retraining**		**Retrained with enhanced**	
	Dev	Eval	Dev	Eval
Baseline	54.17	47.90	54.17	**47.90**
OMLSA	59.08	54.20	53.59	48.65
Sparse NMF	51.22	46.30	45.22	39.13
DNN-MSA	36.68	29.72	36.13	29.03
LSTM-MSA	31.45	25.01	31.63	25.44
BLSTM-SSA-PSA	25.52	**19.81**	24.79	**19.11**

Single channel results
SSA=speech state aware (uses ASR info)

DNN-HMM recognizer with beamformed CHiME-2 data

Enhancement method	WER Dev	WER Eval
BF	25.64	21.12
2ch-NMF	25.13	19.46
BF-LSTM-MSA	19.03	14.82
BF-LSTM-PSA	19.20	14.63
BF-BLSTM-MSA	18.35	14.47
BF+SSA-BLSTM-MSA	18.41	14.25
BF+SSA-BLSTM-PSA	18.19	14.24
BF+ENH+SSA-BLSTM-MSA	**18.16**	**13.95**
BF+ENH+SSA-BLSTM-PSA	18.28	**13.95**

DNN target states from clean data alignment
with sequentially discriminative training

Recent studies: complex mask prediction

- Phase prediction is hard, however there has been recent studies on predicting a complex mask [Williamson&Wang 2016] which is equivalent to predicting the phase
- Complex mask prediction is performed by predicting real and imaginary parts of the ideal mask
- An ideal complex mask can take values from minus infinity to infinity
- The range of the ideal mask is squeezed to be in a limited range [-K,K], and after prediction the range is un-squeezed
- This helps in some datasets, but it is little worse than phase-sensitive mask in some other datasets

Recent studies: joint enhancement/recognition

- Joint enhancement/recognition networks for noise robust ASR [Wang&Wang 2016, Gao&Du&Dai&Lee 2015]
- The basic idea (which was also there in studies in 1990's) is to concatenate enhancement and recognition networks
- One can start training with an enhancement loss function and then switch to a cross-entropy (classification) loss function
- Feature extraction is built into the network, or enhancement is done in the feature domain
- Mask prediction can still be employed within the network

Appendix

-more information on toolkits

- additional information

Python based toolkits

- Theano (U Montreal Y Bengio lab)
  - Uses symbolic language to define a network
  - Compiles the network in C++ to run on GPU
  - Hard to debug errors in code
  - A bit hard to learn all details
- Tensorflow (Google)
  - Backed by google, large user base
  - Rapidly changing and expanding
- Theano wrappers
  - Keras (also wraps Tensorflow)
  - Lasagne
  - Theanets

Other python based toolkits

- Chainer
  - Easy to learn
  - Seems easier to debug than Theano
- MXNet
  - Claims to be flexible, fast
  - Seems a bit harder to learn

C++ toolkits

- Torch 7
  - Generic toolkit, used a lot by ML researchers
  - LUA based scripting on top of C++ low level code
- Caffe
  - Mostly for vision problems
  - But can be used for RNNs too
  - Flexible network definition, prototxt, python interface
- CNTK (computational network toolkit)
  - Microsoft Research – originated from speech group
  - Scripts for defining networks
  - Highly efficient code
- Currennt
  - Good for LSTMs, but not flexible

Matlab

- Matlab usually considered slow for learning deep nets
- Still, a lot of toolkits exist
- Many researchers make their matlab code available
- You can write your own matlab code as well
- Some resources:
  - http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial (very good to understand neural nets for beginners)
  - Matlab's neural network toolbox - old one, and not specifically for deep learning
  - Many different ones come up in google searches, need a ranking system among them

Which toolkit should I learn and use?

- It takes a lot of time to learn one toolkit
- Choose one according to your needs and learn it
- You may need to modify the code, so learn how the toolkit works internally, not as a black box
- Avoid ones that keep changing too much internally
- Start with a toolkit after asking around for advice and write some code in it to get a feel
- It may be wise to choose a toolkit recommended by people around you and for which there is immediate help available
- Online forums are also very helpful, so choose one with a large community of support

Other resources

- http://deeplearning.net
- Coursera courses: Andrew Ng's machine learning, Geoff Hinton's neural networks courses
- Online book: http://neuralnetworksanddeeplearning.com
- Yoshua Bengio's Deep Learning book http://deeplearningbook.org

References-1

- [Tamura&Waibel 1988] S. Tamura and A. Waibel, “Noise reduction using connectionist models,” in Proc. ICASSP, 1988, pp. 553–556.
- [Tamura 1989] S. Tamura, “An analysis of a noise reduction neural network,” in Proc. ICASSP, 1989, pp. 2001–2004.
- [Tamura&Nakamura 1990] S. Tamura and M. Nakamura, “Improvements to the noise reduction neural network,” in Proc. ICASSP, 1990, pp. 825–828.
- [Sorensen 1991] H. B. D. Sorensen, "A cepstral noise reduction multi-layer neural network," *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, Toronto, Ont., 1991, pp. 933-936 vol.2.
- [Dawson&Sridharan 1992] M. Dawson and S. Sridharan, “Speech enhancement using time delay neural networks,” Proc. Fourth Australian International Conf. on SST, pp 152-5, Dec 1992.
- [Moon&Hwang 1993] Moon, Seokyong, and J-N. Hwang. "Coordinated training of noise removing networks." *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*. Vol. 1. IEEE, 1993.
- [Le&Mason 1996] T. Le and J. Mason, “Artificial neural networks for nonlinear time-domain filtering of speech,” IEE Proceedings on Vision, Image and Signal Processing, vol. 143, no. 3, pp. 149–154, 1996.
- [Dahl&Claesson 1996] M. Dahl and I. Claesson, “A neural network trained microphone array system for noise reduction,” in Proc. NNSP, 1996, pp. 311–319.
- [Wan&Nelson 1998] E. A. Wan and A. T. Nelson, “Networks for speech enhancement,” in Handbook of neural networks for speech processing, S. Katagiri, Ed. Artech House, 1998.
- [Ephraim&Malah 1984] Ephraim, Yariv, and David Malah. "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator." *Acoustics, Speech and Signal Processing, IEEE Transactions on* 32.6 (1984): 1109-1121.

References-2

- [Hochreiter&Schmidhuber 1997] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.
- [Cohen&Berdugo 2001] Cohen, Israel. "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging." *Speech and Audio Processing, IEEE Transactions on* 11.5 (2003): 466-475.
- [Lee&Seung 2001] Lee, Daniel D., and H. Sebastian Seung. "Algorithms for non-negative matrix factorization." *Advances in neural information processing systems*. 2001.
- [Smaragdis&Brown 2003] Smaragdis, Paris, and Judith C. Brown. "Non-negative matrix factorization for polyphonic music transcription." *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*. IEEE, 2003.
- [Schmidt&Olsson 2006] Schmidt, Mikkel, and Rasmus Olsson. "Single-channel speech separation using sparse non-negative matrix factorization." (2006).
- [Vincent&Gribonval&Fevotte 2006] E Vincent, R Gribonval, C Févotte, “Performance measurement in blind audio source separation,” Audio, Speech, and Language Processing, IEEE Transactions on 14 (4), 1462-1469, 2006.
- [Hu&Loizou 2008] Hu, Yi, and Philipos C. Loizou. "Evaluation of objective quality measures for speech enhancement." *Audio, Speech, and Language Processing, IEEE Transactions on* 16.1 (2008): 229-238.
- [Virtanen&Cemgil&Godsill 2008] Virtanen, Tuomas, Ali Taylan Cemgil, and Simon Godsill. "Bayesian extensions to non-negative matrix factorisation for audio signal modelling." *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008.

References-3

- [Boldt&Ellis 2009] Boldt, Jesper B., and Daniel PW Ellis. "A simple correlation-based model of intelligibility for nonlinear speech enhancement and separation." *EUSIPCO 2009: 17th European Signal Processing Conference, August 24-28, 2009, Glasgow, Scotland*. European Association for Signal, Speech, and Image Processing, 2009.
- [Mohamed&Dahl&Hinton 2009] Abdel-rahman Mohamed, George E. Dahl, Geoffrey E. Hinton. "Deep Belief Networks for Phone Recognition." *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.
- [Hershey&Rennie&Olsen&Kristjansson 2010] Hershey, J. R., Rennie, S. J., Olsen, P. A., & Kristjansson, T. T. "Super-human multi-talker speech recognition: A graphical modeling approach." *Computer Speech & Language*, *24*(1), 45-66, 2010.
- [Grais&Erdogan 2011] Grais, Emad M., and Hakan Erdogan. "Single channel speech music separation using nonnegative matrix factorization and spectral masks." *Digital Signal Processing (DSP), 2011 17th International Conference on*. IEEE, 2011.
- [Yu&Deng 2011] Yu, Dong, and Li Deng. "Deep learning and its applications to signal and information processing [exploratory dsp]." *Signal Processing Magazine, IEEE* 28.1 (2011): 145-154.
- [Taal&Hendriks&Heusdens&Jensen 2011] Taal, Cees H., et al. "An algorithm for intelligibility prediction of time–frequency weighted noisy speech." *Audio, Speech, and Language Processing, IEEE Transactions on* 19.7 (2011): 2125-2136.
- [Wang&Wang 2013] Wang, Yuxuan, and DeLiang Wang. "Towards scaling up classification-based speech separation." *Audio, Speech, and Language Processing, IEEE Transactions on* 21.7 (2013): 1381-1390.
- [Huang&Kim&Johnson&Smaragdis 2014] Po-Sen Huang; Minje Kim; Hasegawa-Johnson, M.; Smaragdis, P., "Deep learning for monaural speech separation," *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on* , vol., no., pp.1562,1566, 4-9 May 2014.

References-4

- [Grais&Sen&Erdogan 2014] Grais, E.M.; Sen, M.U.; Erdogan, H., "Deep neural networks for single channel source separation," *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on* , vol., no., pp.3734,3738, 4-9 May 2014.
- [Weninger&Hershey&LeRoux&Schuller 2014] Felix Weninger, Jonathan Le Roux, John R. Hershey, Björn Schuller, "Discriminatively Trained Recurrent Neural Networks for Single-Channel Speech Separation," to appear in Proc. IEEE GlobalSIP 2014 Symposium on Machine Learning Applications in Speech Processing, Dec 2014.
- [Weninger et.al. 2014] Felix Weninger, Shinji Watanabe, Jonathan Le Roux, John R. Hershey, Yuuki Tachioka, Jürgen Geiger, Björn Schuller, Gerhard Rigoll: *"The MERL/MELCO/TUM system for the REVERB Challenge using Deep Recurrent Neural Network Feature Enhancement"*, Proc. REVERB Workshop held in conjunction with ICASSP 2014 and HSCMA 2014, IEEE, Florence, Italy, 10.05.2013
- [Weninger et.al. 2014] Felix Weninger, Shinji Watanabe, Jonathan Le Roux, John R. Hershey, Yuuki Tachioka, Jürgen Geiger, Björn Schuller, Gerhard Rigoll: *"The MERL/MELCO/TUM system for the REVERB Challenge using Deep Recurrent Neural Network Feature Enhancement"*, Proc. REVERB Workshop held in conjunction with ICASSP 2014 and HSCMA 2014, IEEE, Florence, Italy, 10.05.2013
- [Weninger&LeRoux&Hershey&Watanabe 2014] Felix Weninger, Jonathan Le Roux, John R. Hershey, Shinji Watanabe, "Discriminative NMF and its application to single-channel source separation," *Proc. ISCA Interspeech 2014 (Interspeech 2014)*, Sep. 2014.
- [Xu&Du&Dai&Lee 2014] Yong Xu; Jun Du; Li-Rong Dai; Chin-Hui Lee, "An Experimental Study on Speech Enhancement Based on Deep Neural Networks," *Signal Processing Letters, IEEE* , vol.21, no.1, pp.65,68, Jan. 2014.
- [Wang&Narayanan&Wang 2014] Yuxuan Wang; Narayanan, A.; DeLiang Wang, "On Training Targets for Supervised Speech Separation," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on* , vol.22, no.12, pp.1849,1858, Dec. 2014.

References-5

- [Wang&Wang 2015] Yuxuan Wang; DeLiang Wang, “A deep neural network for time-domain signal reconstruction," *ICASSP 2015*.
- [Souden&Benesty&Affes 2010] M Souden, J Benesty, S Affes, “On optimal frequency domain multichannel linear filtering for noise reduction,” *IEEE TASLP*, vol 18, no 2, 2010.
- [Williamson&Wang 2016] Williamson D.S., Wang Y., and Wang D.L. (2016): Complex ratio masking for monaural speech separation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, pp. 483-492.
- [Wang&Wang 2016] Wang Z.-Q. and Wang D.L. (2016): A joint training framework for robust automatic speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, pp. 796-806.
- [Gao&Du&Dai&Lee 2015] T. Gao, J. Du, L. R. Dai and C. H. Lee, "Joint training of front-end and back-end deep neural networks for robust speech recognition," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, QLD, 2015, pp. 4375-4379.

Deep learning approaches to multichannel separation

Speaker: Emmanuel Vincent

Time-domain representation

Reminder: in the general case with I microphones

$$\mathbf{x}(t) = \sum_{j=1}^{J} \mathbf{c}_j(t)$$

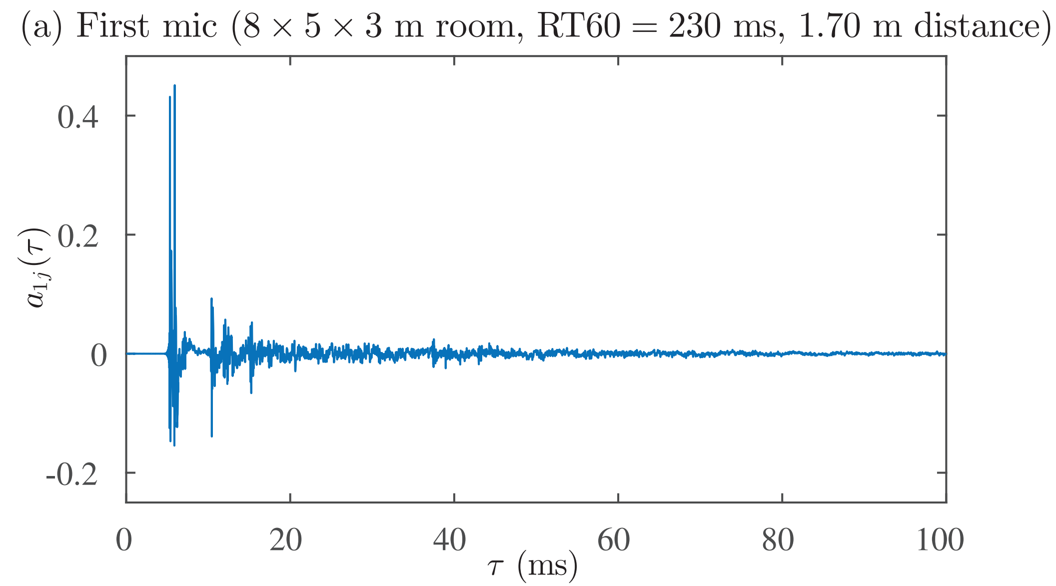
$\mathbf{x}(t)$: $I \times 1$ mixture signal
$\mathbf{c}_j(t)$: $I \times 1$ spatial image of source j
t: discrete time

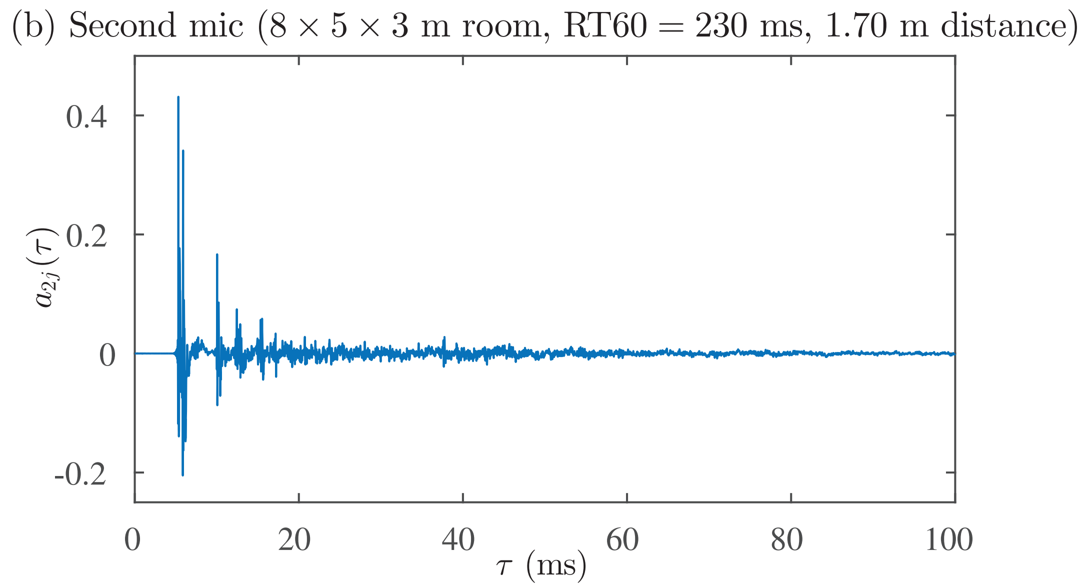
In the case of a point source:

$$\mathbf{c}_j(t) = \sum_{\tau=0}^{\infty} \mathbf{a}_j(\tau)\, s_j(t-\tau)$$

$\mathbf{a}_j(\tau)$: $I \times 1$ vector of acoustic impulse responses
$s_j(t)$: single-channel source signal

Acoustic impulse responses





Narrowband approximation

Assuming low reverberation:

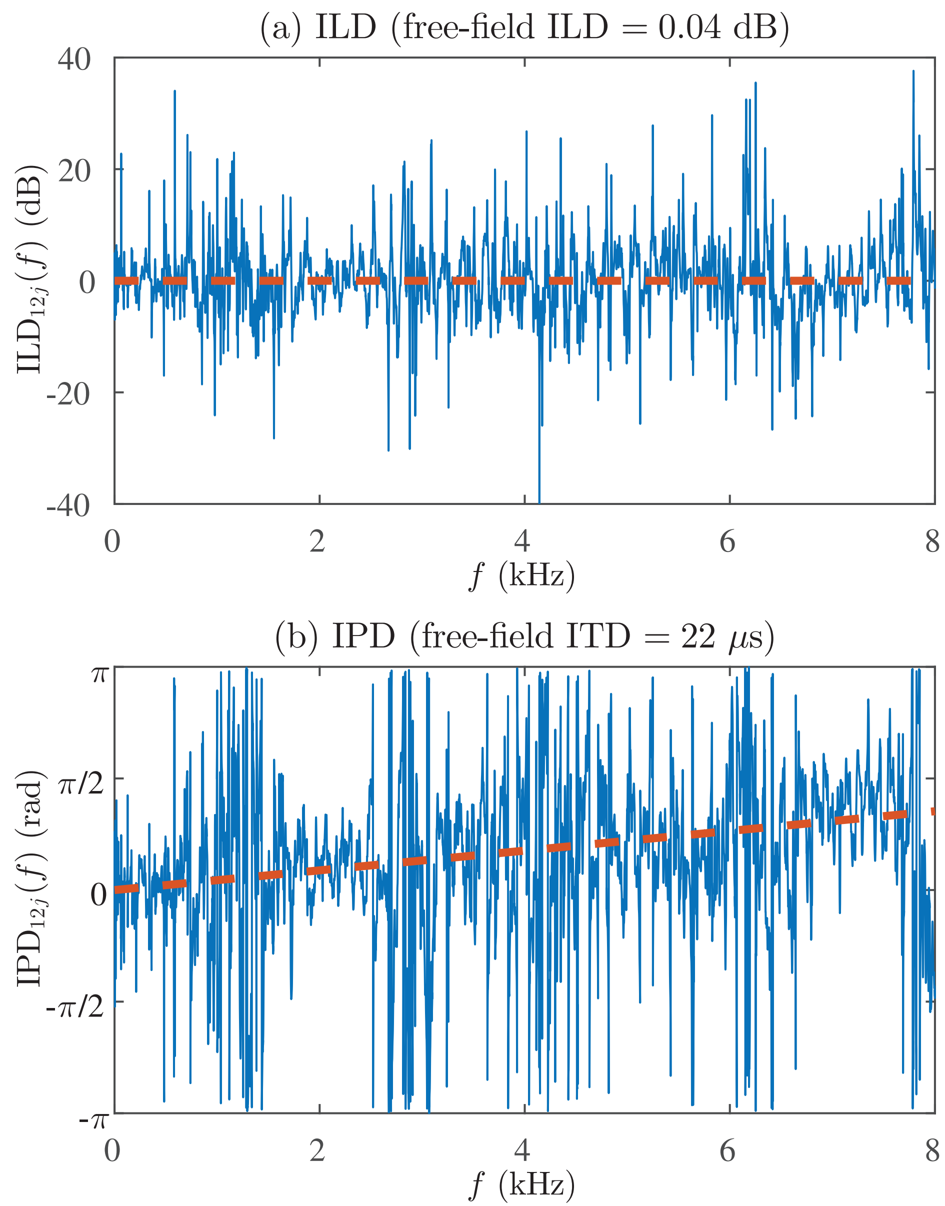
$$\mathbf{c}_j(n,f) \approx \mathbf{a}_j(f)\, s_j(n,f)$$

$\mathbf{c}_j(n,f)$: $I \times 1$ STFT of $\mathbf{c}_j(t)$
$\mathbf{a}_j(f)$: $I \times 1$ vector of acoustic transfer functions
$s_j(n,f)$: STFT of $s_j(t)$

Magnitude and phase of $\mathbf{a}_j(f)$ and $s_j(n,f)$ difficult to disambiguate $\Rightarrow$ model the relative transfer functions between mics instead:

- level difference (ILD): $\mathrm{ILD}_{ii'j}(f) = |a_{ij}(f)|/|a_{i'j}(f)|$
- phase difference (IPD): $\mathrm{IPD}_{ii'j}(f) = \angle a_{ij}(f) - \angle a_{i'j}(f) \pmod{2\pi}$
- time difference (ITD): $\mathrm{ITD}_{ii'j}(f) = \mathrm{IPD}_{ii'j}(f)/2\pi f \pmod{1/f}$

Interchannel level and phase differences



Spatial covariance matrix

With higher reverberation, sound comes from many directions at once.

