Deep Clustering: Discriminative embeddings for segmentation and separation

John Hershey
Zhuo Chen
Jonathan Le Roux
Shinji Watanabe

Problem to solve: general audio separation

- Goal:Analyze complex audio scene into its components
  - Different sound may be overlapping and partially obscure each other
  - Number of sound may be unknown
  - Sound types may be known or unknown
  - Multiple instances of a particular type may be present

- Many potential applications
  - Use separated components: enhancement, remix, karaoke, etc.
  - Recognition & detection: speech recognition, surveillance, etc.
  - Robots
    - robots need to handle the “cocktail-party problem”
    - need to be aware of sound in environment
    - no easy sensor-based solution for robots (e.g., close talking microphone)
    - humans can do this amazingly well

- More important goal: understand how human brain work

Why is general audio separation difficult?

- Incredible variety of sound types
  - Human voice: speech, singing…
  - Music: many kinds of instruments (strings, woodwind, percussion)
  - Natural sound: animals, environmental…
  - Man-made sounds: mechanical, sirens…
  - Countless unseen novel sounds
- The “modeling problem”
  - Difficult to make models for each type of sound
  - Difficult to make one big model that applies to any sound type
  - Sounds obscure each other in a state dependent way
    - Which sound dominates a particular part of the spectrum depends on the states of all sounds.
    - Knowing which sound dominates makes it easy to determine states
    - Knowing the states makes it easy to determine which sound dominates
    - Chicken and egg problem: ***the joint problem is intractable!***

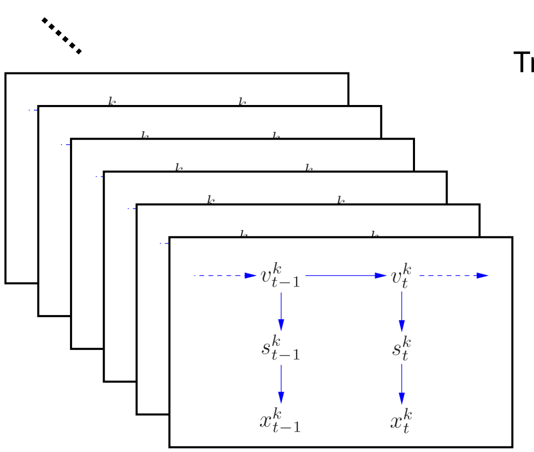
Previous attempts

- CASA (1990s~early 2000s)
  - Segment spectrogram based on Gestalt “grouping cues”
  - Usually no explicit model of the sources
  - Advantage: potentially flexible generalization
  - Disadvantage: rule based, difficult to model “top-down” constraints.

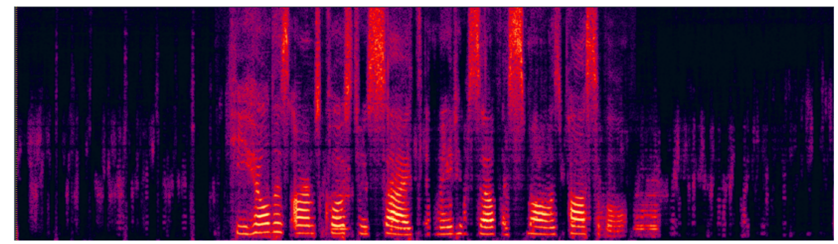
- Model based systems (early 2000s ~ now)
  - Examples: non-negative matrix factorization, factorial hidden Markov models
  - Model assumptions hardly ever match data
  - Inference is intractable, difficult to discriminatively train

- Neural networks
  - Work well for known target source type, but difficult to apply to many types
  - Problem of structuring the output labels in the case of multiple instances of the same type
  - Unclear how to handle novel sound types or classes. No instances seen during training
  - Some special type of adaptation is needed