Zero-mean multichannel Gaussian model:

$$\mathbf{c}_j(n,f) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{c}_j}(f))$$
$$\sim \mathcal{N}(\mathbf{0}, \sigma^2_{s_j}(n,f)\mathbf{R}_j(f))$$

$\mathbf{\Sigma}_{\mathbf{c}_j}(f)$: $I \times I$ source covariance matrix

$\sigma^2_{s_j}(n,f)$: short-term power spectrum

$\mathbf{R}_j(f)$: $I \times I$ spatial covariance matrix

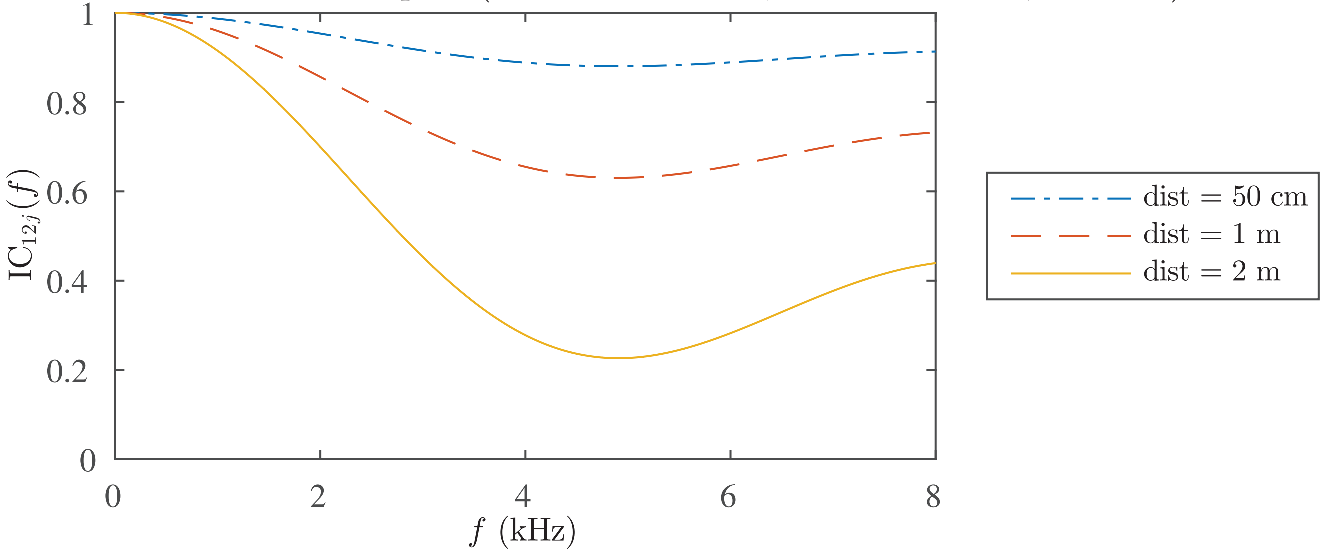
$\mathbf{R}_j(f) = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix}$ can be parameterized in terms of

- ILD $\sqrt{r_{11}/r_{22}}$,
- IPD $\angle r_{12}$,
- coherence (IC) $|r_{12}|/\sqrt{r_{11}r_{22}}$

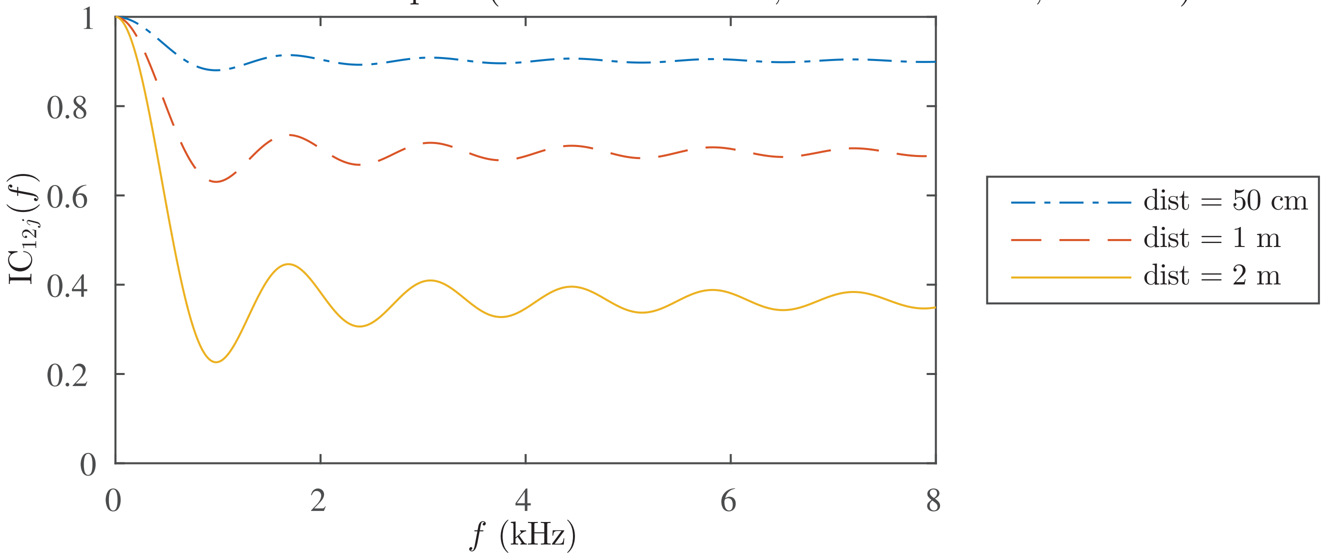
IC encodes the diffuseness of the sound field.

Interchannel coherence

(a) Theoretical IC for mics 5 cm apart ($8 \times 5 \times 3$ m room, RT60 = 230 ms, DOA $0°$)



(a) Theoretical IC for mics 25 cm apart ($8 \times 5 \times 3$ m room, RT60 = 230 ms, DOA $0°$)



Single-channel separation using spatial features

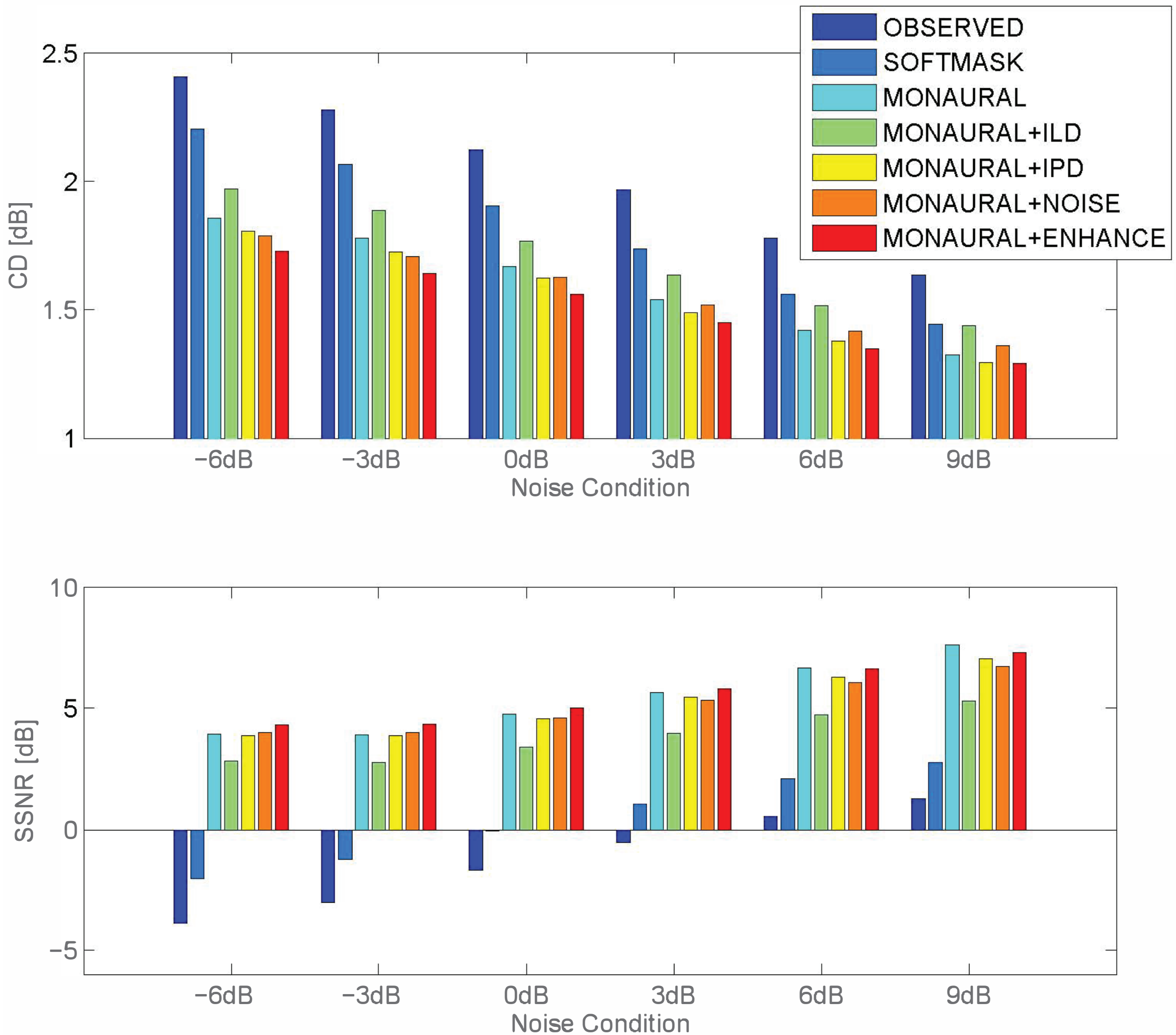
- Append spatial features to the inputs
  - ILD,
  - cos(IPD),
  - full interchannel cross-correlation,
  - speech magnitude spectrum after beamforming,
  - speech/noise magnitude spectrum after multichannel GMM...
- Train a DNN to compute a single-channel mask $m_j(n, f)$
  - exploit the fact that the features of the mixture are similar to those of the predominant source,
  - ensure the training set covers all possible angles,
  - and/or shift the IPD according to source localization
- Apply it to one channel or as a post-filter after conventional beamforming, e.g., delay-and-sum (DS)

$$\widehat{c}_{ij}(n, f) = m_j(n, f) x_i(n, f)$$
$$\widehat{s}_j(n, f) = m_j(n, f) x_{\mathrm{BF}}(n, f)$$

$\widehat{c}_{ij}(n, f)$: spatial image estimate
$\widehat{s}_j(n, f)$: source estimate
$x_{\mathrm{BF}}(n, f)$: beamformer output

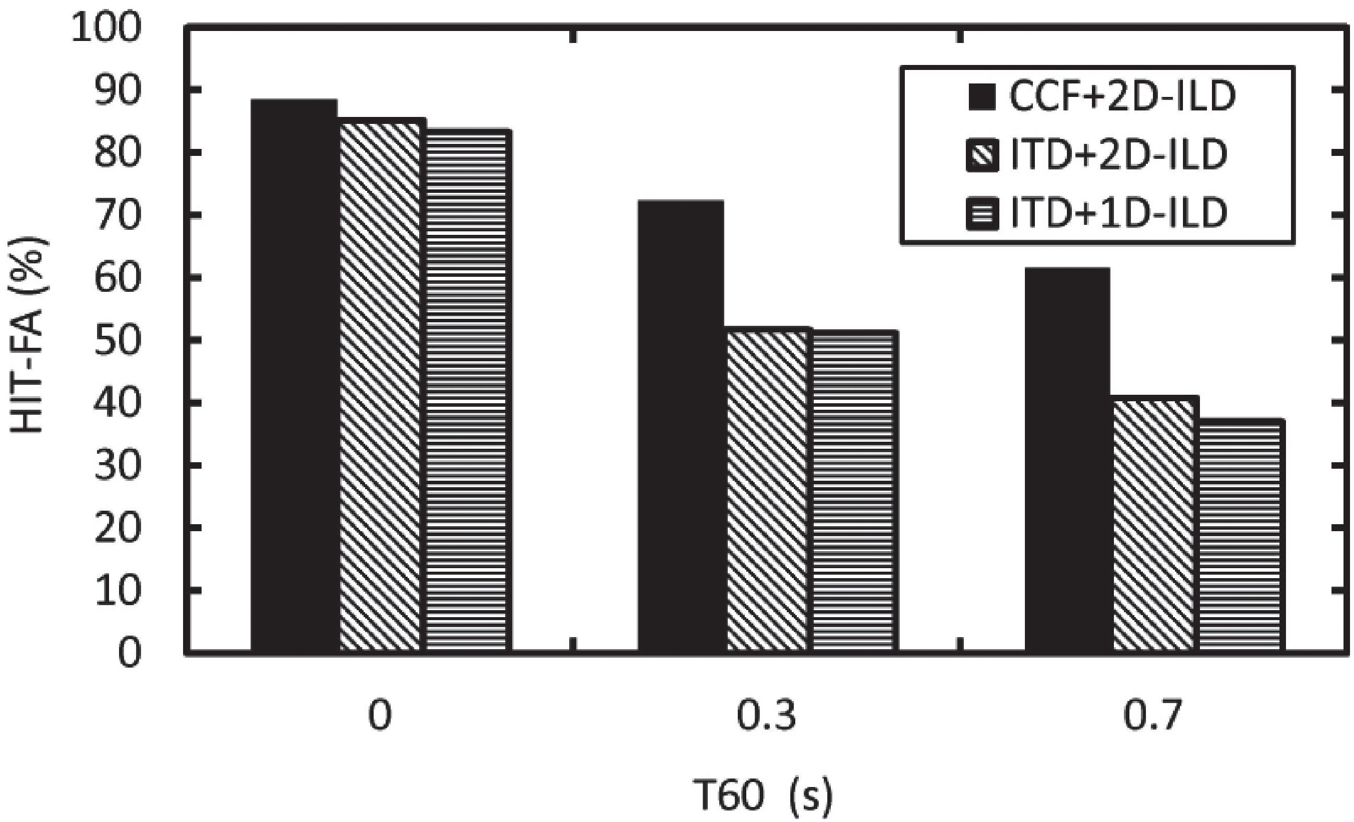
Results (single spatial feature)

Araki et al. – CHiME-1, cepstral distorsion & segmental SNR



Results (multiple spatial features)

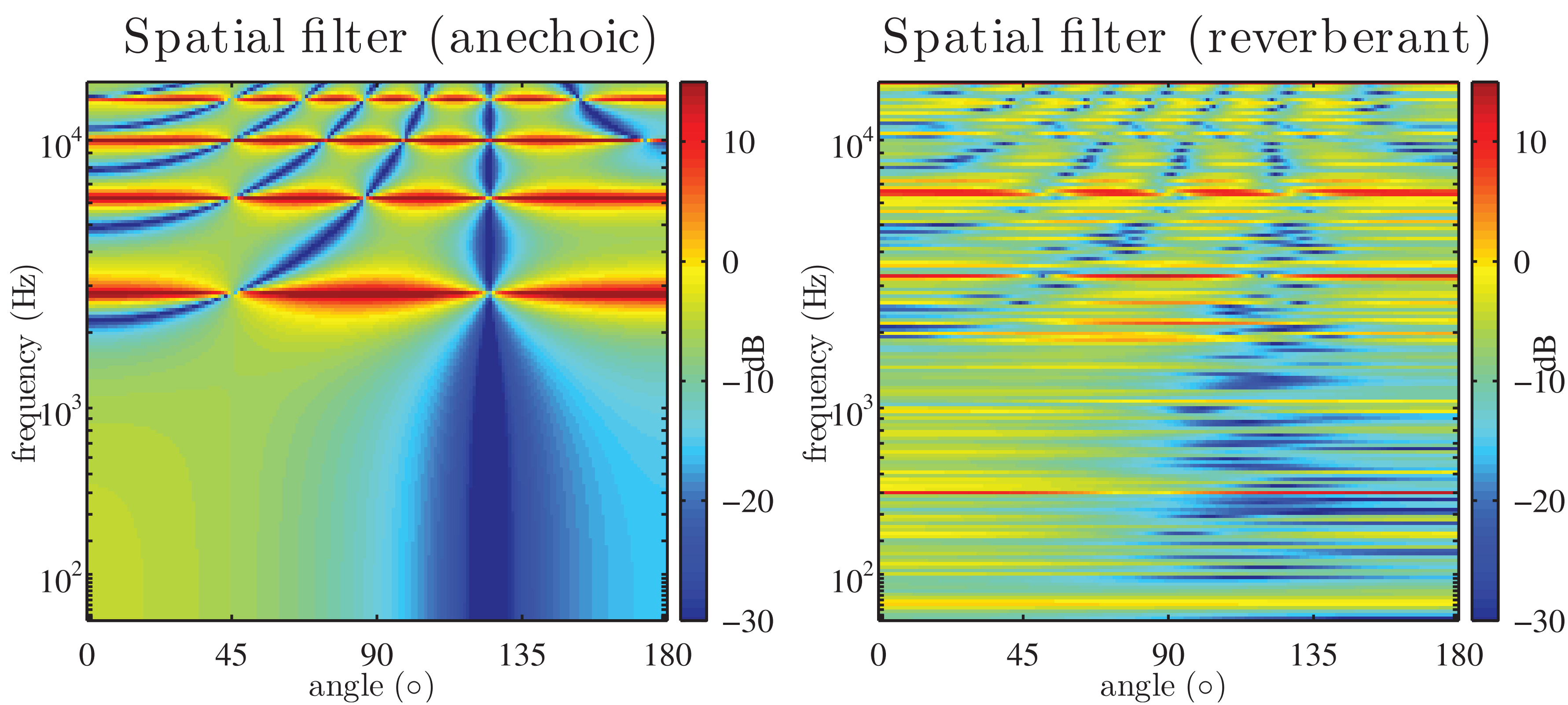
Jiang et al. – Mixtures of 2 TIMIT sentences at 0 dB SNR, hit - false alarm rate



Multichannel vs. single-channel separation

Multichannel adaptive filtering (aka beamforming)

- can cancel up to $I - 1$ coherent sources
- distorts speech less than single-channel masking or post-filtering



Model-based multichannel separation

Multichannel filter:

$$\widehat{\mathbf{c}}_j(n,f) = \mathbf{W}_j(n,f)^H \mathbf{x}(n,f)$$
$$\widehat{s}_j(n,f) = \mathbf{w}_j(n,f)^H \mathbf{x}(n,f)$$

$\mathbf{W}_j(n,f)$: $I \times I$ matrix
$\mathbf{w}_j(n,f)$: $I \times 1$ vector

Model-based approach:

- use a DNN to estimate the source statistics $\mathbf{\Sigma}_{\mathbf{c}_j}(n,f)$,
- derive $\mathbf{W}_j(n,f)$ or $\mathbf{w}_j(n,f)$ according to a filter design criterion.

MSE-based filter design: narrowband case

Under the narrowband approximation, with $\mathbf{c}_{\neq j}(n,f) = \sum_{j' \neq j} \mathbf{c}_{j'}(n,f)$:

$$\widehat{s}_j(n,f) - s_j(n,f) = \underbrace{[\mathbf{w}_j^H(n,f)\mathbf{a}_j(f) - 1]s_j(n,f)}_{\text{speech distorsion}} + \underbrace{\mathbf{w}_j^H(n,f)\mathbf{c}_{\neq j}(n,f)}_{\text{residual noise}}.$$

Weighted mean square error (MSE) criterion:

$$\min_{\mathbf{w}_j(n,f)} = |\mathbf{w}_j^H(n,f)\mathbf{a}_j(f) - 1|^2 \sigma_{s_j}^2(n,f) + \mu \mathbf{w}_j^H(n,f)\mathbf{\Sigma}_{\mathbf{c}_{\neq j}}(n,f)\mathbf{w}_j(n,f).$$

$$\Rightarrow \mathbf{w}_j(n,f) = \frac{\sigma_{s_j}^2(n,f)\mathbf{\Sigma}_{\mathbf{c}_{\neq j}}^{-1}(n,f)\mathbf{a}_j(f)}{\mu + \sigma_{s_j}^2(n,f)\mathbf{a}_j^H(f)\mathbf{\Sigma}_{\mathbf{c}_{\neq j}}^{-1}(n,f)\mathbf{a}_j(f)}$$

Special cases:

- $\mu \to 0$: minimum variance distorsionless response (MVDR)
- $\mu = 1$: multichannel Wiener filter (MWF) $= \mathbf{\Sigma}_{\mathbf{x}}^{-1}(n,f)\sigma_{s_j}^2(n,f)\mathbf{a}_j(f)$

Note:

- all spatial filters are equal, only the spectral gain changes
- μ can be interpreted as an oversubtraction factor

MSE-based filter design: general case

In the general case:

$$\widehat{\mathbf{c}}_j(n,f) - \mathbf{c}_j(n,f) = \underbrace{[\mathbf{W}_j^H(n,f) - \mathbf{I}]\mathbf{c}_j(n,f)}_{\text{speech distorsion}} + \underbrace{\mathbf{W}_j^H(n,f)\mathbf{c}_{\neq j}(n,f)}_{\text{residual noise}}.$$

Weighted MSE criterion:

$$\min_{\mathbf{W}_j(n,f)} = [\mathbf{w}_j^H(n,f) - \mathbf{I}]\mathbf{\Sigma}_{\mathbf{c}_j}(n,f)[\mathbf{w}_j(n,f) - \mathbf{I}]$$
$$+ \mu \mathbf{W}_j^H(n,f)\mathbf{\Sigma}_{\mathbf{c}_{\neq j}}(n,f)\mathbf{W}_j(n,f).$$

$$\Rightarrow \mathbf{W}_j(n,f) = [\mathbf{\Sigma}_{\mathbf{c}_j}(n,f) + \mu\mathbf{\Sigma}_{\mathbf{c}_{\neq j}}(n,f)]^{-1}\mathbf{\Sigma}_{\mathbf{c}_j}(n,f).$$

Special cases:

- $\mu \to 0$: distorsionless noise reduction not feasible anymore
- $\mu = 1$: multichannel Wiener filter (MWF) $= \mathbf{\Sigma}_{\mathbf{x}}^{-1}(n,f)\mathbf{\Sigma}_{\mathbf{c}_j}(n,f)$

SNR-based filter design

Maximum SNR (MSNR) criterion:

$$\max_{\mathbf{w}_j(n,f)} = \frac{|\mathbf{w}_j^H(n,f)\mathbf{a}_j(f)|^2\sigma_{s_j}^2(n,f)}{\mathbf{w}_j^H(n,f)\boldsymbol{\Sigma}_{\mathbf{c}_{\neq j}}(n,f)\mathbf{w}_j(n,f)} = \frac{\mathbf{w}_j^H(n,f)\boldsymbol{\Sigma}_{\mathbf{c}_j}(n,f)\mathbf{w}_j(n,f)}{\mathbf{w}_j^H(n,f)\boldsymbol{\Sigma}_{\mathbf{c}_{\neq j}}(n,f)\mathbf{w}_j(n,f)}.$$

Solution:

- in the narrowband case, $\mathbf{w}_j(n,f) \propto \boldsymbol{\Sigma}_{\mathbf{c}_{\neq j}}^{-1}(n,f)\mathbf{a}_j(f)$,
- in general, $\mathbf{w}_j(n,f) \propto$ principal eigenvector of $\boldsymbol{\Sigma}_{\mathbf{c}_{\neq j}}^{-1}(n,f)\boldsymbol{\Sigma}_{\mathbf{c}_j}(n,f)$ or $\boldsymbol{\Sigma}_{\mathbf{x}}^{-1}(n,f)\boldsymbol{\Sigma}_{\mathbf{c}_j}(n,f)$, aka generalized eigenvalue (GEV) beamformer.

Gain fixing post-filter $w_{\mathrm{BAN}j}(n,f) = \frac{(\mathbf{w}_j^H(n,f)\boldsymbol{\Sigma}_{\mathbf{c}_j}(n,f)\boldsymbol{\Sigma}_{\mathbf{c}_j}(n,f)\mathbf{w}_j(n,f))^{1/2}}{\mathbf{w}_j^H(n,f)\boldsymbol{\Sigma}_{\mathbf{c}_j}(n,f)\mathbf{w}_j(n,f)}$

- makes MSNR equivalent to MDVR in the narrowband case
- but more robust than MVDR when narrowband approx. doesn't hold

Mask-based estimation of source statistics

Speech presence probability (SPP) based estimation of source statistics:

- use a DNN to estimate a single-channel mask $m_j(n,f)$,
- estimate the source statistics recursively as

$$\boldsymbol{\Sigma}_{\mathbf{c}_j}(n,f) = \lambda \boldsymbol{\Sigma}_{\mathbf{c}_j}(n-1,f) + (1-\lambda) m_j(n,f) \mathbf{x}(n,f)\mathbf{x}^H(n,f)$$

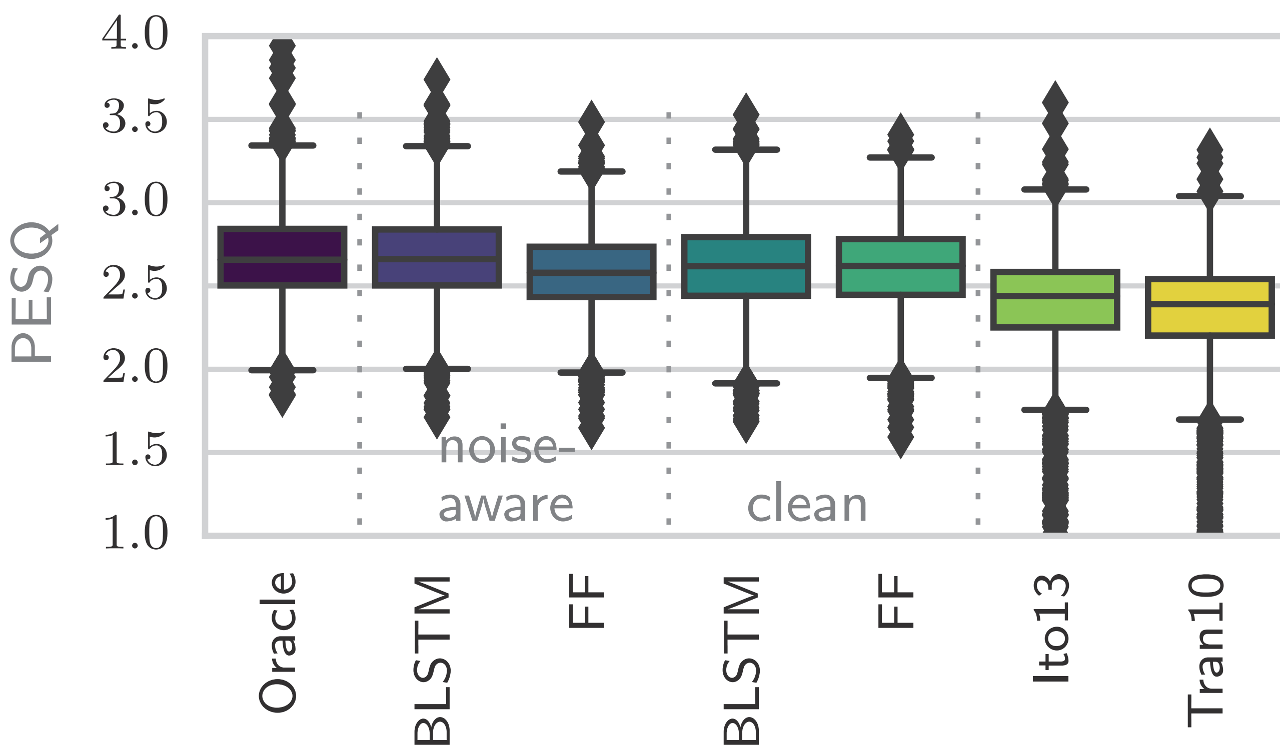
$$\boldsymbol{\Sigma}_{\mathbf{c}_{\neq j}}(n,f) = \lambda \boldsymbol{\Sigma}_{\mathbf{c}_{\neq j}}(n-1,f) + (1-\lambda)[1 - m_j(n,f)] \mathbf{x}(n,f)\mathbf{x}^H(n,f)$$

  with forgetting factor λ ($\lambda \to 1$: time-invariant filter).

Note: assumes the source statistics vary slowly over time.

Results

Heymann et al. – CHiME-3 simulated development set, perceptual speech quality
noise-aware: training on speech+noise mixtures
clean: training on clean speech (target mask = TF bins totaling 99% power)
Ito13, Tran10: spatial clutering techniques



Iterative EM-based estimation of source statistics

Classical alternative approach:

- factor $\boldsymbol{\Sigma}_{\mathbf{c}_j}(n,f)$ into spectral (quick) and spatial (slow) parameters

$$\mathbf{c}_j(n,f) \sim \mathcal{N}(\mathbf{0}, v_j(n,f)\mathbf{R}_j(f))$$

$v_j(n,f)$: short-term power spectrum
$\mathbf{R}_j(f)$: spatial covariance matrix

- alternately reestimate $v_j(n,f)$ and $\mathbf{R}_j(f)$ in the maximum likelihood (ML) sense using an expectation maximization (EM) algorithm

$$\max_{\{v_j(n,f),\mathbf{R}_j(f)\}} \sum_{nf} \log \mathcal{N}(\mathbf{x}(n,f)|\mathbf{0}, \textstyle\sum_j v_j(n,f)\mathbf{R}_j(f))$$

Tweak: use a DNN to reestimate $v_j(n,f)$ instead.

Iterative EM-based estimation of source statistics

- Initialization: $v_j(n,f) \leftarrow f_{\mathcal{W}}[|\mathbf{w}_{\mathrm{DS}}^H \mathbf{x}(n,f)|]$, $\mathbf{R}_j(f) \leftarrow \mathbf{I}$
- E-step: estimate the posterior statistics of the sources

$$\mathbf{W}_j(n,f) = \left[\sum_{j'} v_{j'}(n,f)\mathbf{R}_{j'}(f)\right]^{-1} v_j(n,f)\mathbf{R}_j(f) \quad \text{(MWF)}$$

$$\widehat{\mathbf{c}}_j(n,f) = \mathbf{W}_j^H(n,f)\mathbf{x}(n,f)$$

$$\widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_j}(n,f) = \widehat{\mathbf{c}}_j(n,f)\widehat{\mathbf{c}}_j^H(n,f) + [\mathbf{I} - \mathbf{W}_j^H(n,f)]v_j(n,f)\mathbf{R}_j(f)$$

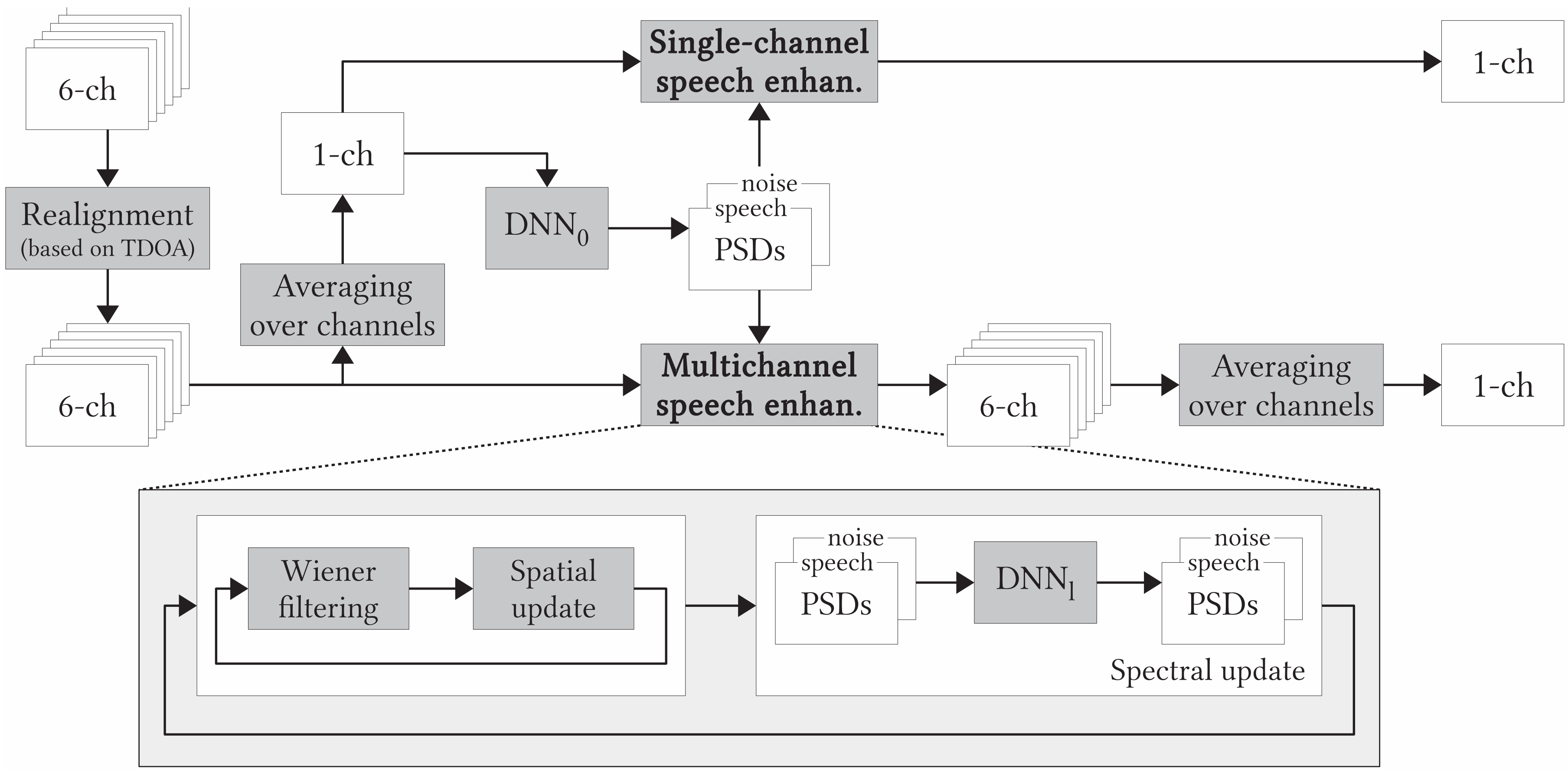
- M-step: update the parameters

$$\mathbf{R}_j(f) \leftarrow \frac{1}{N}\sum_n \frac{\widehat{\mathbf{R}}_{\mathbf{c}_j}(n,f)}{v_j(n,f)}$$

$$\xi_j(n,f) \leftarrow \frac{1}{I}\operatorname{tr}(\mathbf{R}_j(f)^{-1}\widehat{\boldsymbol{\Sigma}}_{\mathbf{c}_j}(n,f)) \quad \text{(power spectrum estimate)}$$

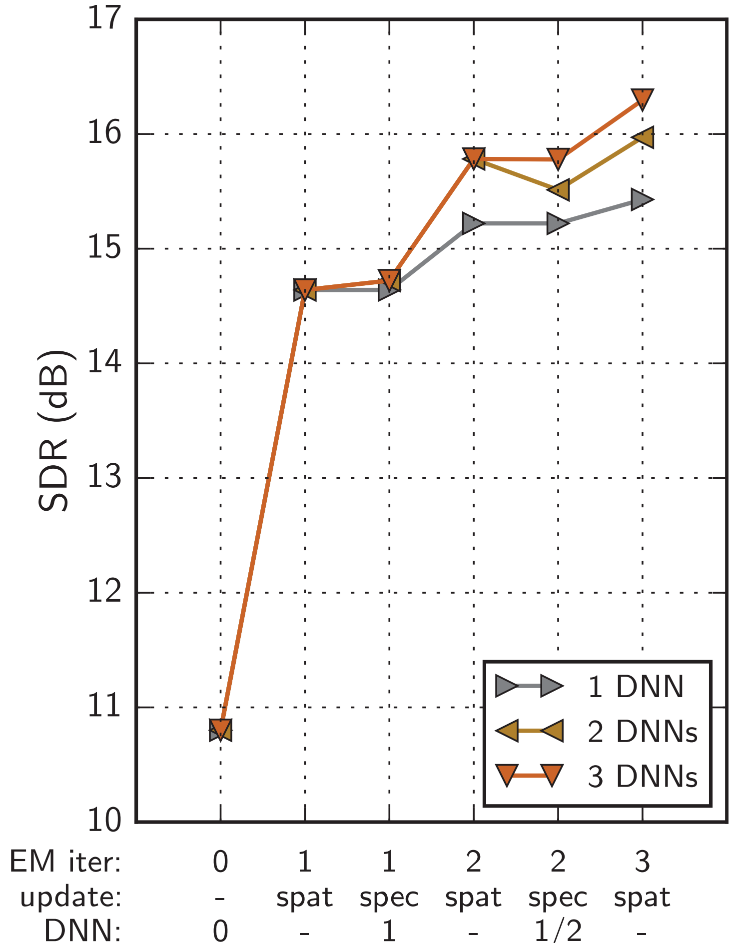
$$v_j(n,f) \leftarrow f_{\mathcal{W}}[\xi_j(n,f)] \quad \text{(improve estimate by DNN)}$$

Iterative EM-based estimation of source statistics



Results

Nugraha et al. – CHiME-3 simulated test set, signal-to-distorsion ratio



Results

Nugraha et al. – CHiME-3 real test set, DNN-based ASR backend

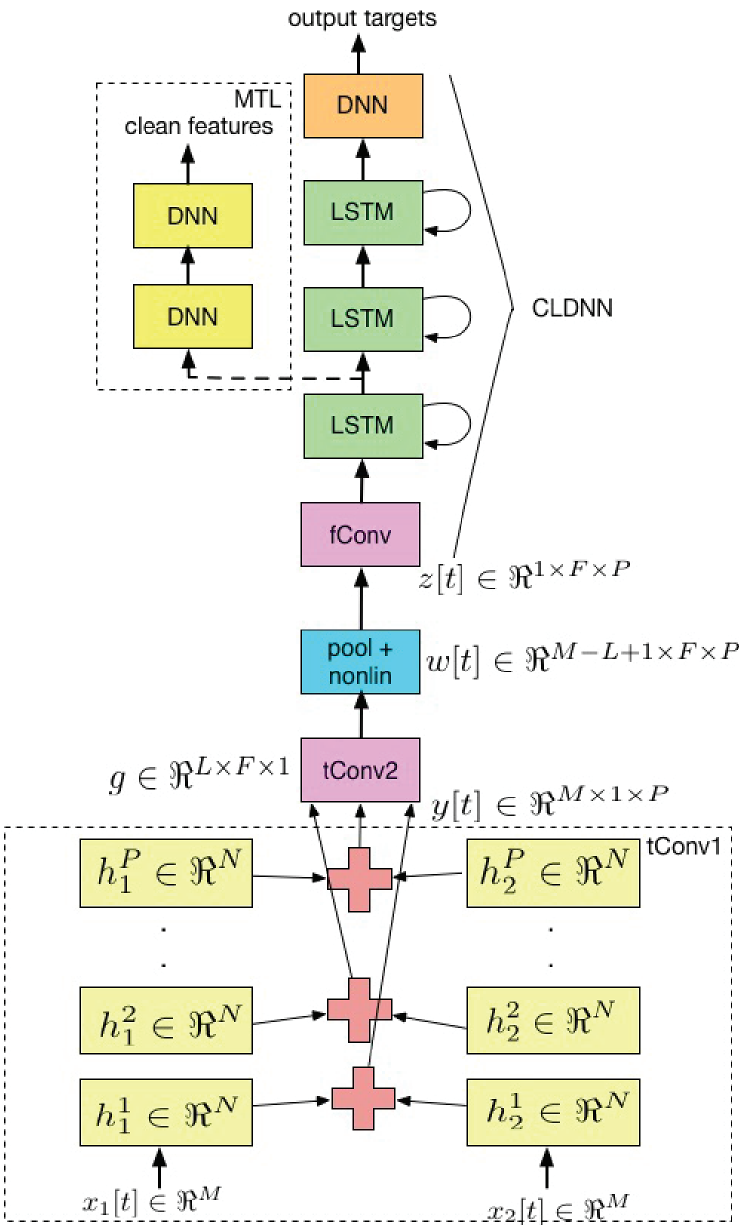
Noisy	WER=19.28%
DS beamforming	WER=13.70%
Multichannel NMF	WER=13.41%
Initialization	WER=15.18%
Update $\mathbf{R}_j$ (iter 1)	WER=11.46%
Update v_j (iter 1)	WER=11.46%
Update $\mathbf{R}_j$ (iter 2)	WER=10.79%
Update v_j (iter 2)	WER=11.12%
Update $\mathbf{R}_j$ (iter 3)	WER=10.14%

Single-step multichannel separation

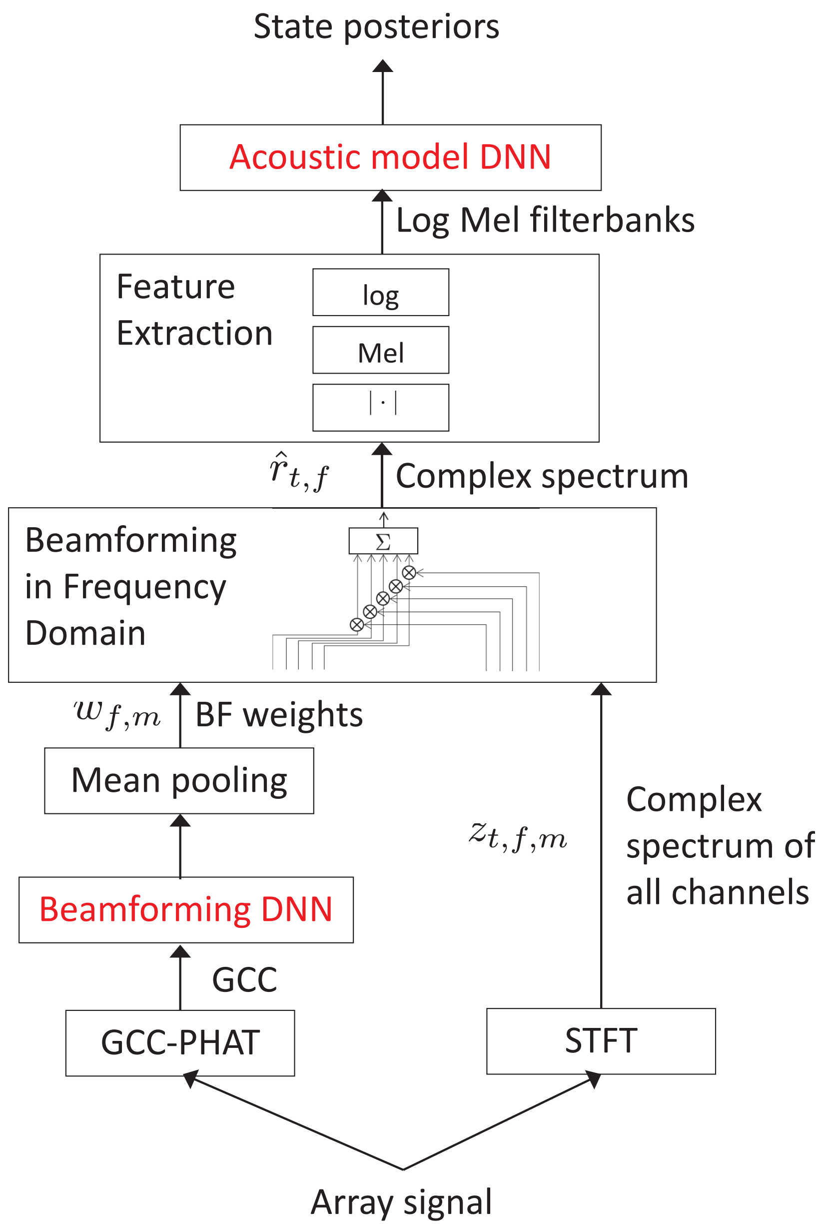
Instead of using a DNN to compute the source statistics, use it to

- compute the beamformer weights
- . . . or compute the beamformed signal directly!

Fixed beamforming layer



Adaptive beamforming network



Results

Xiao et al. – AMI, DNN-based ASR backend

Method	WER (%)
Single distant mic	53.8
DS beamforming	47.9
Beamforming network	44.7
Headset	25.5

References

Multichannel separation

S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multi-microphone speech enhancement and source separation", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, to appear.

Single-channel separation using spatial features

Y. Jiang, D. L. Wang, R. S. Liu, and Z. M. Feng, "Binaural classification for reverberant speech segregation using deep neural networks", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):2112–2121, 2014.

P. Pertilä and J. Nikunen, "Distant speech separation using predicted time-frequency masks from spatial features", *Speech Communication*, 68:97–106, 2015.

S. Araki, T. Hayashi, M. Delcroix, M. Fujimoto, K. Takeda, and T. Nakatani, "Exploring multi-channel features for denoising-autoencoder based speech enhancement", in *Proc. ICASSP*, pp. 116–120, 2015.

References

Model-based multichannel separation

J. Heymann, L. Drude, and R. Haeb-Umbach, “Neural network based spectral mask estimation for acoustic beamforming”, in *Proc. ICASSP*, pp. 196–200, 2016.

A. A. Nugraha, A. Liutkus, and E. Vincent, “Multichannel audio source separation with deep neural networks”, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(9):1652–1664, 2016.

Single-step multichannel separation

P. Swietojanski, A. Ghoshal, and S. Renals, “Convolutional neural networks for distant speech recognition”, *IEEE Signal Processing Letters*, 21(9):1120–1124, 2014.

T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, and M. Bacchiani, “Factored spatial and spectral multichannel raw waveform CLDNNs”, in *Proc. ICASSP*, pp. 5075–5079, 2016.

X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. R. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, “Deep beamforming networks for multi-channel speech recognition”, in *Proc. ICASSP*, pp. 5745–5749, 2016.