Model-based Source Separation



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



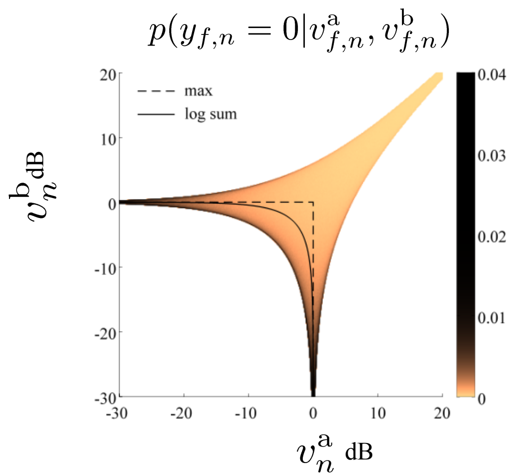
He held his arms close to…

Signal Models

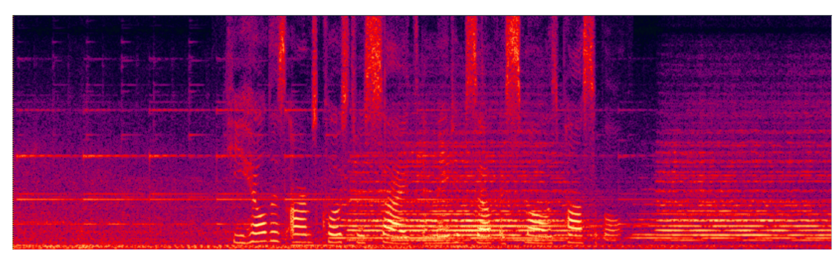
Inference

Predictions

Interaction Models

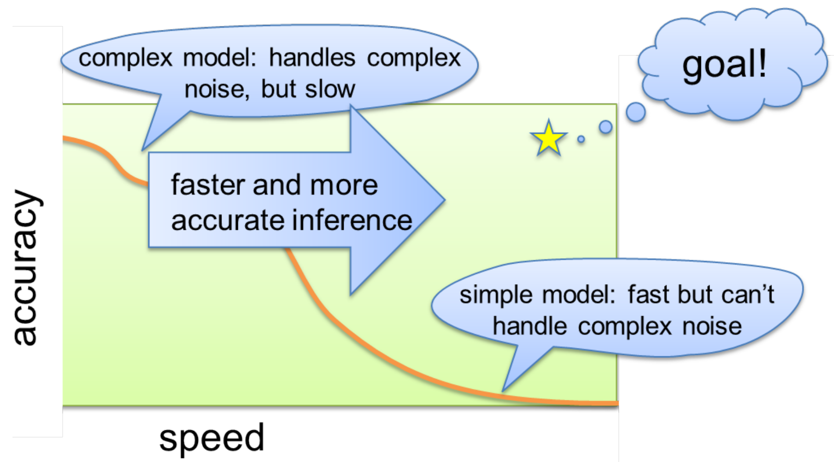


Data



Problems of generative model

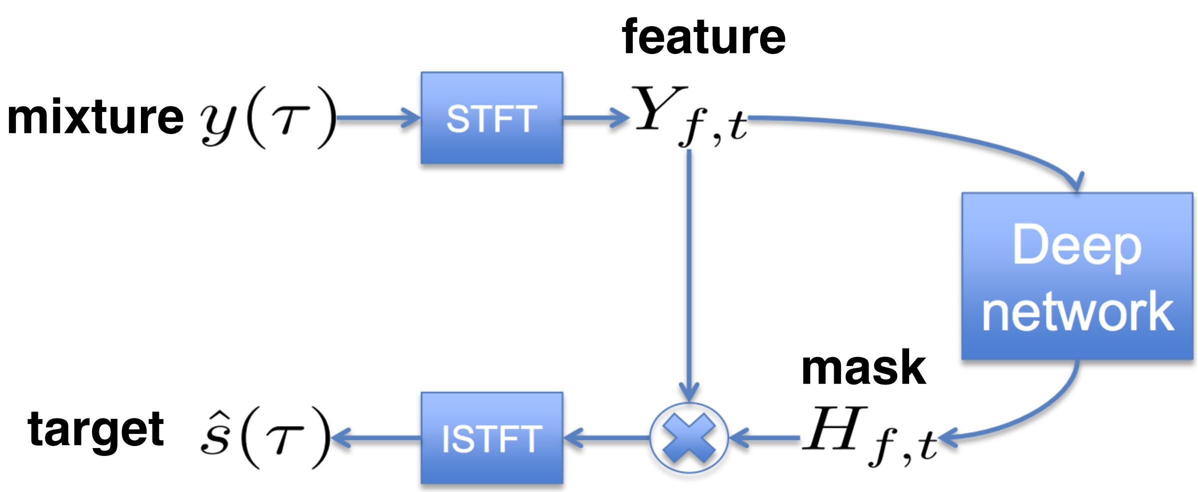
- Trade-offs between speed and accuracy
- Limitation to separate similar classes
- More broadly, no way the brain is doing like this