New directions in deep-learning approaches

Speaker: Jonathan Le Roux

Interspeech 2016 Tutorial

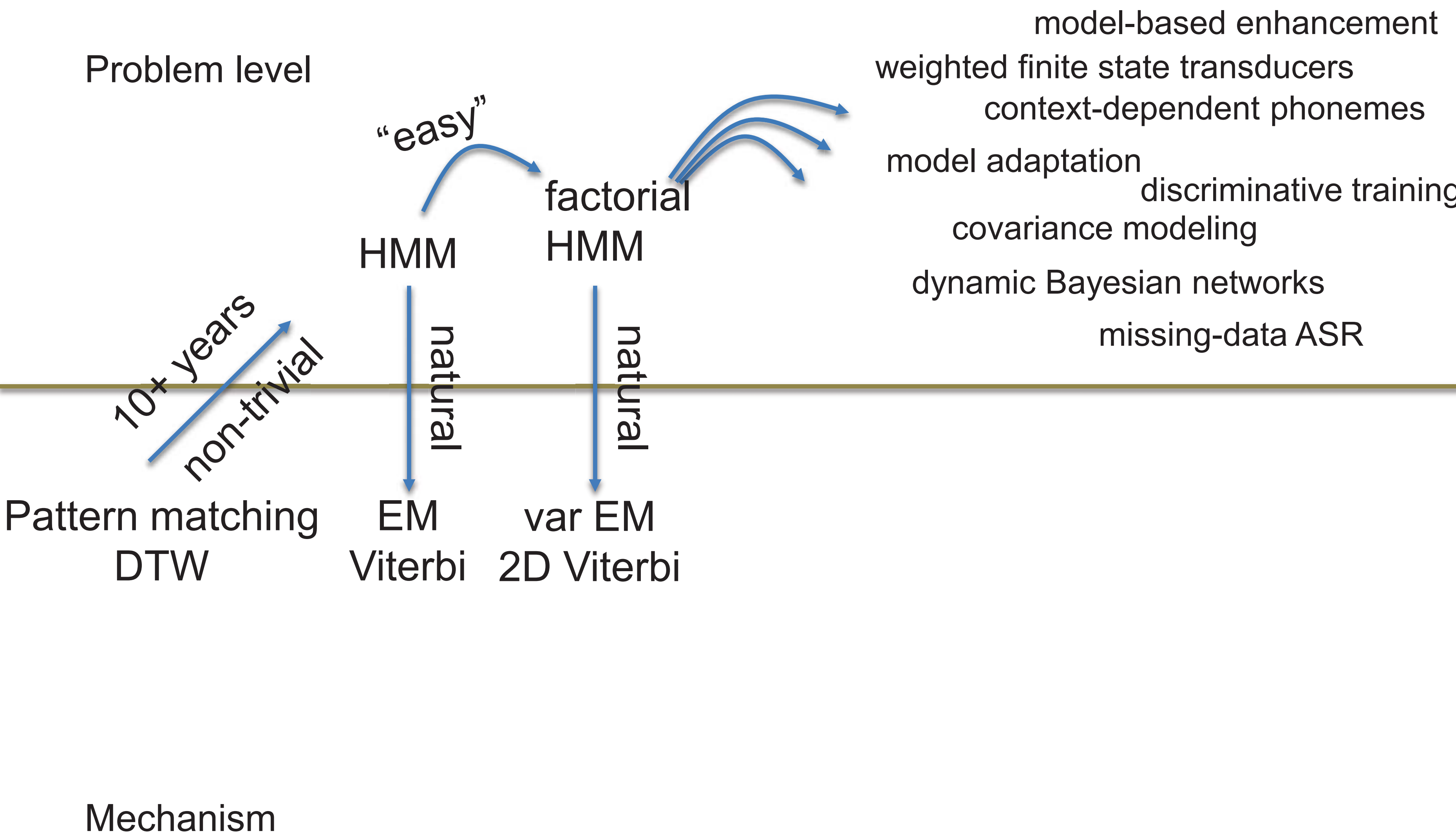
Data-driven approaches to speech enhancement and separation

Model-based vs Deep learning

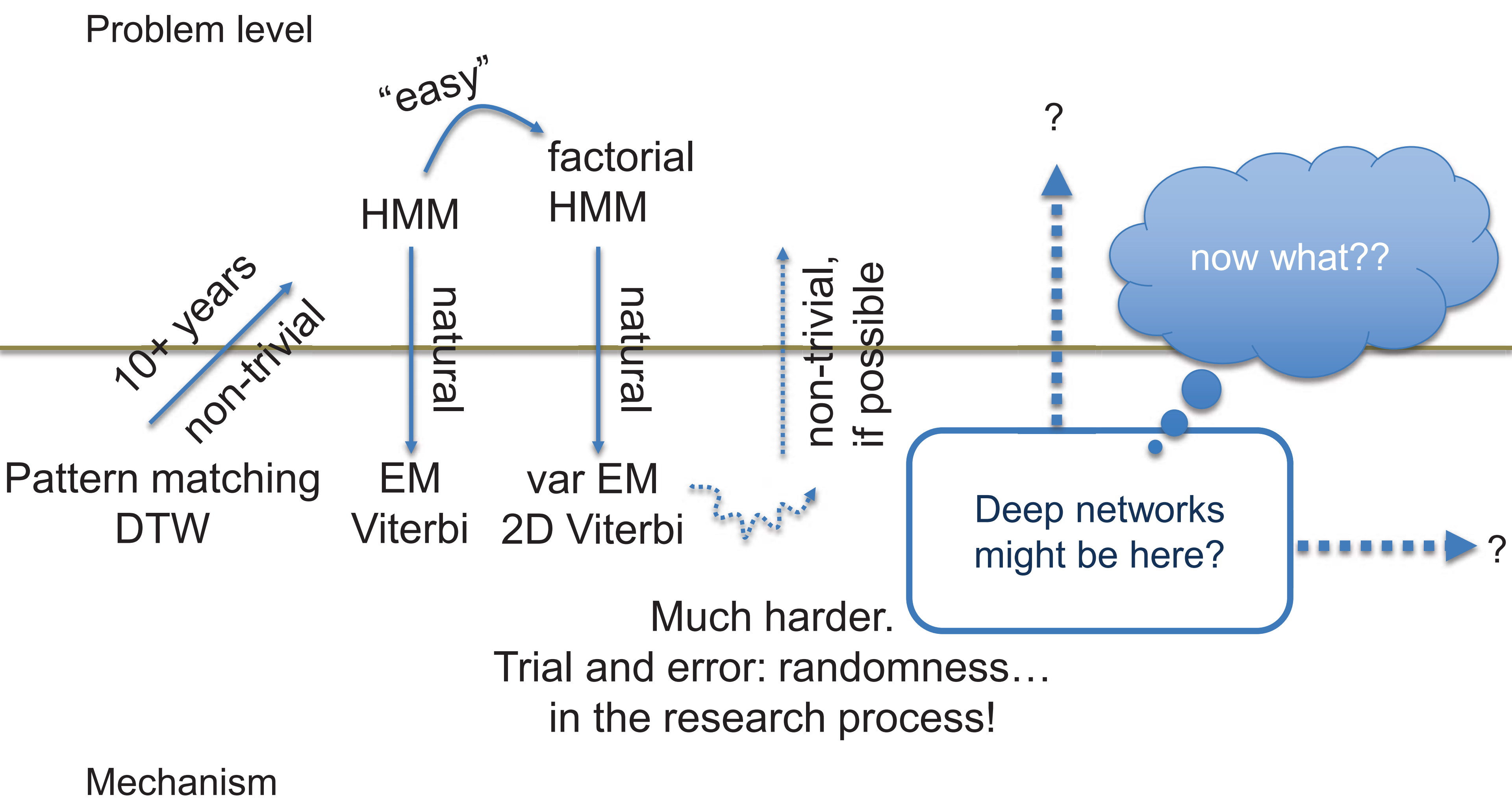
- Both very successful machine learning approaches
- They are polar opposites in some ways
- We want to have the advantages of both

	Generative model-based	Conventional DNN	?
Use problem domain knowledge	☺	**not so much**	☺
Insight for improvement	☺	**difficult**	☺
Easy Inference	**optimizing, please wait…**	☺	☺
Easy discriminative training	**bi-level optimization?**	☺	☺
Invariance to unseen noise and reverberation, array geometry, mic ordering	☺	**tough, but maybe…**	☺

Problem level vs. mechanism level

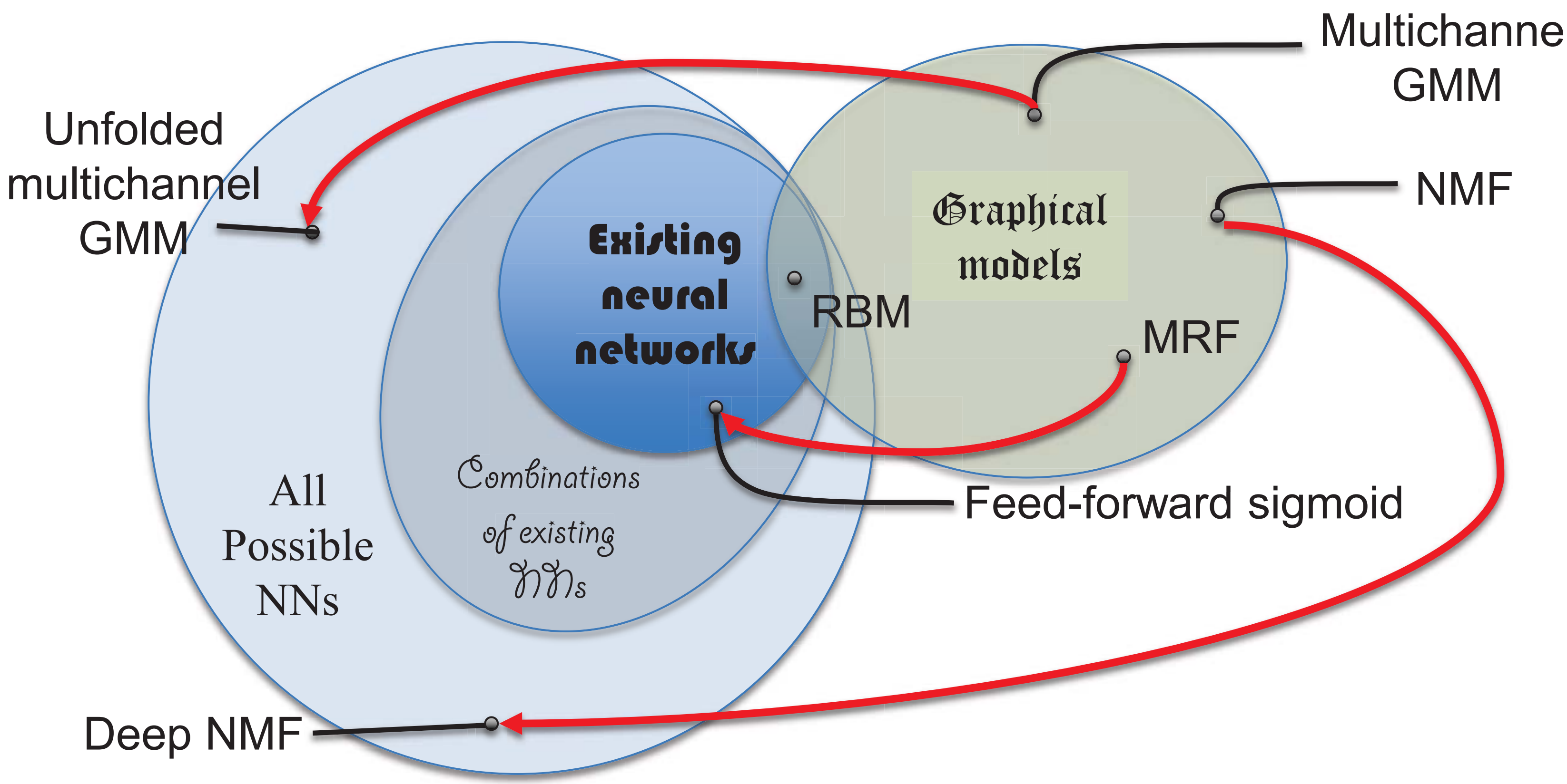


Problem level vs. mechanism level



Deep Unfolding: DNNs from generative models

- Is there a model whose inference algorithm is a DNN?
- Then we could explore model variations to get new DNNs

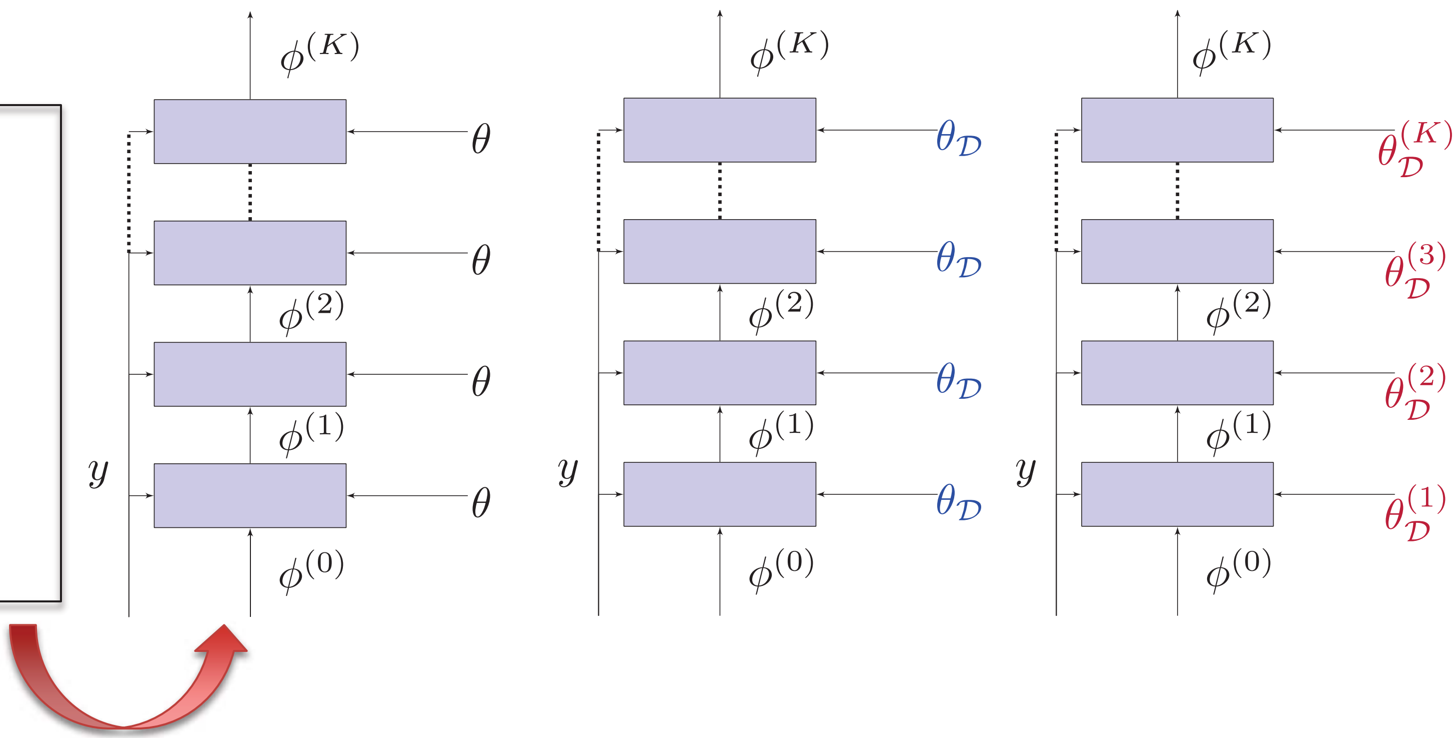


Deep Unfolding Recipe

1. Define model, train source models
2. Derive iterative inference algorithm
3. Unfold iterations into layers in a network
4. Discriminatively train the source parameters θ
5. Untie the parameters across layers

Examples:
NMF: $\phi \rightarrow \mathbf{H},\ \theta \rightarrow \mathbf{W}$
GMM: $\phi \rightarrow \bar{\boldsymbol{\pi}},\ \theta \rightarrow \boldsymbol{\mu}, \mathbf{v}$

Iterative algorithm:
For k=1:K,
Update $\phi^{(k)}$
using $\phi^{(k-1)}$
and data y

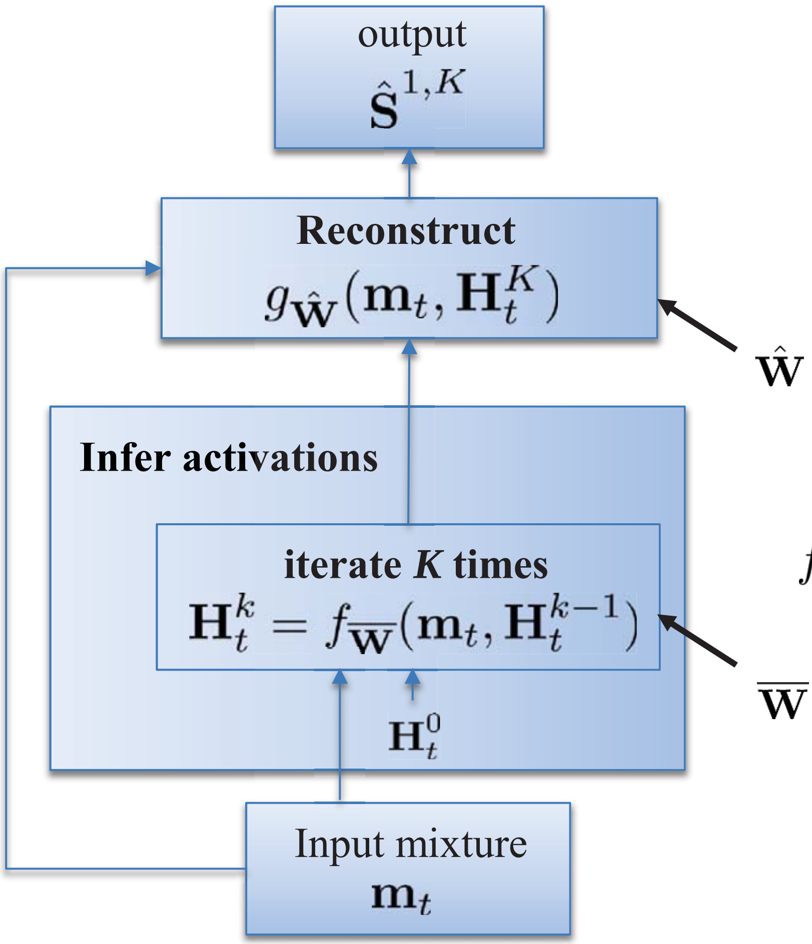


Example: unfolding NMF

- Based on simple approximations/assumptions:
  - Sources add in power spectrum
  - Sources represented as non-negative combination of non-negative bases
- Issue with classical NMF separation:
  - Speech/noise bases trained on isolated sources
  - At test time, get activation on the mixture

  Mismatch!
- Deep unfolding of NMF → "Deep NMF"
  - Unfolds inference algorithm from mixture to source estimates
  - Removes mismatch between training and test
  - Leads to radically different deep network architecture

Looking back at discriminative NMF



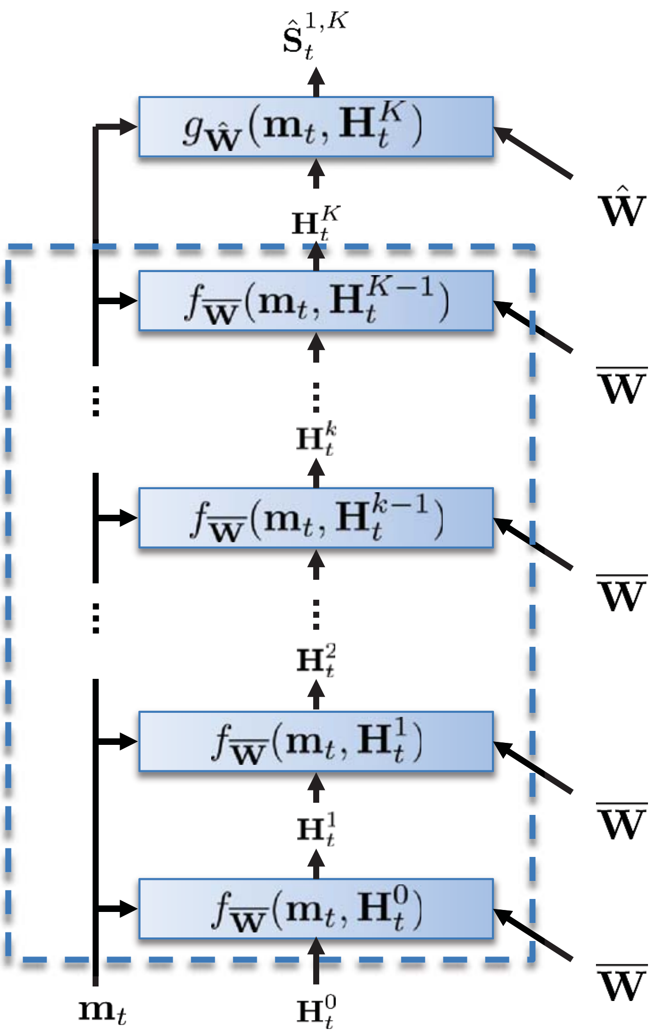
Reconstruction

$$g_{\hat{\mathbf{W}}}(\mathbf{m}_t, \mathbf{H}_t^K) = \frac{\hat{\mathbf{W}}^1 \hat{\mathbf{H}}^1}{\sum_l \hat{\mathbf{W}}^l \hat{\mathbf{H}}^l} \circ \mathbf{M}$$

Analysis

$$f_{\overline{\mathbf{W}}}(\mathbf{m}_t, \mathbf{H}_t^{k-1}) = \mathbf{H}_t^{k-1} \circ \frac{\overline{\mathbf{W}}^T (\mathbf{m}_t \circ (\overline{\mathbf{W}} \mathbf{H}_t^{k-1})^{\beta_1 - 2})}{\overline{\mathbf{W}}^T (\overline{\mathbf{W}} \mathbf{H}_t^{k-1})^{\beta_1 - 1} + \mu}$$

Unfolding the NMF iterations



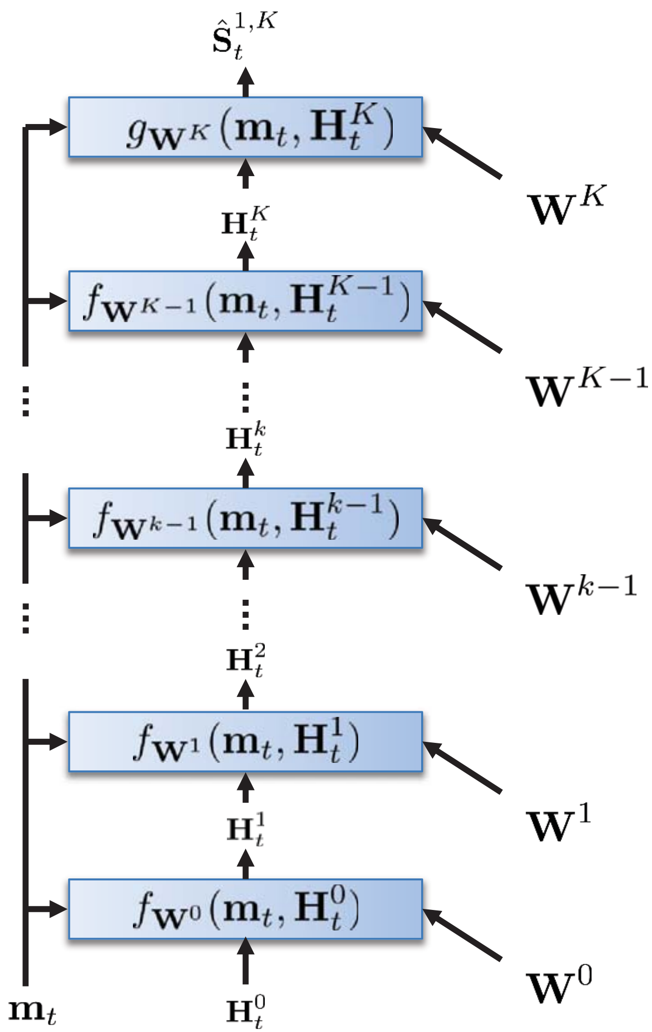
Reconstruction

$$g_{\hat{\mathbf{W}}}(\mathbf{m}_t, \mathbf{H}_t^K) = \frac{\hat{\mathbf{W}}^1 \hat{\mathbf{H}}^1}{\sum_l \hat{\mathbf{W}}^l \hat{\mathbf{H}}^l} \circ \mathbf{M}$$