Neural network works well for some tasks in source separation

- State-of-the-art performance in across-type separation
  - Speech enhancement: Speech vs. Noise
  - Singing music separation: Singing vs. Music

Noisy **Enhanced**

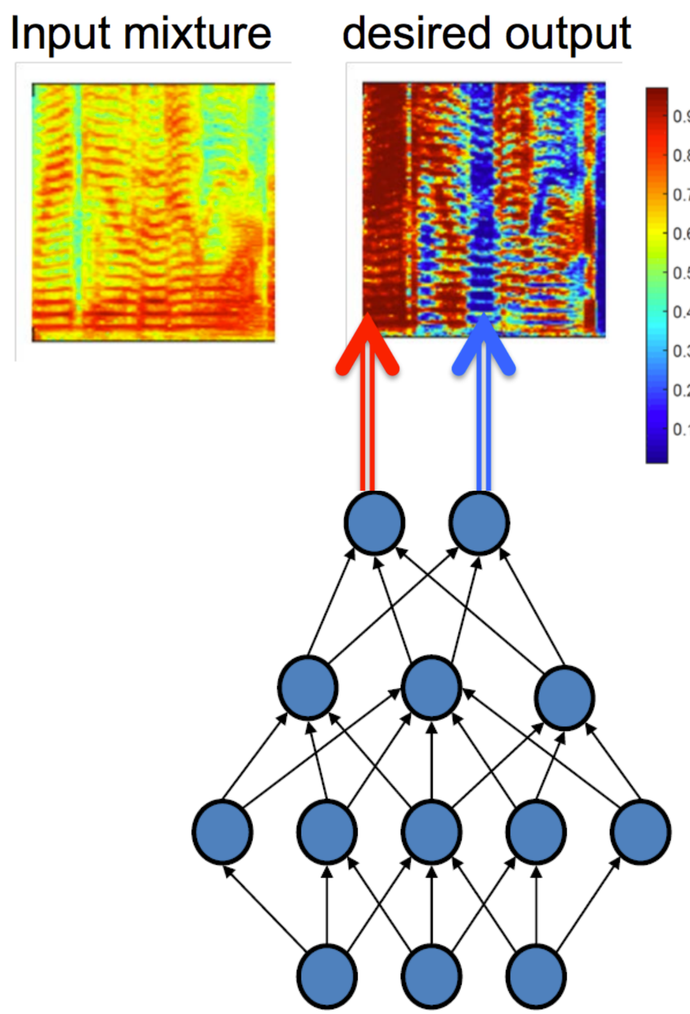


- Auto-encoder style Objective function: $L = \sum \|H_{f,t} - F(Y_{f,t})\|^2$

However,

- Limitation in scaling up for multiple sources
  - When more than two sources, which target to use?
  - How to deal with unknown number of sources?

- Output permutation problem
  - When the sources are similar
  - e.g. when separating mixture of speech from two speakers, all parts are speech, then which slot should identify which speaker?