Analysis

$$f_{\overline{\mathbf{W}}}(\mathbf{m}_t, \mathbf{H}_t^{k-1}) = \mathbf{H}_t^{k-1} \circ \frac{\overline{\mathbf{W}}^T (\mathbf{m}_t \circ (\overline{\mathbf{W}}\mathbf{H}_t^{k-1})^{\beta_1 - 2})}{\overline{\mathbf{W}}^T (\overline{\mathbf{W}}\mathbf{H}_t^{k-1})^{\beta_1 - 1} + \mu}$$

Deep NMF



Untying the basis sets $\mathbf{W}^k$ for each layer naturally leads to deep network architecture

- Activation coefficients $\mathbf{H}_t^k$: hidden layers
- Each layer's activation function $f_{\mathbf{W}^{k-1}}$ performs a multiplicative update:

$$\mathbf{H}_t^k = \mathbf{H}_t^{k-1} \circ \frac{(\mathbf{W}^{k-1})^T(\mathbf{m}_t \circ (\mathbf{W}^{k-1}\mathbf{H}_t^{k-1})^{\beta_1-2})}{(\mathbf{W}^{k-1})^T(\mathbf{W}^{k-1}\mathbf{H}_t^{k-1})^{\beta_1-1} + \mu}$$

- Input mixture $\mathbf{m}_t$ used in all layers
- Output function $g_{\mathbf{W}^K}$ computes enhanced speech estimate
- Parameters $\mathbf{W}^k$ can be trained using (non-negative, multiplicative) back-prop to minimize $\mathcal{E} = D_2(\mathbf{S}^1 \mid \hat{\mathbf{S}}^{1,K})$

Results on CHiME-2 speech enhancement task

- Significantly outperforms NMF
- Promising results, but not yet as good as the best nets

R^l bases/source
K unfolded iterations

Top C layers discriminatively trained

SDR [dB] — T context frames

$R^l = 1000,\ K = 25$	$T=1$	$T=3$	$T=5$	$T=9$
$C=0$ (SNMF)	9.0	9.4	9.5	9.5
$C=1$ (DNMF)	10.0	10.3	10.2	10.3
$C=2$	10.3	10.7	10.6	10.5
$C=3$				10.8
$C=4$				10.9

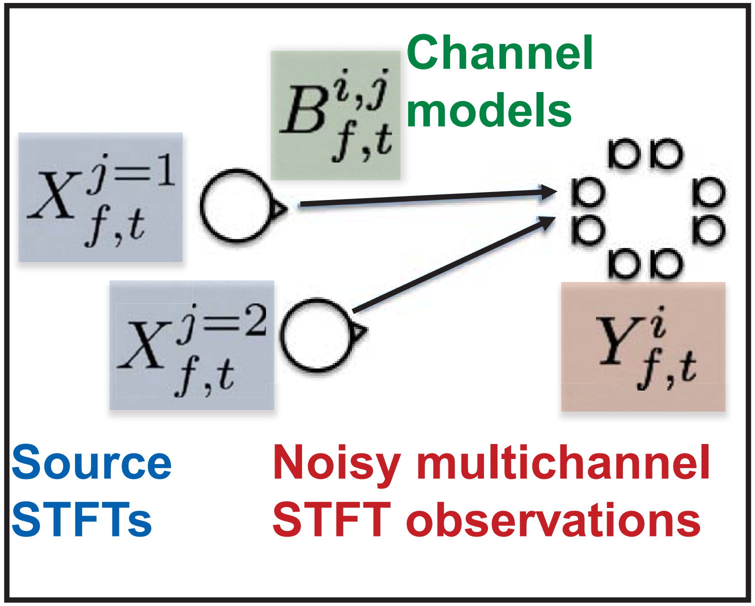
#params:				
	0.4 M	1.2 M	2.0 M	3.6 M
	0.8 M	1.6 M	2.4 M	4.0 M
	1.2 M	2.0 M	2.8 M	4.4 M
				4.8 M
				5.2 M

SDR [dB]			Avg.
DNN	dft	MA	10.5
DNN	dft	SA	11.2
DNN	mel	SA	11.5
LSTM	mel	MA	12.8
LSTM	mel	SA	13.0
LSTM	mel	PSA	13.4

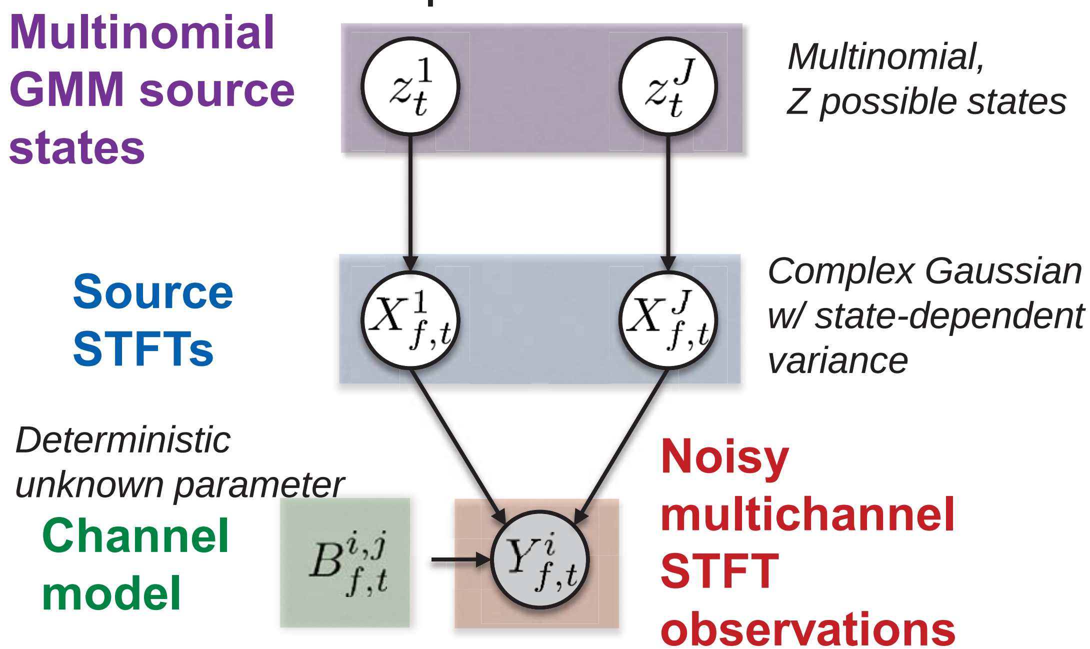
#params: 4.1 M

Generative model: multichannel GMM

Physical scenario



Graphical model

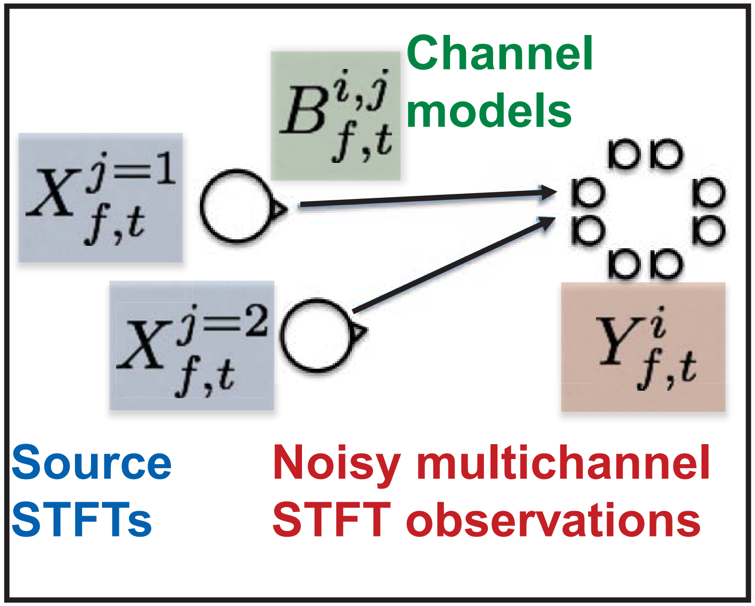


$$Y^i_{f,t} = \sum_j B^{i,j}_f X^j_{f,t} + V^i_{f,t}$$

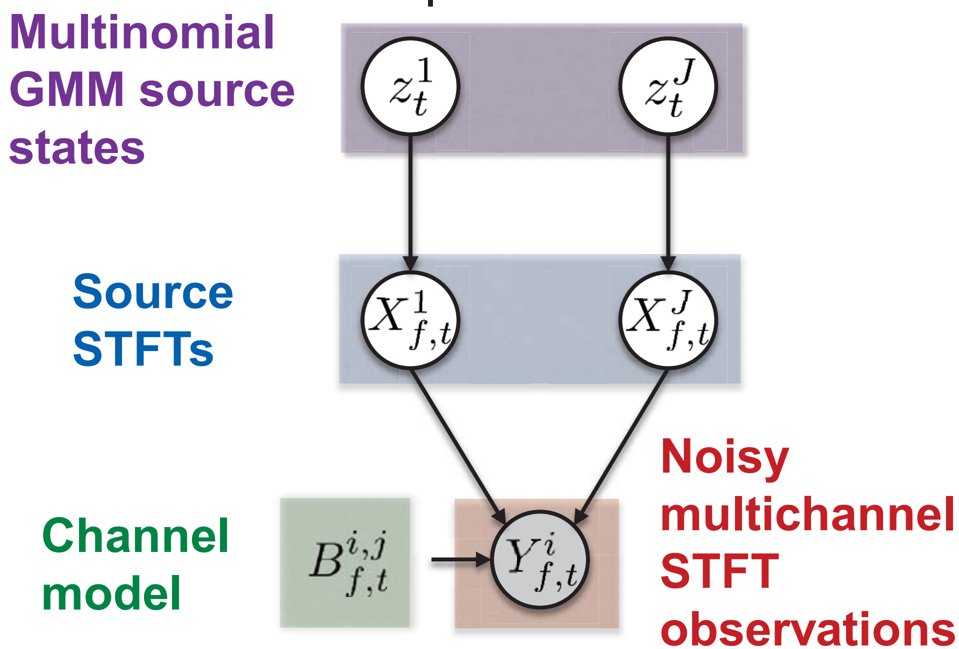
- Multichannel GMM (MCGMM): probabilistic model of complex-valued multichannel STFT
  - GMM source models
  - Narrowband channel model

Generative model: multichannel GMM

Physical scenario



Graphical model

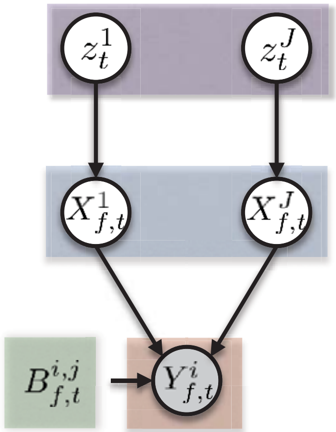


- For k=1:K, estimate:
  - Source GMM state probabilities
  - Source means (complex STFTs)
  - Channel model

<u>Iterative variational inference algorithm</u>

Unfolding the multichannel GMM

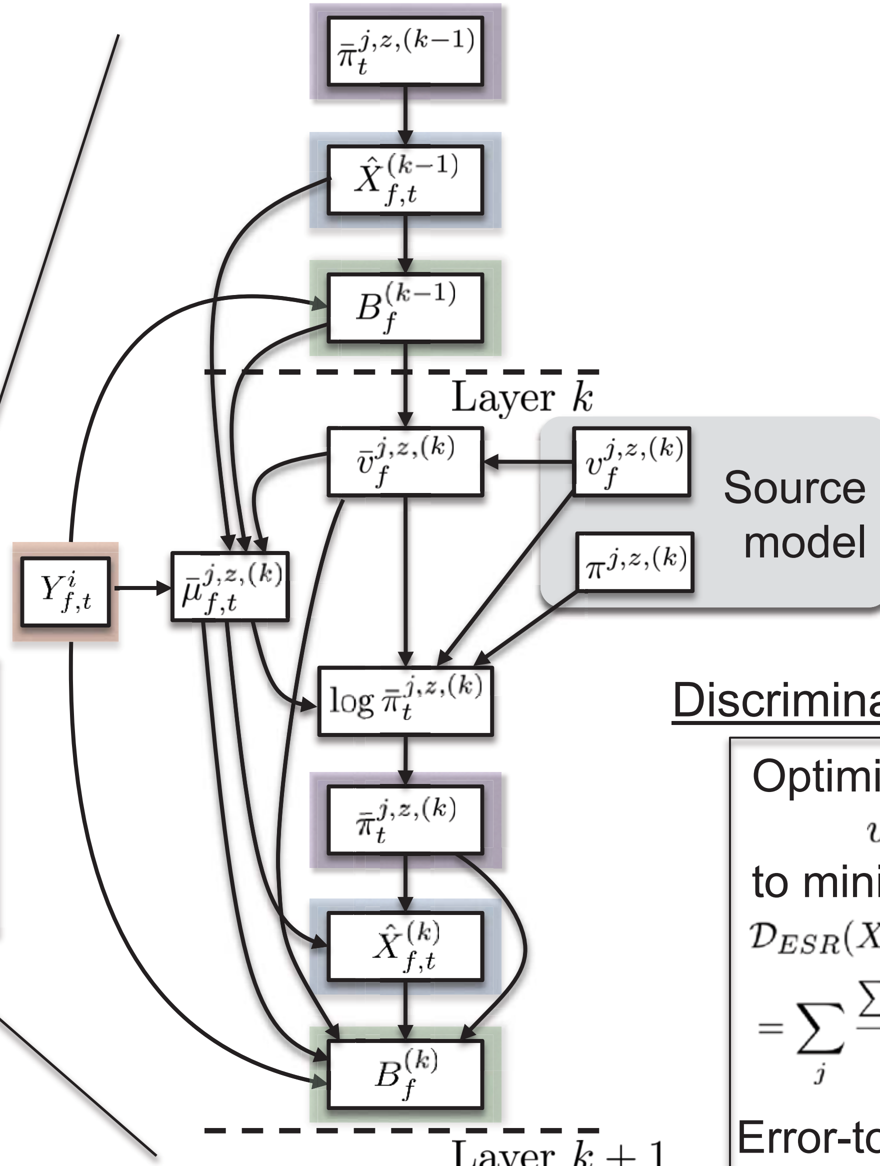
Graphical model



Iterative inference

- For k=1:K, estimate:
  - Source state probabilities
  - Source means
  - Channel model

One layer of unfolded network



Discriminative training:

Optimize

$$v^{(k)}, \ \pi^{j,z,(k)}$$

to minimize

$$\mathcal{D}_{ESR}(X_{f,t}, \hat{X}_{f,t}^{(K)})$$

$$= \sum_j \frac{\sum_{f,t} |\hat{X}_{f,t}^j - X_{f,t}^j|^2}{\sum_{f,t} |X_{f,t}^j|^2}$$

Error-to-source cost

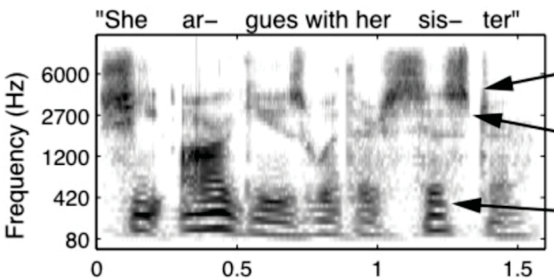
Deep clustering:
Cracking the general cocktail party problem

Solving the general source separation problem

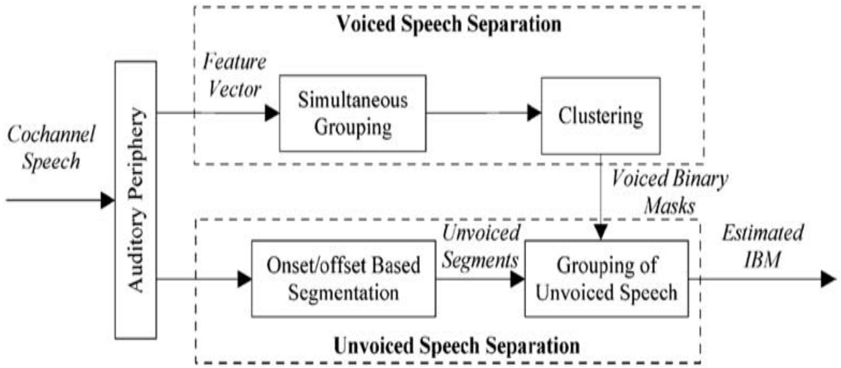
- To solve the general source separation problem, we need a method meeting the following requirements:
  - Single channel
  - Speaker/class independent
  - Discriminative
  - Practical complexity
- Previous approaches are missing some
- Should be feasible: humans do it

Clustering Approaches to Separation

- CASA approaches cluster based on hand-crafted similarity features

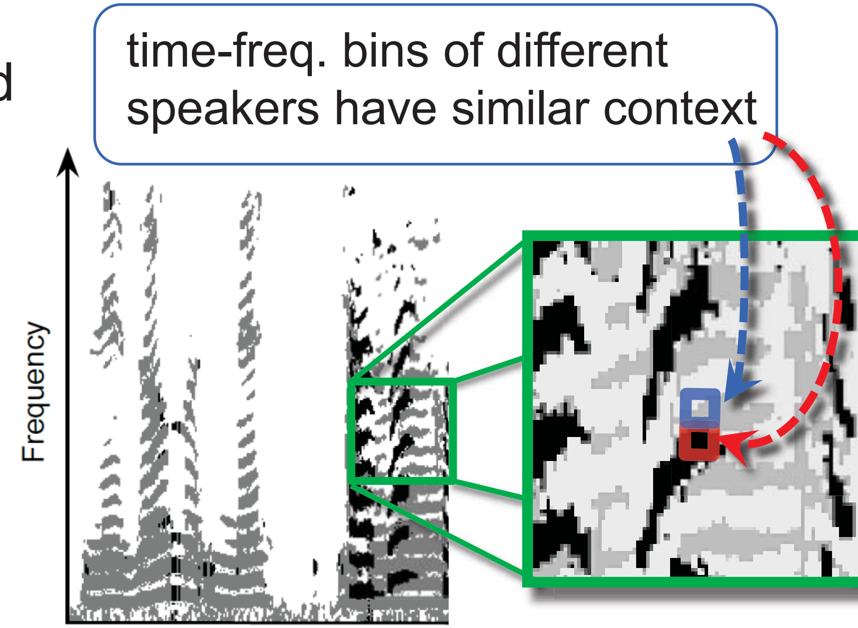


CASA system: Hu & Wang (2013)



  - Relative strengths of cues unknown
  - Learning from data not straightforward

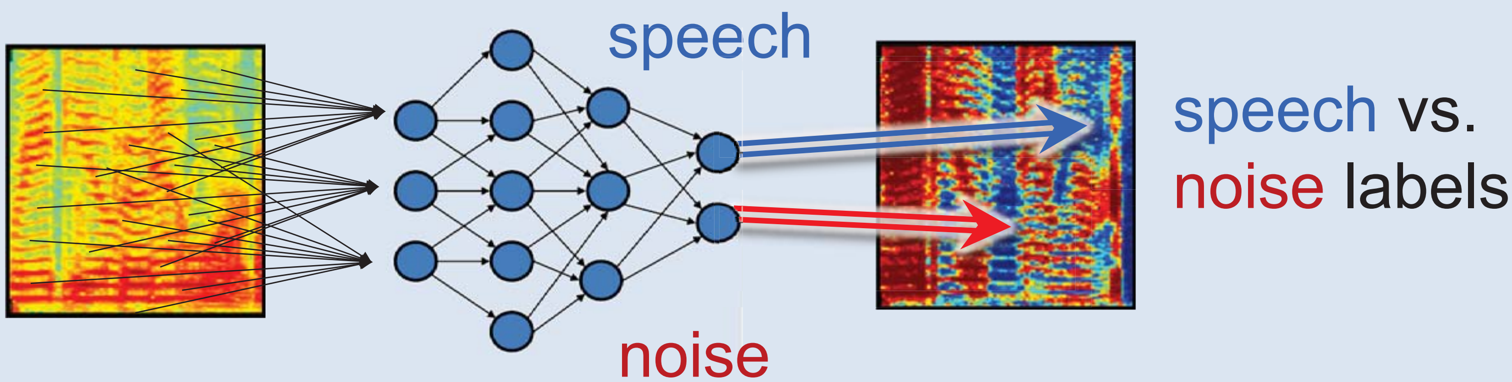
- Spectral clustering approach
  - eigen-decomposition → learning hard
  - need context of sources to measure similarity between T-F bins
  - context contains mixture of sources, so need to separate first (catch-22)



Spectral clustering: Bach & Jordan (2006)

Permutation problem for classification approaches

Classification approaches work for speech + noise,



… but what about speech + speech?