Separating mixed speakers—a slightly harder problem

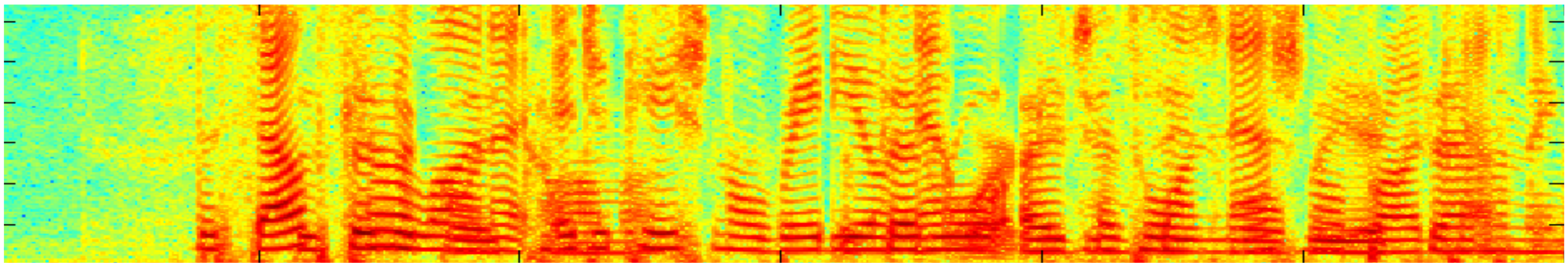
- Mixture of speech from two speakers
  - Sources have similar characteristics
  - Interested in all sources
  - Simplest example of a cocktail party problem

- Investigated several ways of training neural network

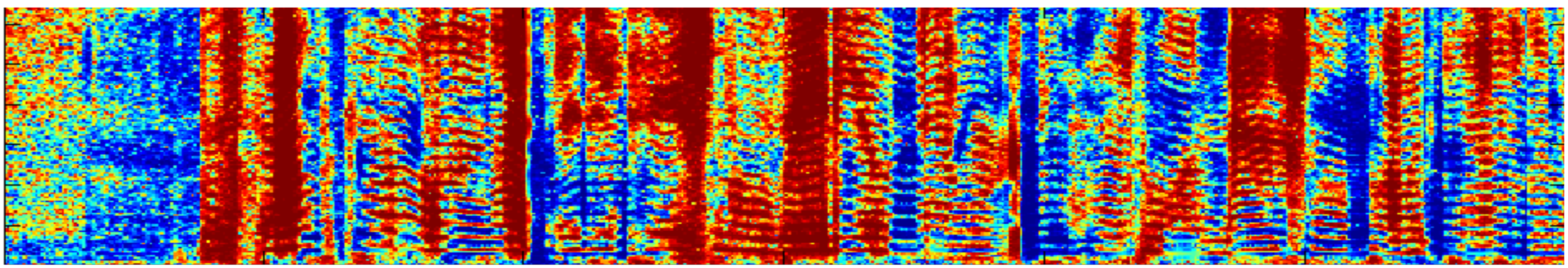
  On small chunks of signal:
  - Use oracle permutation as clue
    - Train the network by back-propagating difference with best-matching speaker
  - Use strongest amplitude as clue
    - Train the network to separate the strongest source

The neural network failed to separate speakers

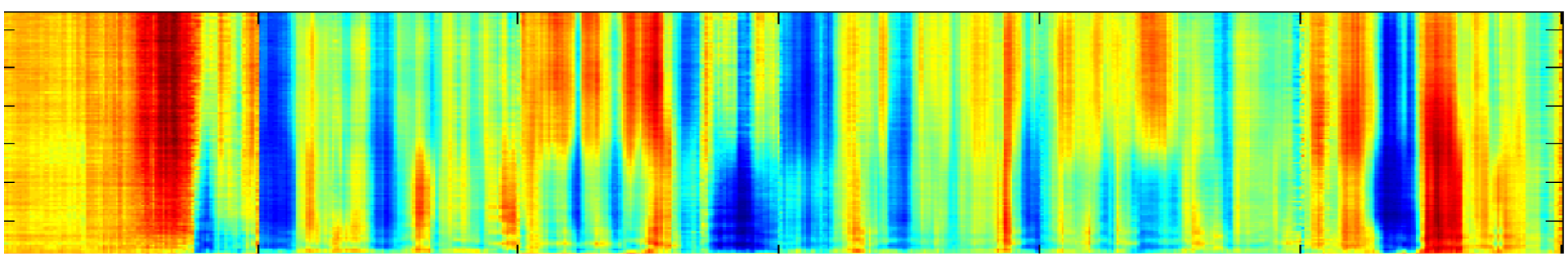
Input mixture



Oracle output



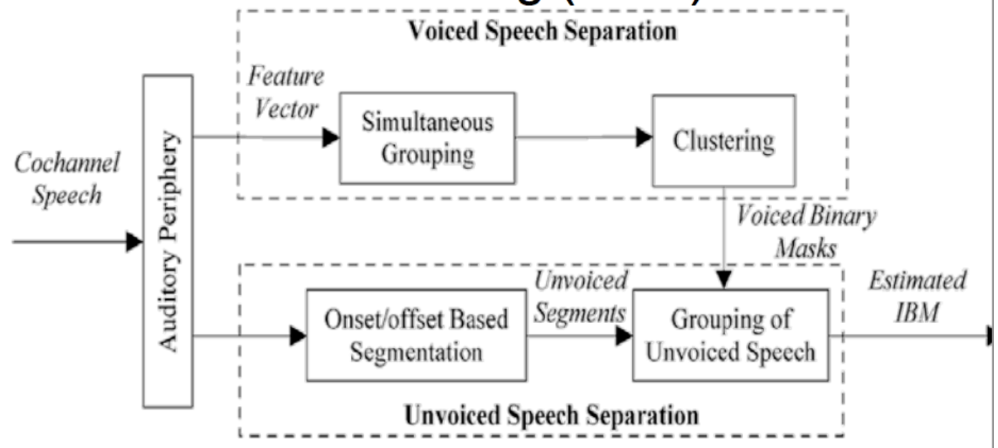
DNN output



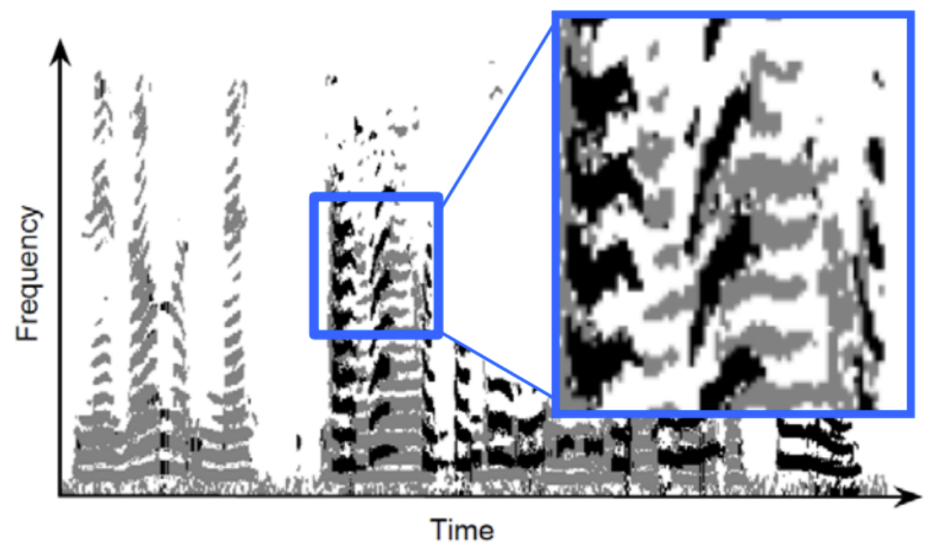
Clustering Approaches to Separation

- Clustering approaches handle the permutation problem
- CASA approaches cluster based on hand-crafted similarity features:
- Proximity in time, frequency
  - – Common amplitude modulation
  - – Common frequency modulation
  - – Harmonicity using pitch tracking
- Spectral clustering was used to combine CASA features via multiple kernel learning
- Catch-22 with features: whole patch of context needed, but this overlaps multiple sources