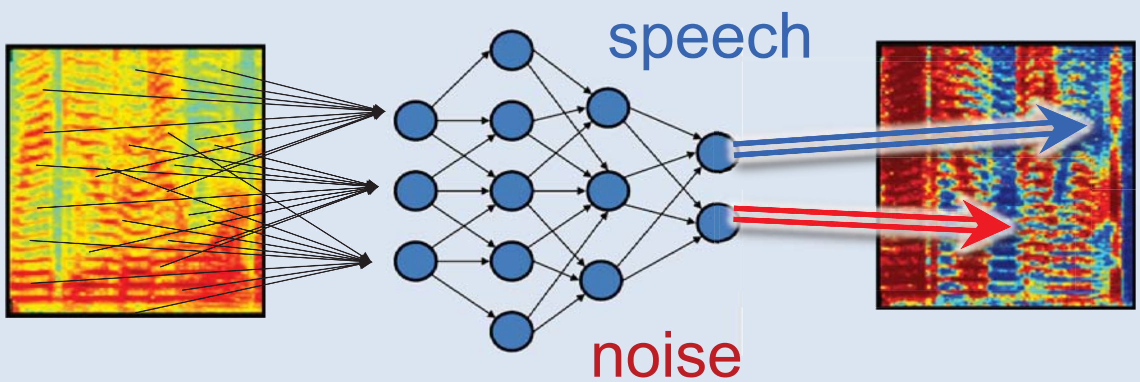
- **Need to handle the permutation problem:**

$$\mathcal{C}_E = \min_{\pi \in \mathcal{P}} \sum_{c,t,f} (\underbrace{s_{c,t,f}}_{c\text{-th reference}} - \underbrace{\tilde{s}_{\pi(c),t,f}}_{\pi(c)\text{-th estimate}})^2, \quad \mathcal{P} : \text{permutations on } \{1, \ldots, C\}$$

- **Even then, not easy for the network to learn**

Affinity-based training

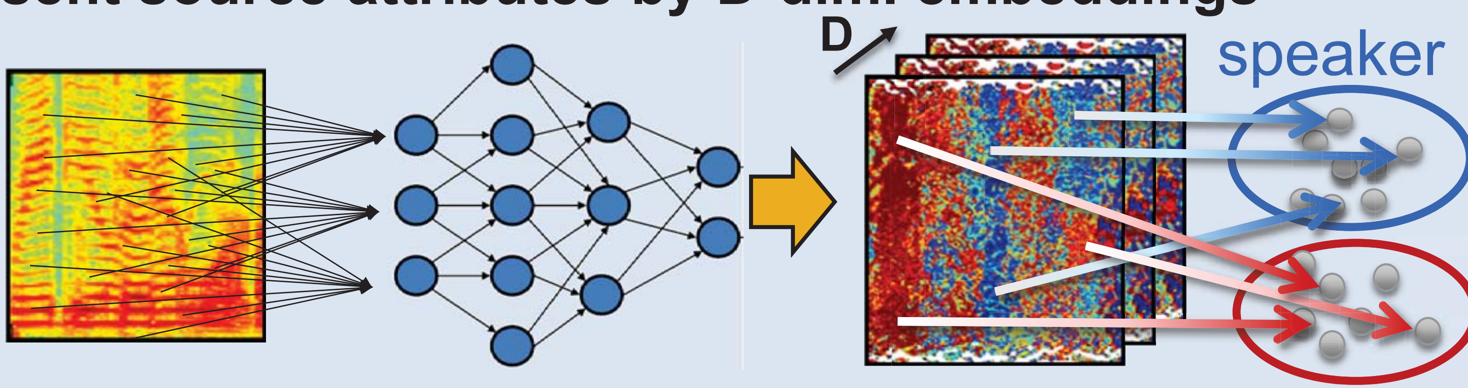
Classification approaches work for speech + noise



speech vs. noise labels

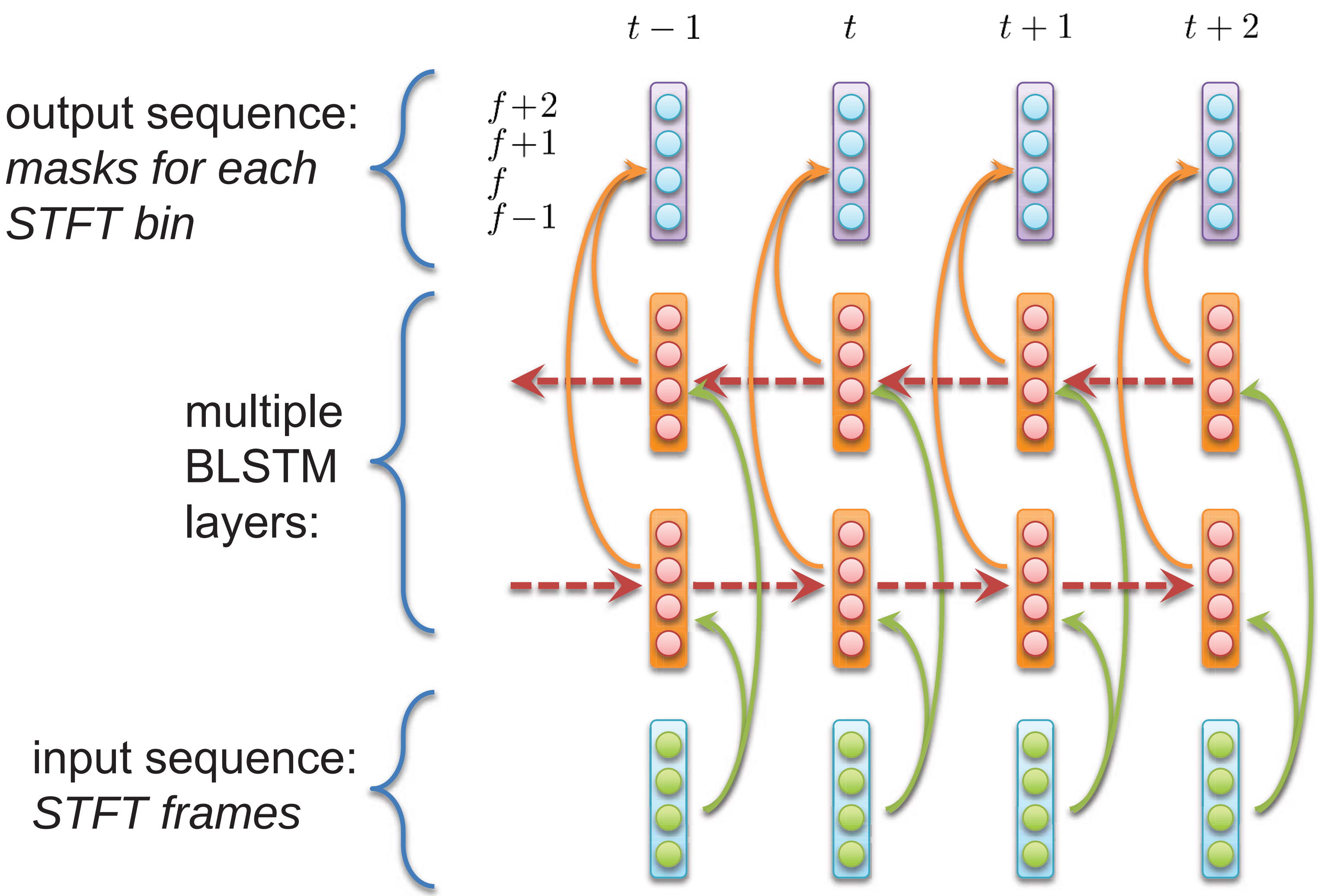
How can we handle speech + speech?

Key: represent source attributes by D-dim. embeddings

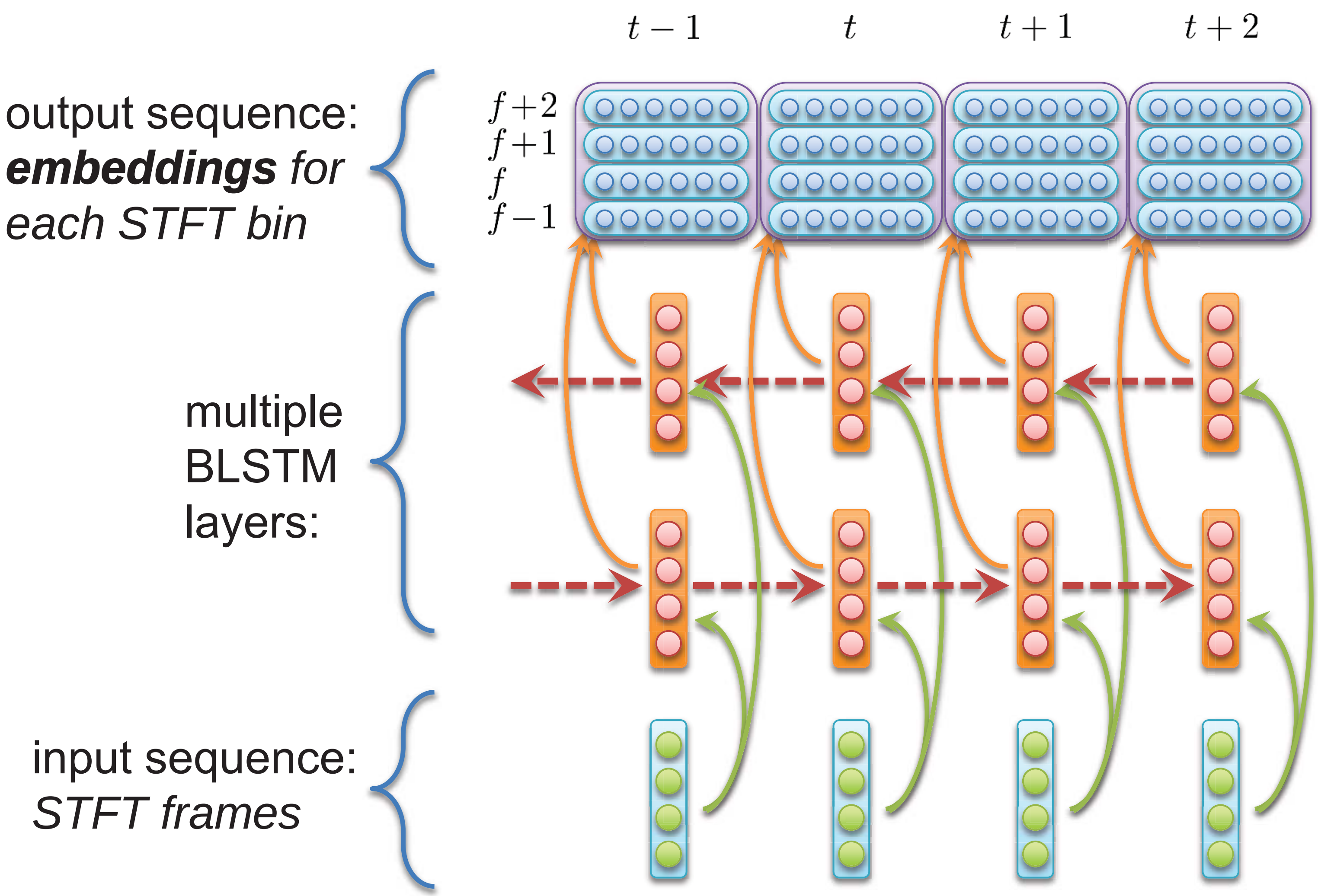


train network to map same-source bins close together speaker 2

Speech Enhancement / Separation Models



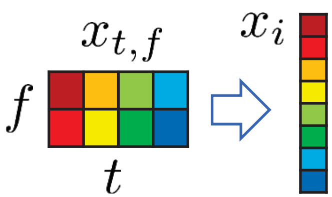
Embedding output



Deep clustering objective function

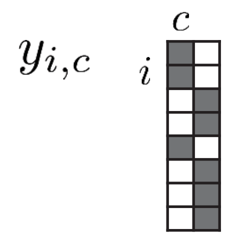
time-frequency bins: $i = (t, f)$

input spectrogram: $X = (x_i)$



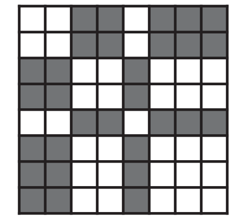
one-hot labels:

$Y = (y_{i,c})$



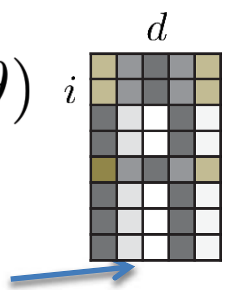
ideal affinity matrix:

$A = YY^T$



network embeddings: $V = (v_{i,d})$

$$v_{i,d} = h_{i,d}(X; \theta)$$

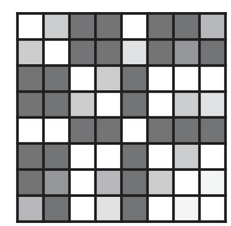


unit length constraint:

$$v_i = \frac{v_i}{|v_i|}$$

estimated affinities

$\hat{A} = VV^T$

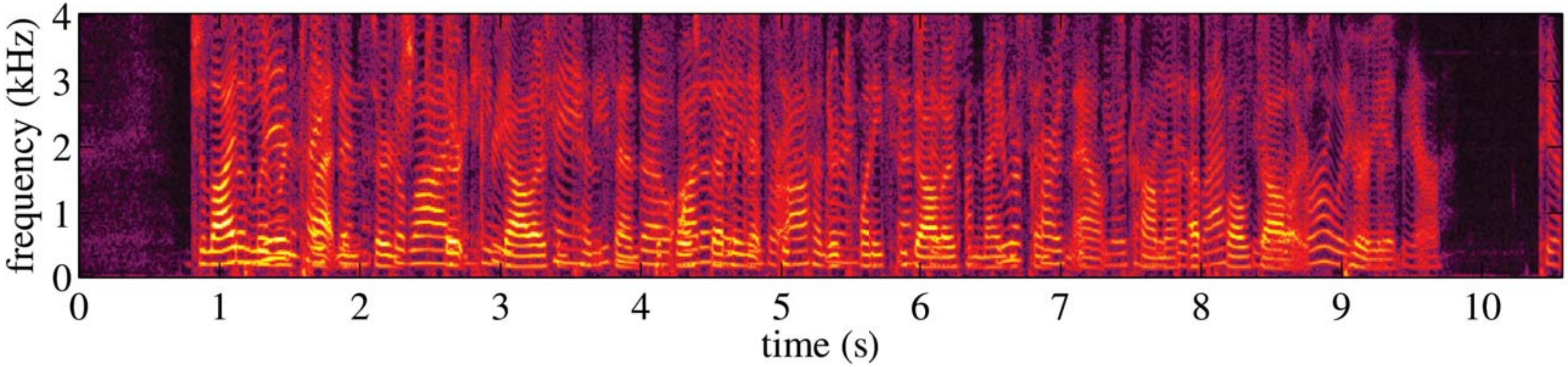


objective: minimize error in affinities over training examples

$$C(\theta, Y) = |\hat{A} - A|_{\mathrm{F}}^2 = |VV^T - YY^T|_{\mathrm{F}}^2$$

$$= \sum_{i,j: y_i = y_j} \left(|v_i - v_j|^2 - 1\right) + \sum_{i,j} \frac{1}{4} \left(|v_i - v_j|^2 - 2\right)^2,$$