Hu & Wang (2013)



Bach & Jordan (2006)



From class-based to partition-based objective

- Class-based objective: estimate the class of an object
  - Learn from training class labels
  - Need to know object class labels
  - Supervised model
  - E.g. :

model

$$C(\theta) = |V - Y|_{\mathrm{F}}^2$$

target

- Partition-based objective: estimate what belongs together
  - Learn from labels of partitions
  - No need to know object class labels
  - Semi-supervised model
  - E.g. :

model

$$C(\theta) = \sum_{Y_j = Y_i} |V_i - V_j|_{\mathrm{F}}^2$$

target

Learning the affinity

- One could thus think of directly estimating affinities using some model:

$$\hat{A}_i = g_\theta(X_i)$$

- For example, by minimizing the objective:

$$\mathcal{L}(\theta) = |A - \hat{A}|_F^2$$

- But, affinity matrices are large

- Factoring them can be time consuming with complexity $\mathcal{O}(N^3)$

- Current speedup methods for spectral clustering such as Nyström method use low-rank approximation to $\hat{A}_i$

- If the rank of the approximation is $K < N$, then we can compute the eigenvectors of $\hat{A}_i$ in $\mathcal{O}(K^2 N)$ -- Much faster!

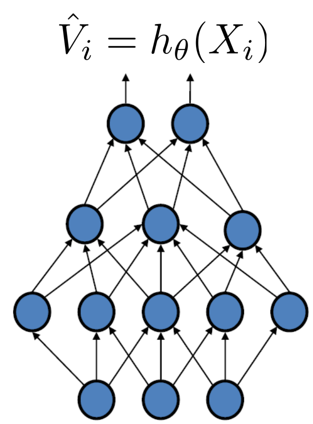
Learning the affinity

- Instead of approximating a high-rank affinity matrix, we train the model to produce a low-rank one, by construction:

$$\hat{A} = VV^T$$

where we estimate $V_i = h_\theta(X_i)$, a K-dimensional embedding

- We propose to use deep networks
  - Deep networks have recently made amazing advances in speech recognition
  - Offer a very flexible way of learning good intermediate representations
  - Can be trained straightforwardly using stochastic gradient descent on



Affinity-based objective function

$$C(\theta) = |VV^T - YY^T|_\mathrm{F}^2 = \sum_{i,j:y_i=y_j} (\langle v_i, v_j \rangle - 1)^2 + \sum_{i,j:y_i \neq y_j} (\langle v_i, v_j \rangle - 0)^2,$$

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

$$s.t \quad \sum_k |v_{i,k}|^2 = 1, \quad \forall i$$

- High-dimensional embedding
- First term directly related with K-means objective
- Second term “spreads” all the data points from each other

where:

- $V \in \mathcal{R}^{N \times K}$: the output of the network, K-dimensional embedding for each time-frequency bin.
- $Y \in \mathcal{R}^{N \times C}$: the class indicator vector for each time-frequency bin

Avoiding the N x N affinity matrix

- The number of samples N is orders of magnitude larger than the embedding dimension K
  - e.g., for a 10s audio clip, N=129000 T-F bins (256 fft, 10ms hop)
    Affinity matrix has 17 billion entries!

- Low rank structure of $VV^\top$ can avoid saving full affinity matrix
  - When computing the objective function:

$$C = |VV^T - YY^T|_\mathrm{F}^2 = |V^TV|_\mathrm{F}^2 - 2|V^TY|_\mathrm{F}^2 + |Y^TY|_\mathrm{F}^2$$

  - When computing the derivative:

$$\frac{\partial C}{\partial V^T} = 4V(V^TV) - 4Y(Y^TV)$$

Evaluation on speaker separation task

- Network
  - Two BLSTM layers neural network with various layer sizes
- Data
  - Training data
    - 30 h of mixtures of 2 speakers randomly sampled from 103 speakers in WSJ dataset
    - Mixing SNR from -5dB to 5dB
  - Evaluation data
    - Closed speaker set: 10 h of mixtures of other speech from the same 103 speakers
    - Open speaker set: 5 h of mixtures from 16 other speakers
- Baseline methods
  - Closed speaker experiments: Oracle dictionary NMF
  - CASA
  - BLSTM auto encoder with different permutation strategies

Significantly better than the baseline

Table 1: SDR improvements (dB) for different separation methods

method	**CC**	**OC**
oracle NMF	5.1	-
CASA	2.9	3.1
DC oracle k-means	6.5	6.5
DC global k-means	5.9	5.8
BLSTM stronger	1.3	1.2
BLSTM permute	1.3	1.3
BLSTM permute*	1.4	1.2

Table 2: SDR improvements (dB) for different embedding dimensions K and activation functions

	CC		**OC**	
model	DC oracle	DC global	DC oracle	DC global
$K = 5$	-0.8	-1.0	-0.7	-1.1
$K = 10$	5.2	4.5	5.3	4.6
$K = 20$	6.3	5.6	6.4	5.7
$K = 40$	6.5	5.9	6.5	5.8
$K = 60$	6.0	5.2	6.1	5.3
$K = 40$ logistic	6.6	5.9	6.6	6.0