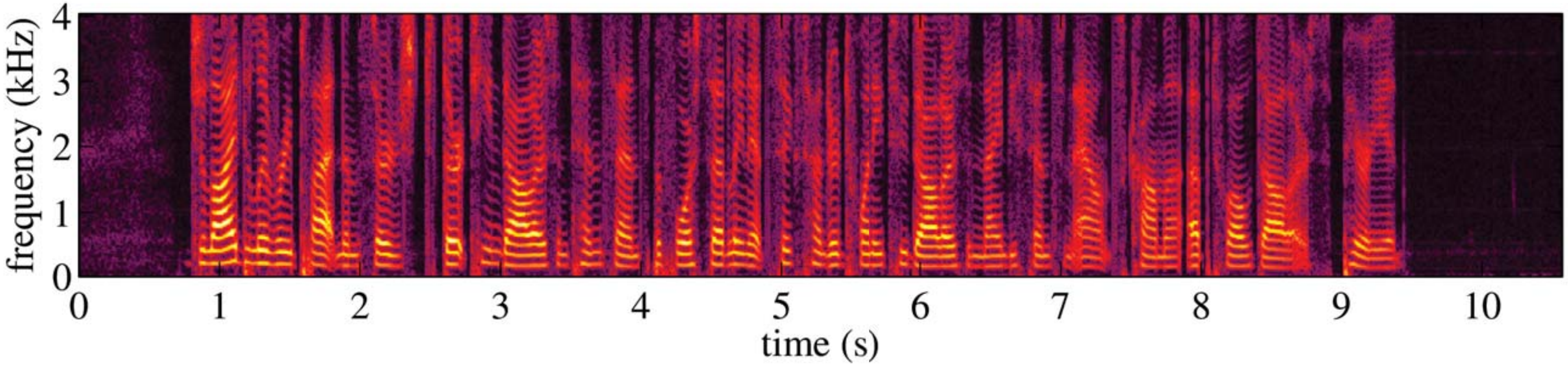
Mixture of two female speakers



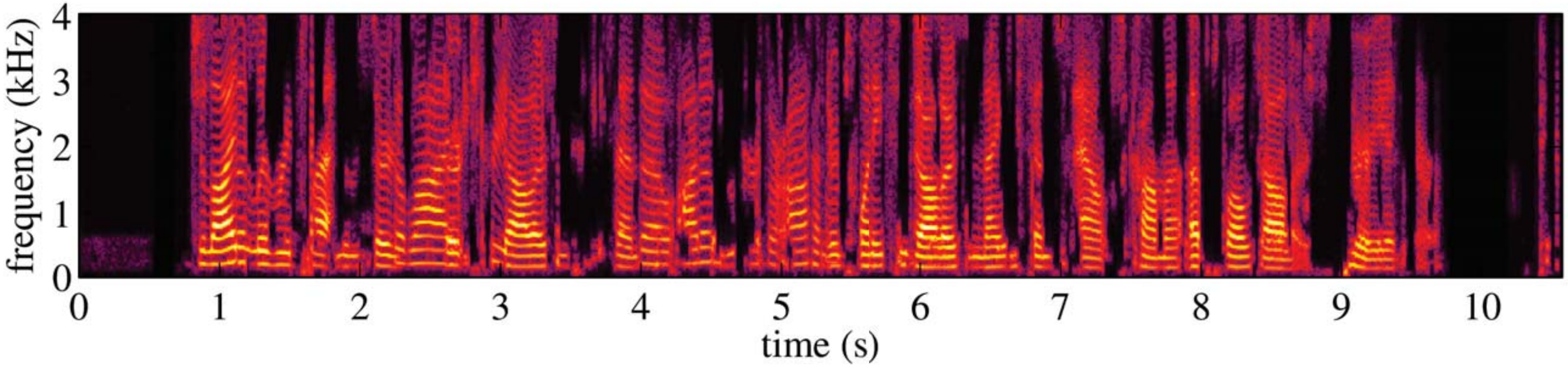
Mixture

Mixture of two female speakers

Speaker 1



Estimate 1

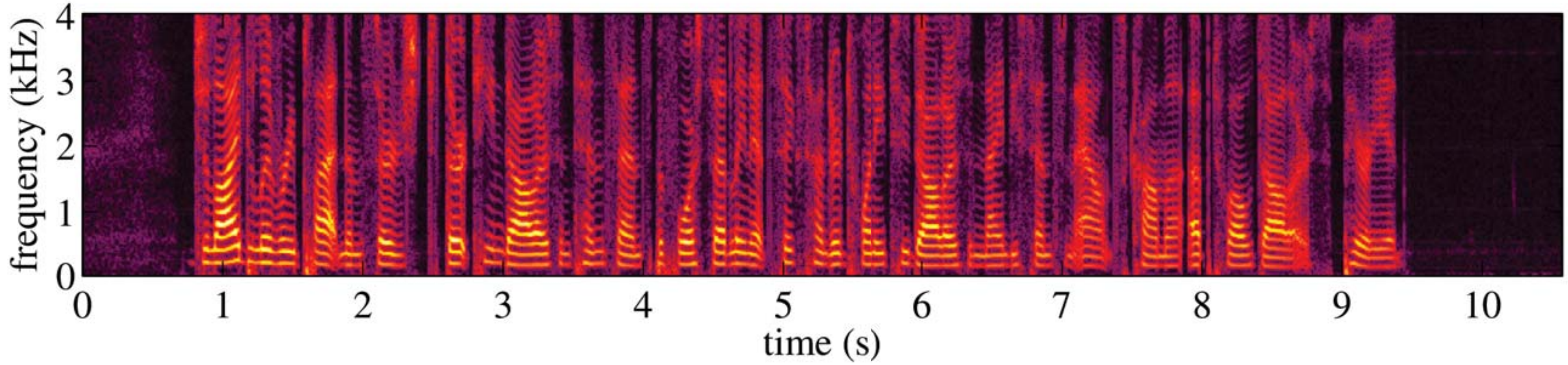


Mixture

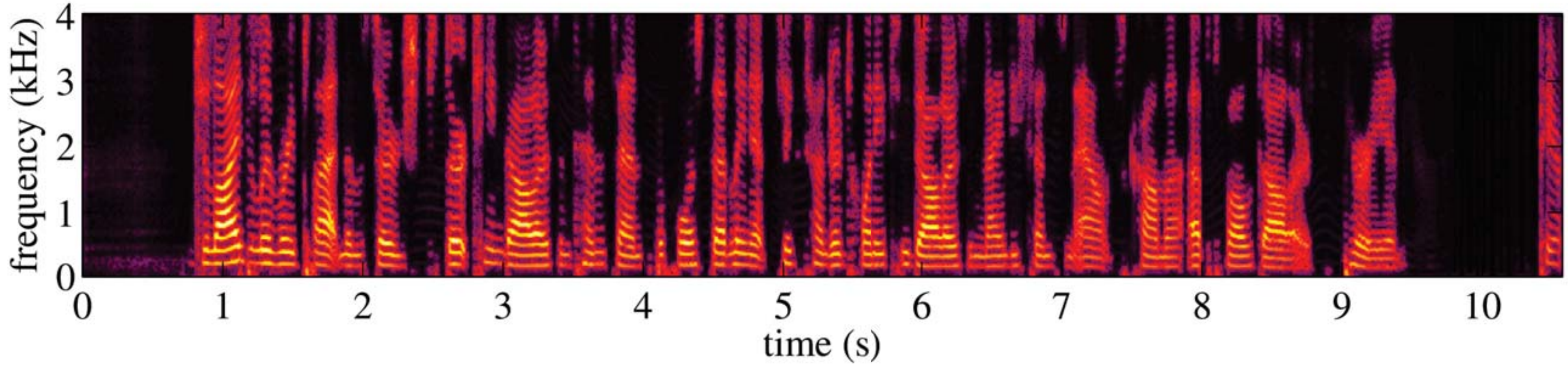
CASA

Mixture of two female speakers

Speaker 1



Estimate 1



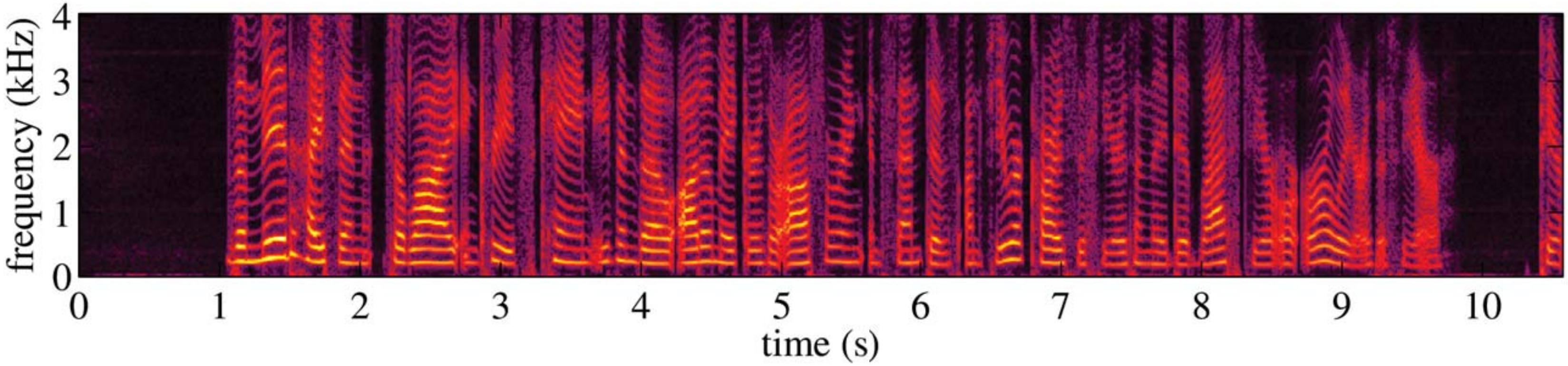
Mixture

CASA

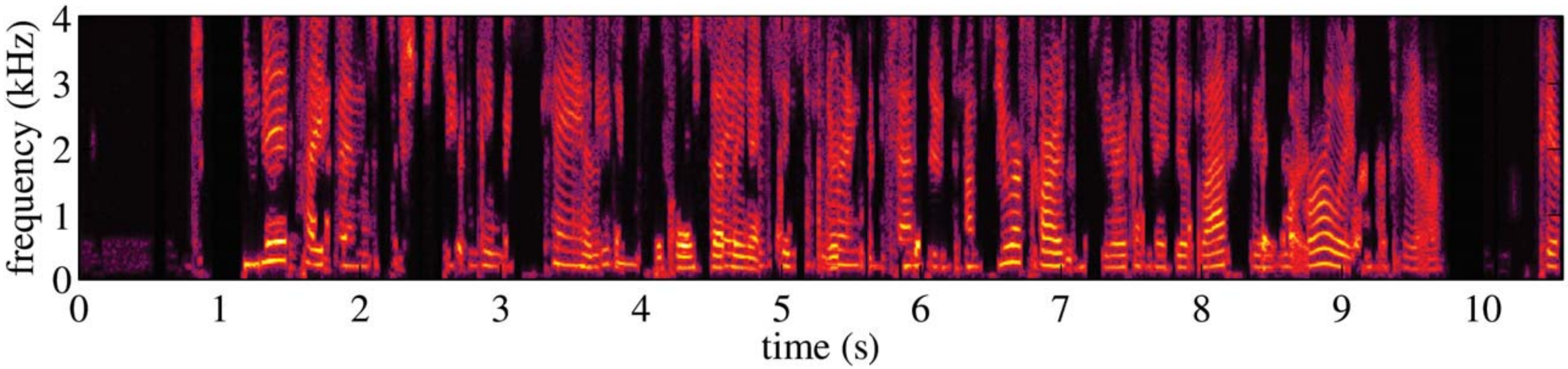
Deep Clustering

Mixture of two female speakers

Speaker 2



Estimate 2



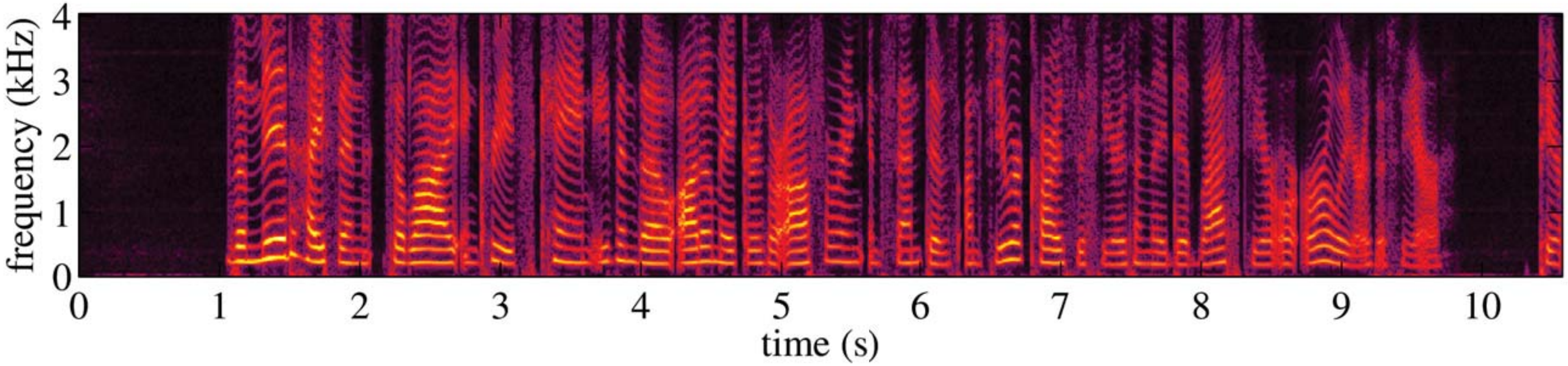
Mixture

CASA

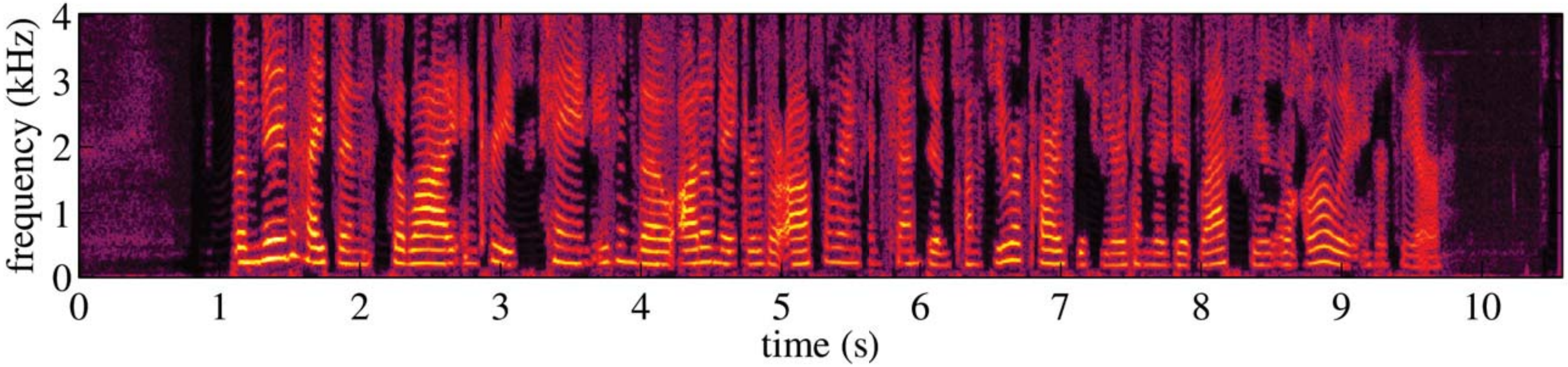
Deep Clustering

Mixture of two female speakers

Speaker 2

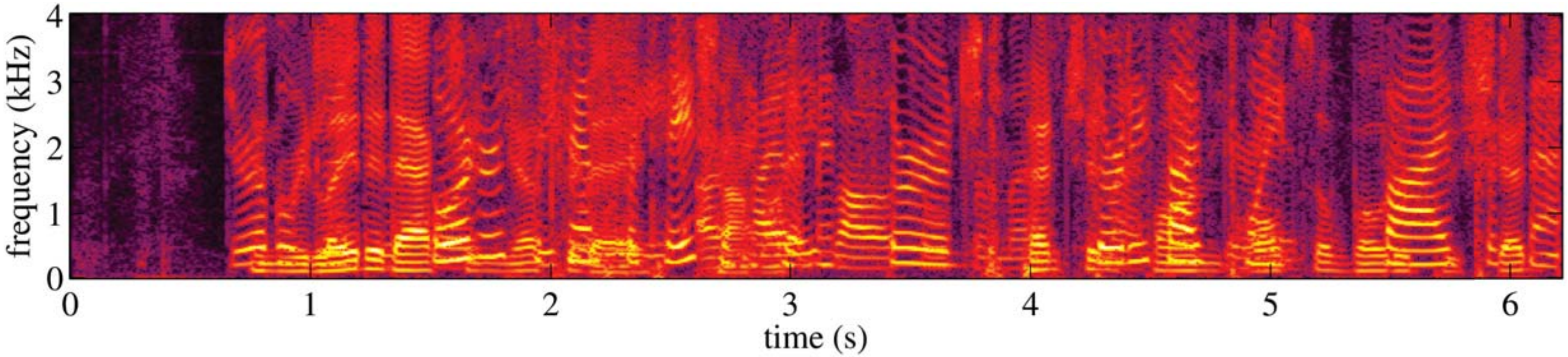


Estimate 2

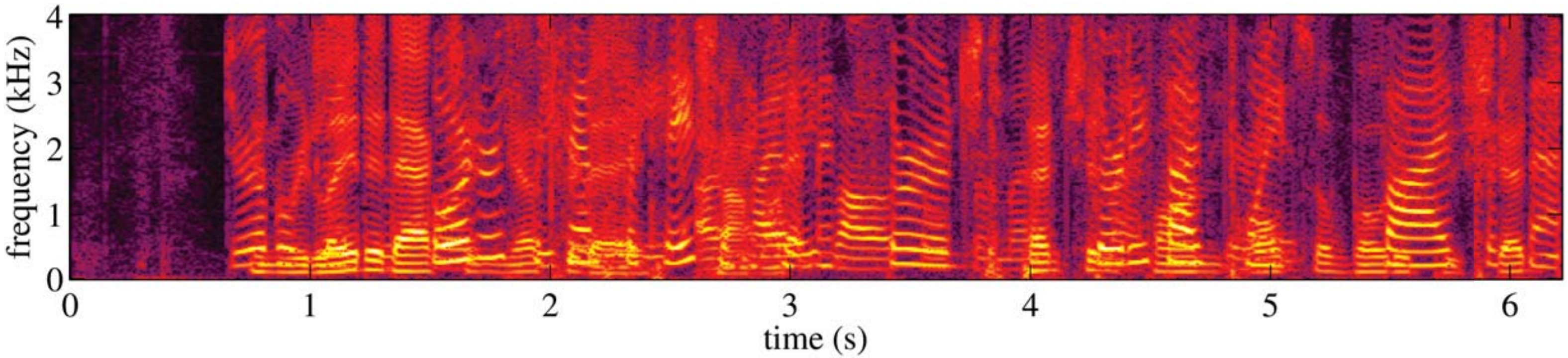


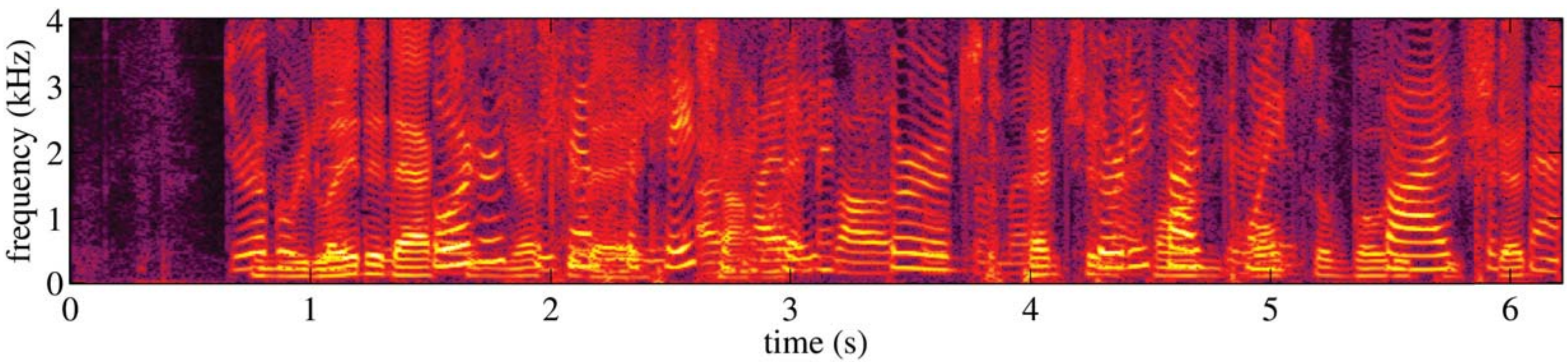
Mixture CASA Deep Clustering

Mixture of three speakers



Mixture

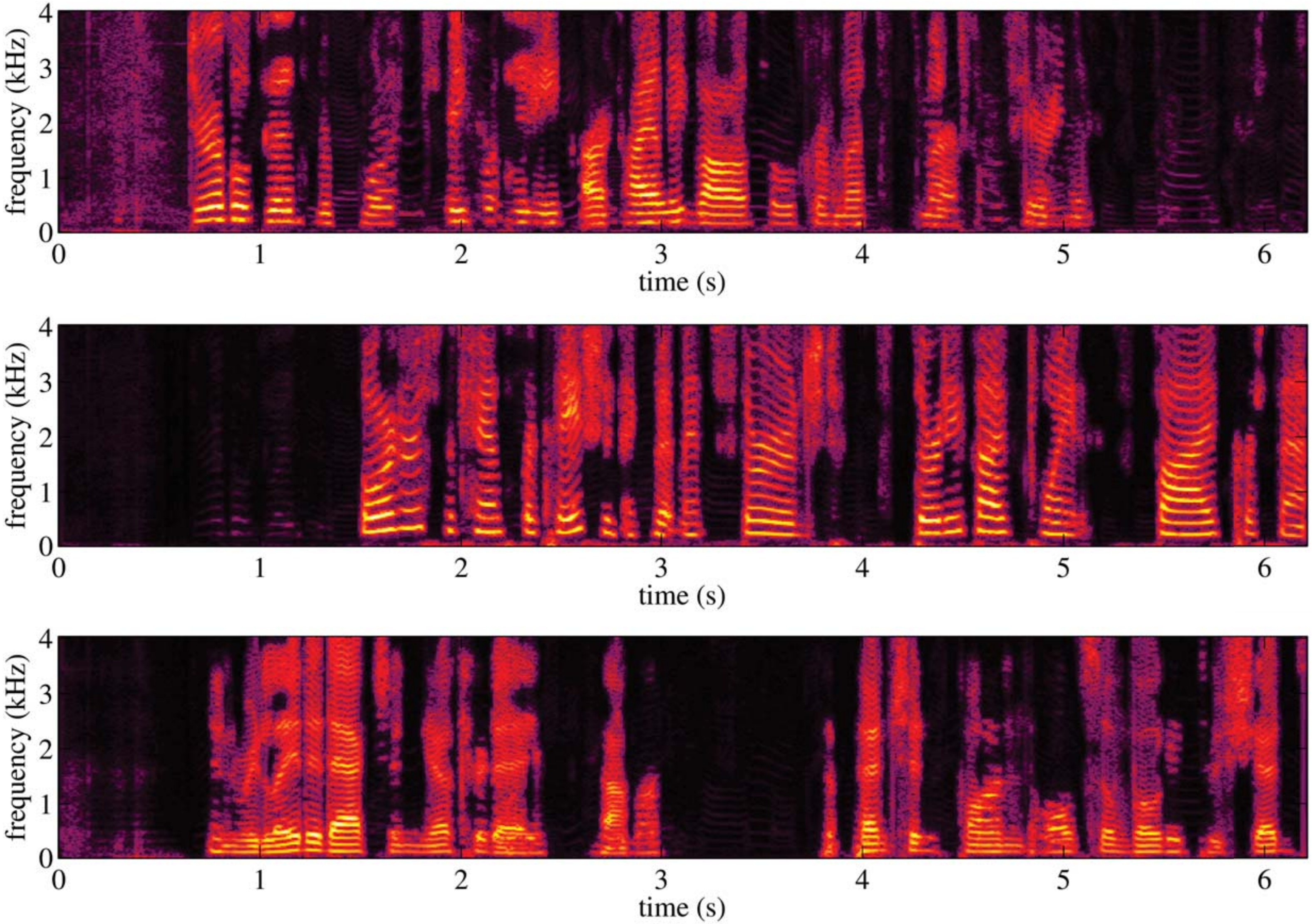




Mixture of three speakers

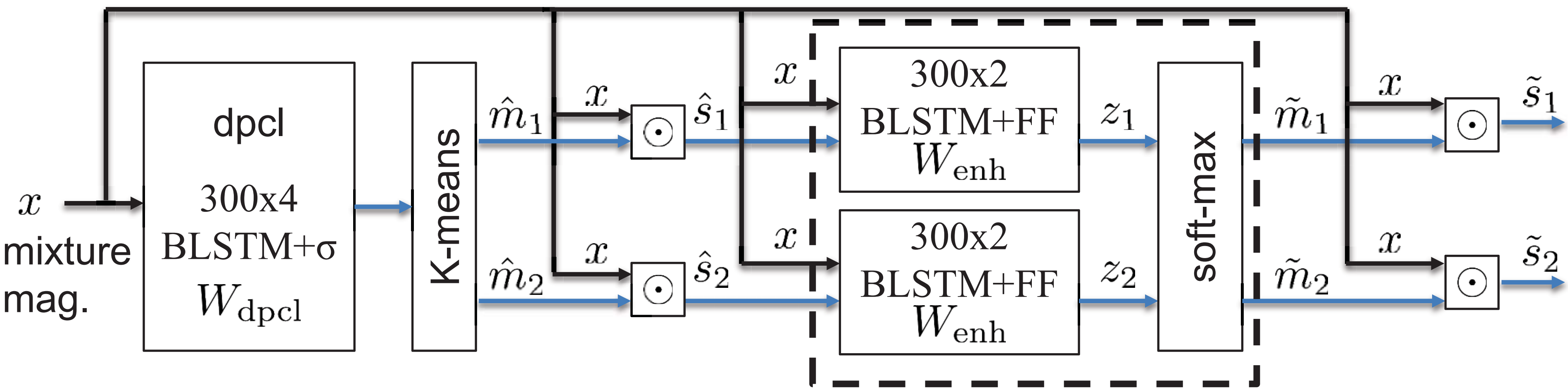
Deep Clustering

Mixture



Optimization for signal reconstruction

- Deep clustering is key, but only first step
  - segment spectrogram into regions dominated by same source
  - does not recover sources in regions dominated by other sources
- Use 2nd-stage enhancement network to improve estimates

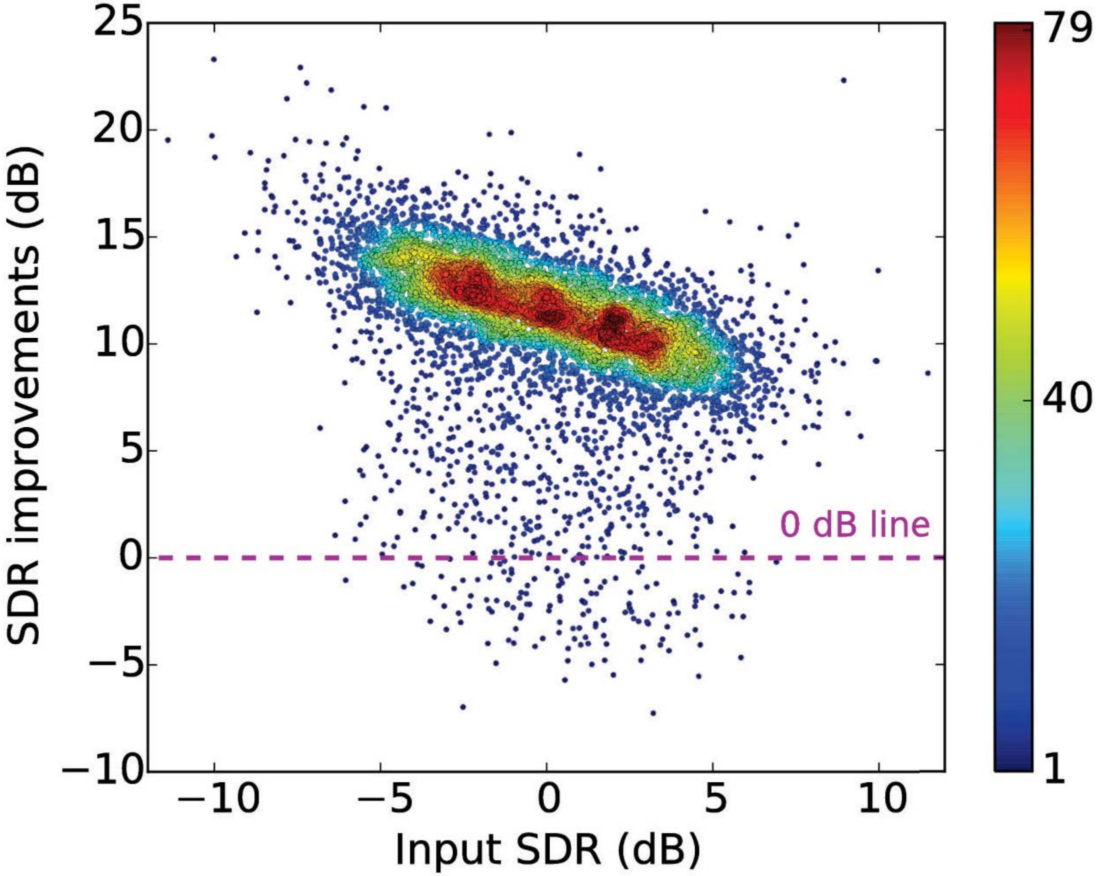


- Permutation-independent objective

$$\mathcal{C}_E = \min_{\pi \in \mathcal{P}} \sum_{c,i} (s_{c,i} - \tilde{s}_{\pi(c),i})^2$$

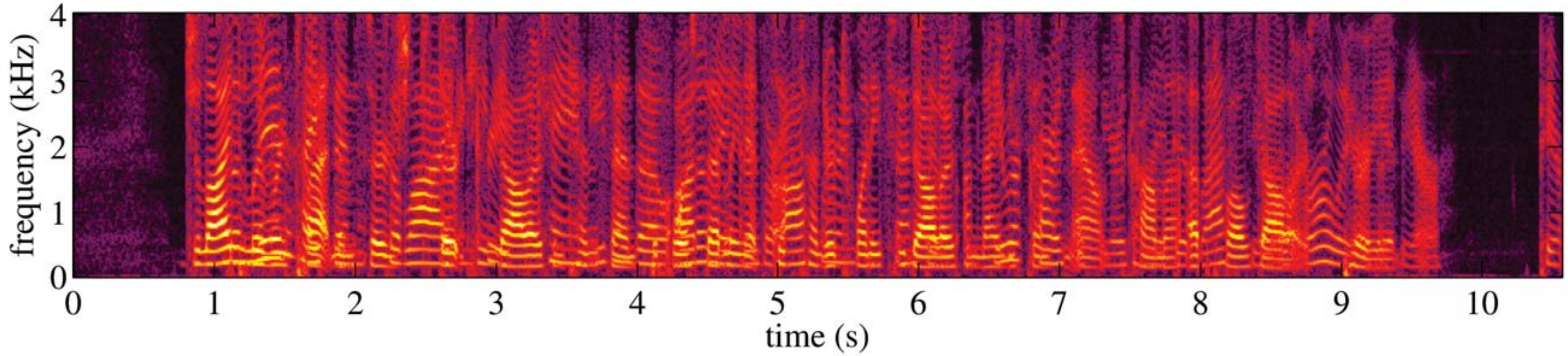
$\mathcal{P}$: permutations on $\{1, \ldots, C\}$

Two-speaker deep clustering results



method	SNR improvement
end-to-end deep clustering	**10.8 dB**
ideal binary mask	13.5 dB