Audio example

- Different gender mixture

Oracle NMF results

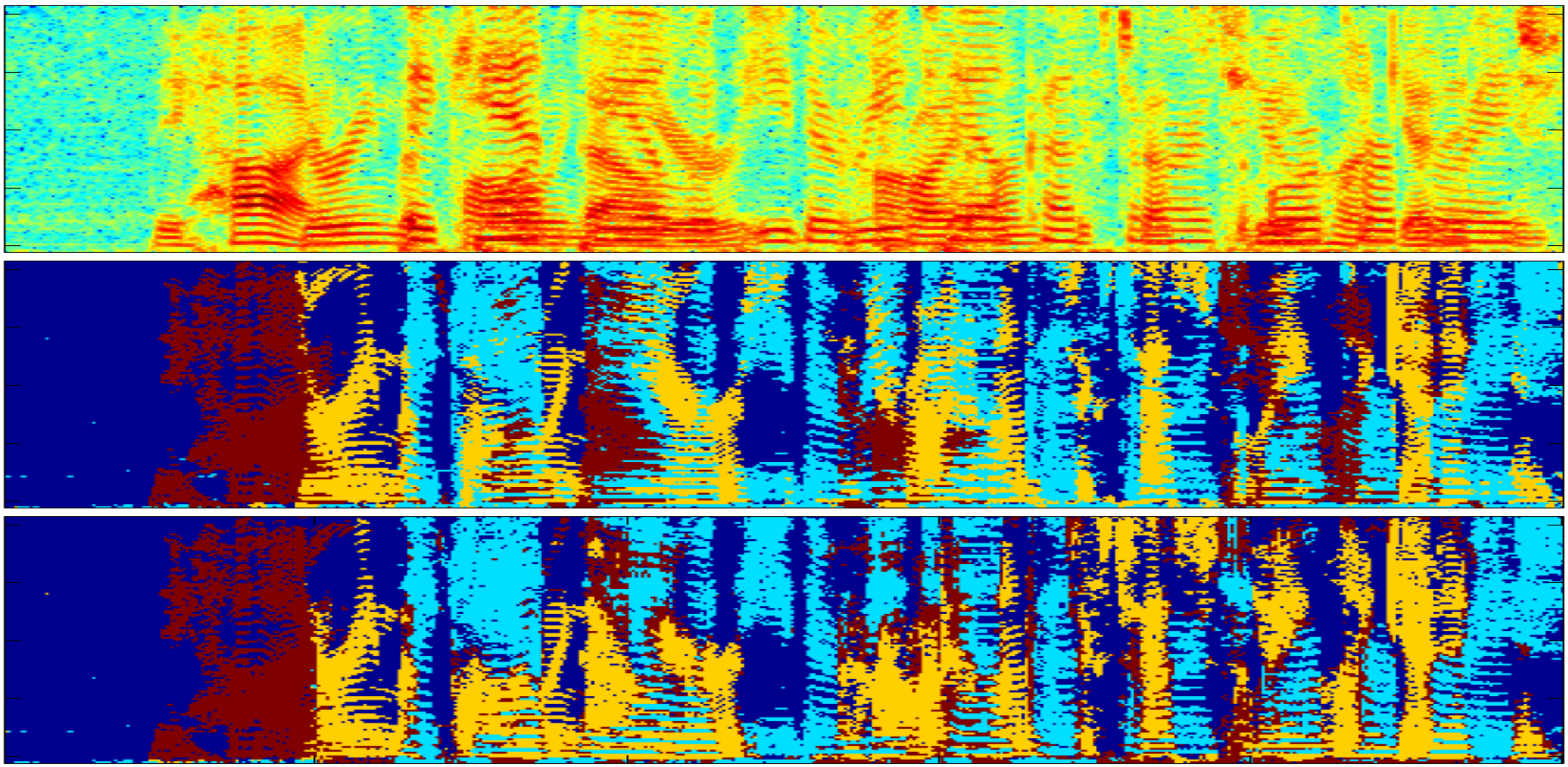
Deep clustering result

- Same gender mixture

Oracle NMF results

Deep clustering results

The same net works on three speakers mixtures



- The network was trained with two speaker mixtures only!

Separation three-speaker mixture

- Data
  - Training data
    - 10 h of mixtures of 3 certain speakers sampled from WSJ dataset
    - Mixing SNR from -5dB to 5dB
  - Evaluation data
    - 4 h of mixtures of other speech from the same speakers

Table 3: SDR improvement (dB) for mixtures of three speakers. Left: three-speaker separation using DC network trained on two-speaker mixtures. Right: separation of three known speakers.

method	**MS-CC**	**MS-OC**
oracle NMF	4.4	-
DC oracle	3.5	2.8
DC global	2.7	2.2

method	**3S-CC**
oracle NMF	4.5
DC oracle	7.0
DC global	6.9
BLSTM stack	6.8

Single speaker separation

- Data
  - Training data
    - 10 h of mixtures of one speaker sampled from 103 speakers in WSJ dataset
    - Adapted data: 10 h of one certain speaker
    - Mixing SNR from -5dB to 5dB
  - Evaluation data
    - Closed speaker: 5 h of mixtures of other speech from the same 103 speaker
    - Closed speaker: 3 h of mixtures of other 16 speaker
    - Adapted data: 10 h of other speech of one certain speaker

Table 2: SDR improvements (dB) for female mixtures

method	CC	OC
DC oracle k-means	0.18	-1.91
DC global k-means	-0.10	-2.54
Adapted DC oracle k-means	-1.22	-2.03
Adapted DC global k-means	-1.19	-2.73
fvf trained DC oracle k-means	0.66	-0.18
fvf trained DC global k-means	-0.24	-2.12
Adapted fvf trained DC oracle k-means	0.80	-0.79
Adapted fvf trained DC global k-means	0.52	-1.54
Certain spk DC oracle k-means	2.39	-
Certain spk DC global k-means	0.97	-

Table 3: SDR improvements (dB)

method	CC	OC
DC oracle k-means	0.18	-1.67
DC global k-means	-0.02	-2.07
CASA	-2.6	-2.7
Adapted DC oracle k-means	-1.04	-1.6
Adapted DC global k-means	-1.05	-2.58

male **female**

mixed

source 1

source 2

Possible extensions

- Different network options
  - Convolutional architecture
  - Multi-task learning
  - Different pre-training
- Joint training through the clustering
  - Combining with deep unfolding
  - Compute gradient through the spectral clustering
- Different tasks
  - General audio separation

Thanks a lot!