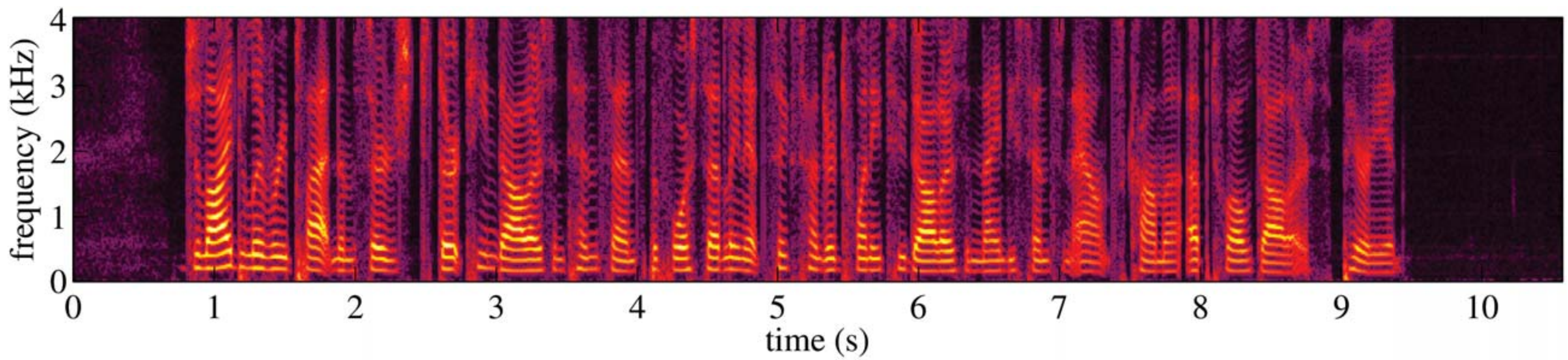
Mixture of two female speakers



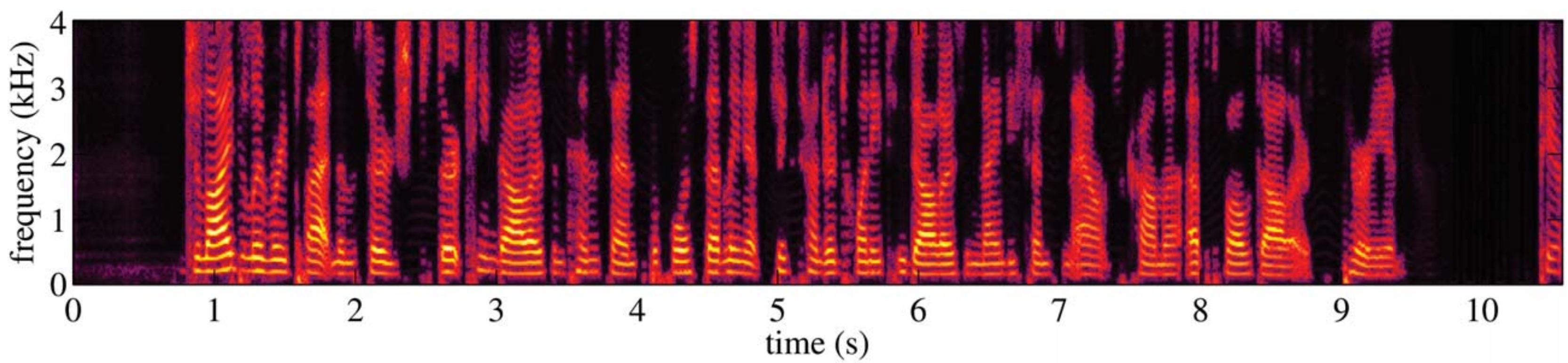
Mixture

Mixture of two female speakers

Speaker 1



Estimate 1

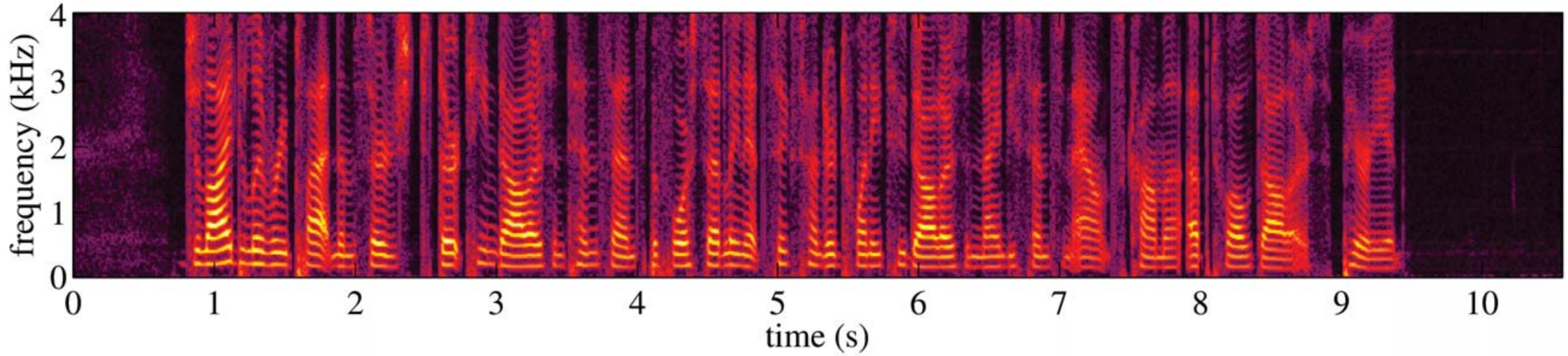


Mixture

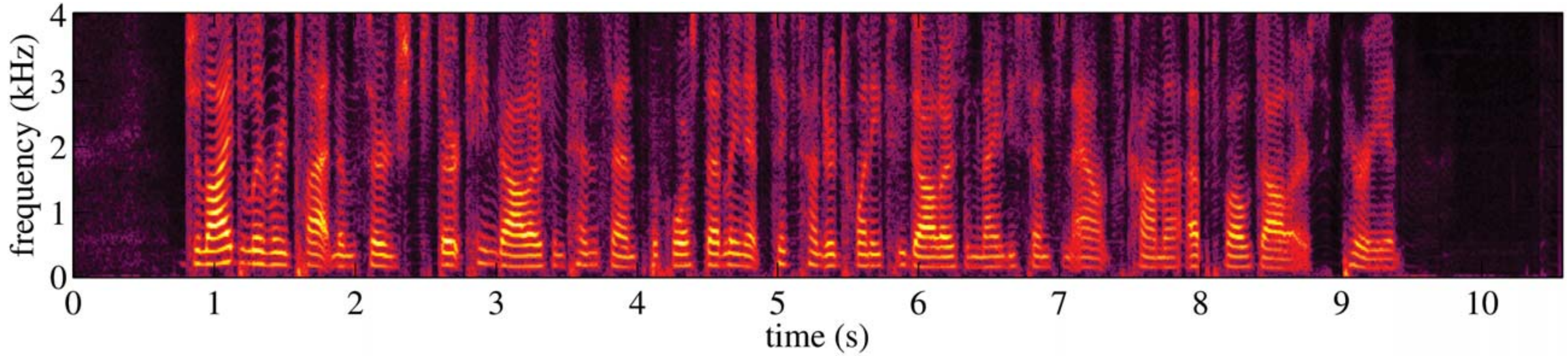
Deep Clustering

Mixture of two female speakers

Speaker 1



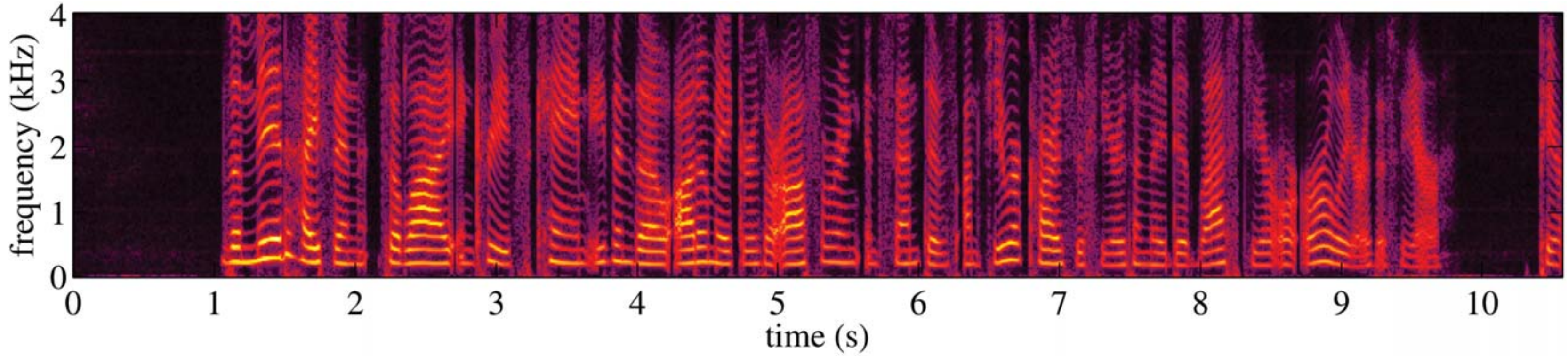
Estimate 1



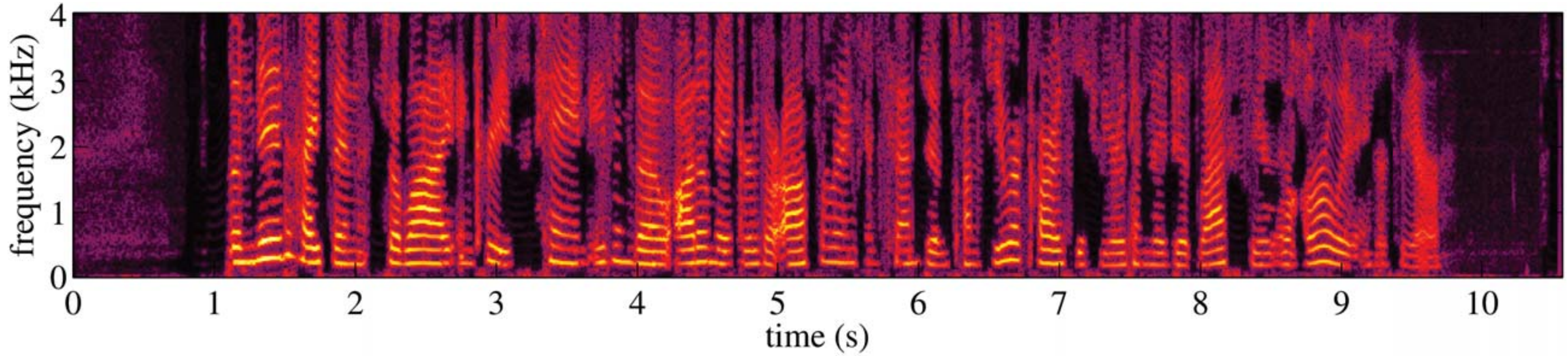
Mixture Deep Clustering End-to-end

Mixture of two female speakers

Speaker 2



Estimate 2



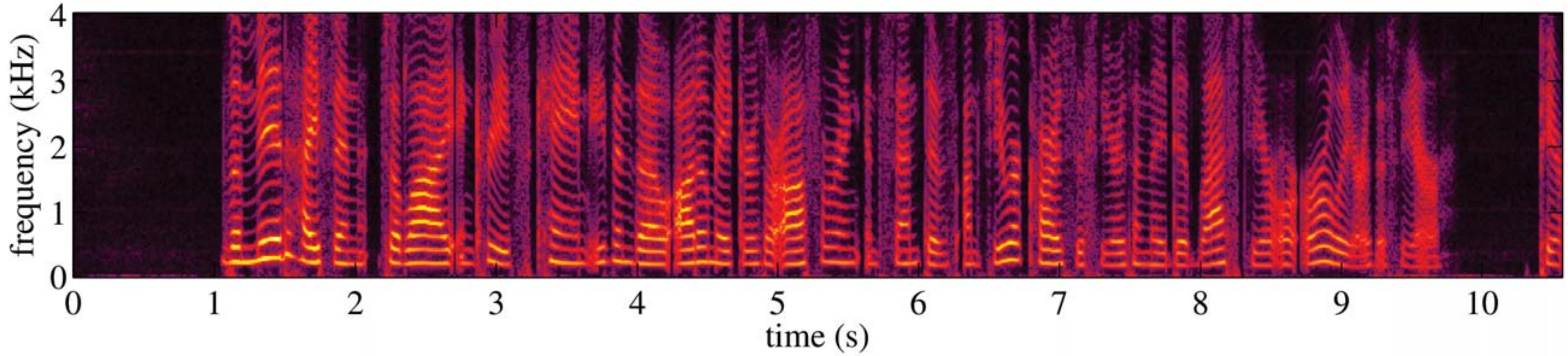
Mixture

Deep Clustering

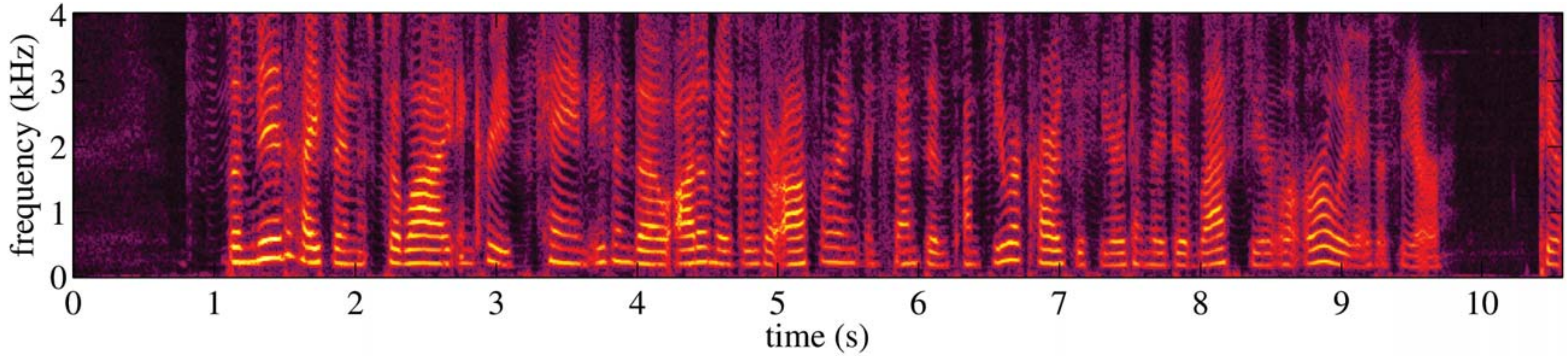
End-to-end

Mixture of two female speakers

Speaker 2



Estimate 2



Mixture Deep Clustering End-to-end

ASR performance

- GMM-based clean-speech WSJ models (Kaldi recipe)
- Despite very good perceptual quality, raw deep clustering (dpcl) not good for ASR, most likely due to near-zero regions
- Enhancement network dramatically improves the results

model	Mag. SNR imp.	WER
noisy	-	89.1 %
dpcl	10.2 dB	87.9 %
dpcl + enh	12.3 dB	32.8 %
end-to-end	12.5 dB	30.8 %
clean	-	19.9 %

References:

- F. Weninger, J. Le Roux, J.R. Hershey, B. Schuller, "Discriminatively Trained Recurrent Neural Networks for Single-Channel Speech Separation," GlobalSIP 2014
- J.R. Hershey, J. Le Roux, F. Weninger, "Deep Unfolding: Model-Based Inspiration of Novel Deep Architectures," arXiv:1409.2574, MERL Tech. Rep., TR2014-117, 2014
- J. Le Roux, J.R. Hershey, F. Weninger, "Deep NMF for Speech Separation," ICASSP 2015
- H. Attias, "New EM algorithms for source separation and deconvolution with a microphone array," ICASSP, 2003
- S. Wisdom, J.R. Hershey, J. Le Roux, S. Watanabe, "Deep Unfolding for Multichannel Source Separation," ICASSP 2016

- J.R Hershey, Z. Chen, J. Le Roux, S. Watanabe, "Deep Clustering: Discriminative Embeddings for Segmentation and Separation," ICASSP 2016
- K. Hu and D. Wang, "An unsupervised approach to cochannel speech separation," IEEE TASLP, 2013
- F. Bach and M. Jordan, "Learning spectral clustering, with application to speech separation", JMLR, 2006
- Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, J.R. Hershey, "Single-Channel Multi-Speaker Separation using Deep Clustering," Interspeech 2016

Wrap-up, perspectives

Speaker: Emmanuel Vincent

Learning-based separation: pros and cons

	learning-free	learning-based
addresses all scenarios	🙁	🙂
data collection effort	🙂	😐
computation time, memory, latency	🙂	😐
separation performance in matched conditions	🙁 to 😐	🙂
separation performance in mismatched conditions	🙁 to 😐	😐 to 🙂

Deep learning-based separation: pros and cons

	generative model based	deep learning based
ease of derivation	☹️	🙂
interpretability	🙂	☹️ to 😐
separation performance in matched conditions	😐	🙂
separation performance in mismatched conditions	😐	😐 to 🙂
computation time (test)	☹️	😐 to 🙂
latency	😐	😐

What was missing in early neural networks?

Earlier uses of neural networks unsuccessful mainly because of

- smaller-sized networks,
- smaller training datasets,
- computational power severely behind current capabilities,
- deep learning tricks like dropout, maxout, ReLU, batch normalization, unsupervised and supervised pretraining not invented yet.

But also

- time-domain or spectral domain prediction instead of mask prediction,
- LSTMs not widespread.

Which pre-deep-learning concepts are we still using?

- Pre-processing:
  - STFT or Mel spectra,
  - 3D auditory motivated features,
  - spatial features,
  - localization and beamforming.
- Intermediate target: time-frequency mask.
- Post-processing:
  - classical post-processing techniques: oversubtraction, thresholding, smoothing. . .
  - spatial filter derivation from source statistics (beamforming),
  - masking and inverse STFT (overlap-add method).

For how long still?

Single-channel phase modeling

Currently:

- model magnitude spectrum, interchannel phase,
- single-channel phase notoriously harder to model

Ideas to explore further:

- test magnitude STFT inversion methods (Griffin & Lim) with DNN,
- predict real and imaginary parts of complex mask,
- complex networks (with complex weights) used in deep unfolding but really useful (restriction of twice larger real network)?

Progress in deep unfolding

Currently, several generative models already unfolded:

- deep NMF for single-channel enhancement,
- deep multichannel MRF-GMM model for multichannel separation,
- end-to-end deep clustering for single-channel speaker separation.

Ideas to explore further:

- key enabling technology for end-to-end processing: discriminative training of iterative algorithms,
- strong potential in adaptation scenarios: balance between power of discriminative training and regularization ability of a generative model for better generalization.

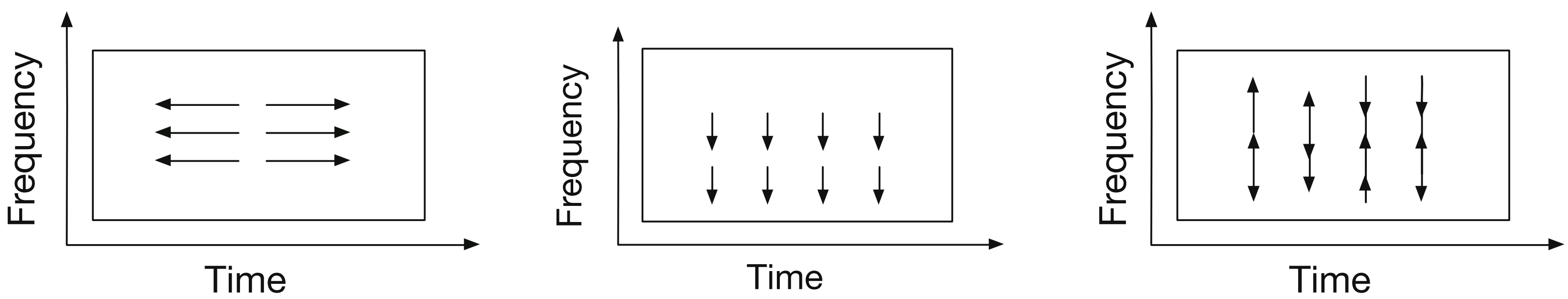
Data simulation

Currently:

- most DNNs trained on simulated mixtures of speech and noise,
- training on real data feasible but approximate target (close-talk mic),
- real training data sometimes better than simulated, sometimes not.

Ideas to explore further:

- understand why simulated training data work or not,
- don't just add speech and noise but explore perturbation of impulse response, SNR, noise rate, vocal tract length, frequency axis. . .



- bystep the limitations of acoustic simulation techniques by teaching a DNN how to simulate data.

Learning from mixtures only

Currently: high quality isolated source signals used

- to train a model of each source,
- or as targets for DNN training.

Not always feasible, e.g., new speaker or new noise unseen in isolation.

Ideas to explore further:

- are low quality source signals still useful?
- semi-supervised training from mixtures without knowledge of underlying source signals.

Robustness to unseen, mismatched conditions

Currently: DNNs experimentally more robust than one would think.

CHiME-3 WER achieved by multichannel DNN+EM enhancement

Training (real)	Test (real) BUS	CAF	PED	STR	Avg.
BUS	21.03	13.06	17.92	9.28	15.32
CAF	31.48	13.15	16.95	8.78	17.59
PED	27.89	12.20	17.04	8.93	16.51
STR	24.30	11.80	16.42	8.48	15.25
1/4 of all	20.83	11.65	15.94	8.72	14.28
all but BUS	22.62	10.72	15.47	7.55	14.09
all but CAF	18.90	10.59	16.07	7.53	13.27
all but PED	18.56	10.76	14.93	8.09	13.08
all but STR	18.19	10.03	15.08	7.94	12.81
3/4 of all	18.84	10.98	15.41	7.79	13.26

1 training environment:
Multicondition: 14.28%
Matched: 14.93%
Mismatched: 16.58%

3 training environments:
Multicondition: 13.26%
Mismatched: 14.02%

Ideas to explore further:

- why are some training environments better than others?
- use DNN to model and steer separation towards valid source spectra.

System fusion

Currently: use

- one method,
- one set of parameters and hyper-parameters.

Ideas to explore further:

- combine the results of multiple methods, e.g., by using their outputs as inputs to a "fusion" DNN

CHiME-2 SDR

9 NMFs with various dictionary size + 9 DNNs with various size and training cost

Method	SDR (dB)
Best individual NMF	5.12
Fused NMFs	8.15
Best individual DNN	9.01
Fused DNNs	9.31
Fused NMFs and DNNs	9.50

- fuse the hidden layers too?

Applications to spoken communication

Currently:

- learning-based separation mostly used in offline scenarios,
- DNN footprint smaller but still large, lack of control on sound quality.

Ideas to explore further:

- explore the impact of various training costs on sound quality,
- explore classical post-processing techniques (oversubtraction, thresholding, smoothing. . .),
- borrow DNN footprint reduction techniques from other fields until it becomes feasible in real time for, e.g., hearing aids!

Applications to human machine interfaces and spoken documents

Currently:

- concatenate enhancement, feature extraction and recognition networks:
  - multi-task training with enhancement and recognition losses,
  - joint training for noise-robust ASR using cross-entropy loss,
- use automatic speech recognition (ASR) to improve enhancement:
  - explicitly use ASR state posteriors as auxiliary input for enhancement;
  - iterate enhancement and recognition.

Ideas to explore further:

- integrate DNN based separation with other tasks: speaker ID, speaker diarization, language ID, keyword spotting. . .
- apply it to other spoken documents: movies, radio, TV. . .

References

Single-channel phase modeling

D. S. Williamson, Y. Wang, and D. L. Wang, "Complex ratio masking for monaural speech separation", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(3):483–492, 2016.

A. M. Sarroff, V. Shepardson, and M. A. Casey, "Learning representations using complex-valued nets", arXiV 1511.06351, 2015.

Data simulation

E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition", *Computer Speech and Language*, to appear.

J. Chen, Y. Wang, and D. L. Wang, "Noise perturbation improves supervised speech separation", in *Proc. LVA/ICA*, pp. 83–90, 2015.

S. Sivasankaran, A. A. Nugraha, E. Vincent, J. A. Morales Cordovilla, S. Dalmia, I. Illina, and A. Liutkus, "Robust ASR using neural network based speech enhancement and feature simulation", in *Proc. ASRU*, pp. 482–489, 2015.

References

Robustness to unseen, mismatched conditions

E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, “An analysis of environment, microphone and data simulation mismatches in robust speech recognition”, *Computer Speech and Language*, to appear.

E. M. Grais, M. U. Sen, and H. Erdogan, “Deep neural networks for single channel source separation”, in *Proc. ICASSP*, pp. 3734–3738, 2014.

M. Kim and P. Smaragdis, “Adaptive denoising autoencoders: a fine-tuning scheme to learn from test mixtures”, in *Proc. LVA/ICA*, pp. 100–107, 2015.

System fusion

J. Le Roux, S. Watanabe, and J. R. Hershey, “Ensemble learning for speech enhancement”, in *Proc. WASPAA*, pp. 1–4, 2013.

X. Jaureguiberry, E. Vincent, and G. Richard, “Fusion methods for speech enhancement and audio source separation”, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(7):1266–1279, 2016.