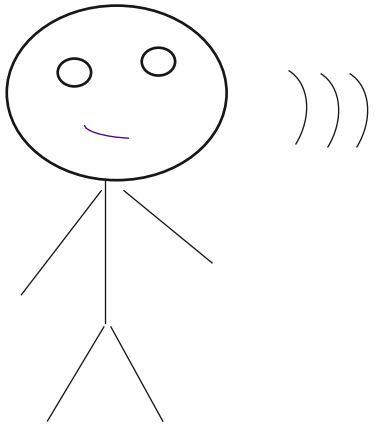
Low-latency Deep Clustering for Speech Separation

*Shanshan Wang, Gaurav Naithani, Tuomas Virtanen

Audio Research Group

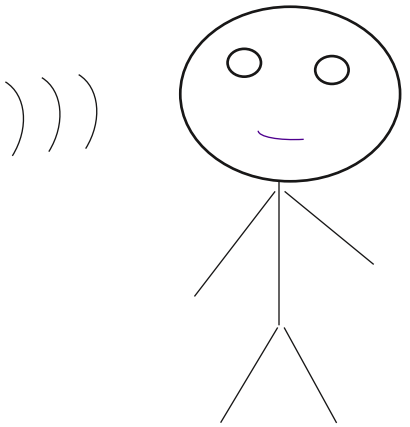
Tampere University

2-speaker mixture separation



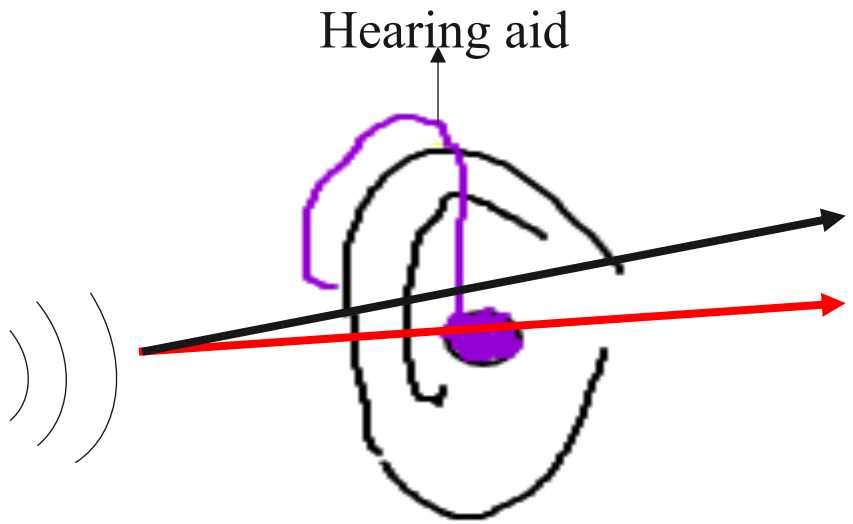
Hello, ICASSP 2019!

Good morning!



Why low latency?

- Low latency is important in many scenarios like telephone calls, live concert performance.
- Especially, for **hearing aids.**



Why low latency?

- Currently, most state of the art of speech separation systems are offline.

- A **low-latency** version is strongly needed!

Main Contribution

- A **low algorithmic latency** adaptation of the deep clustering approach for speaker-independent speech separation.

- Precisely, by investigating

  - 1. network topology that allows online processing

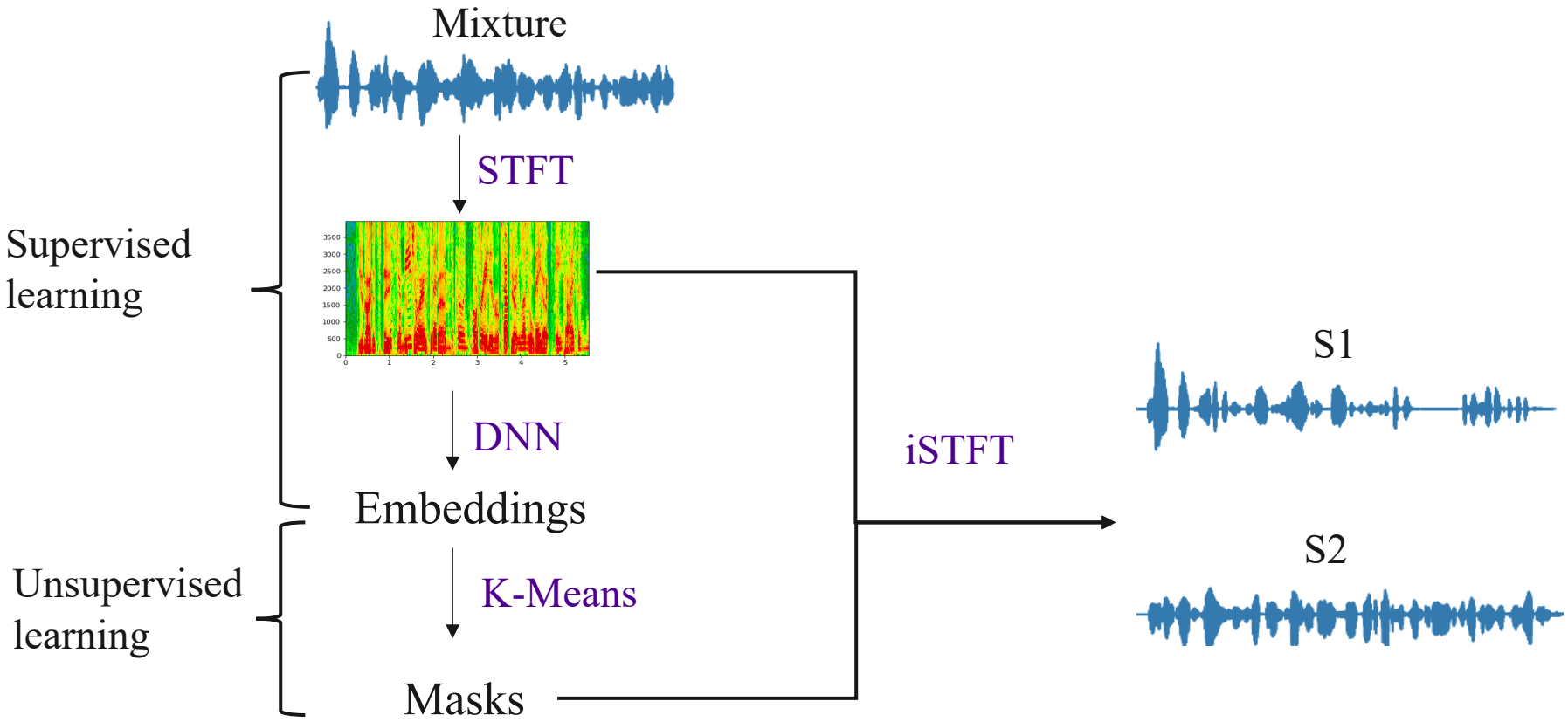
  - 2. time-frequency representation that allows low latency

  - 3. how to obtain speaker models (cluster of embeddings) in short time

Outline

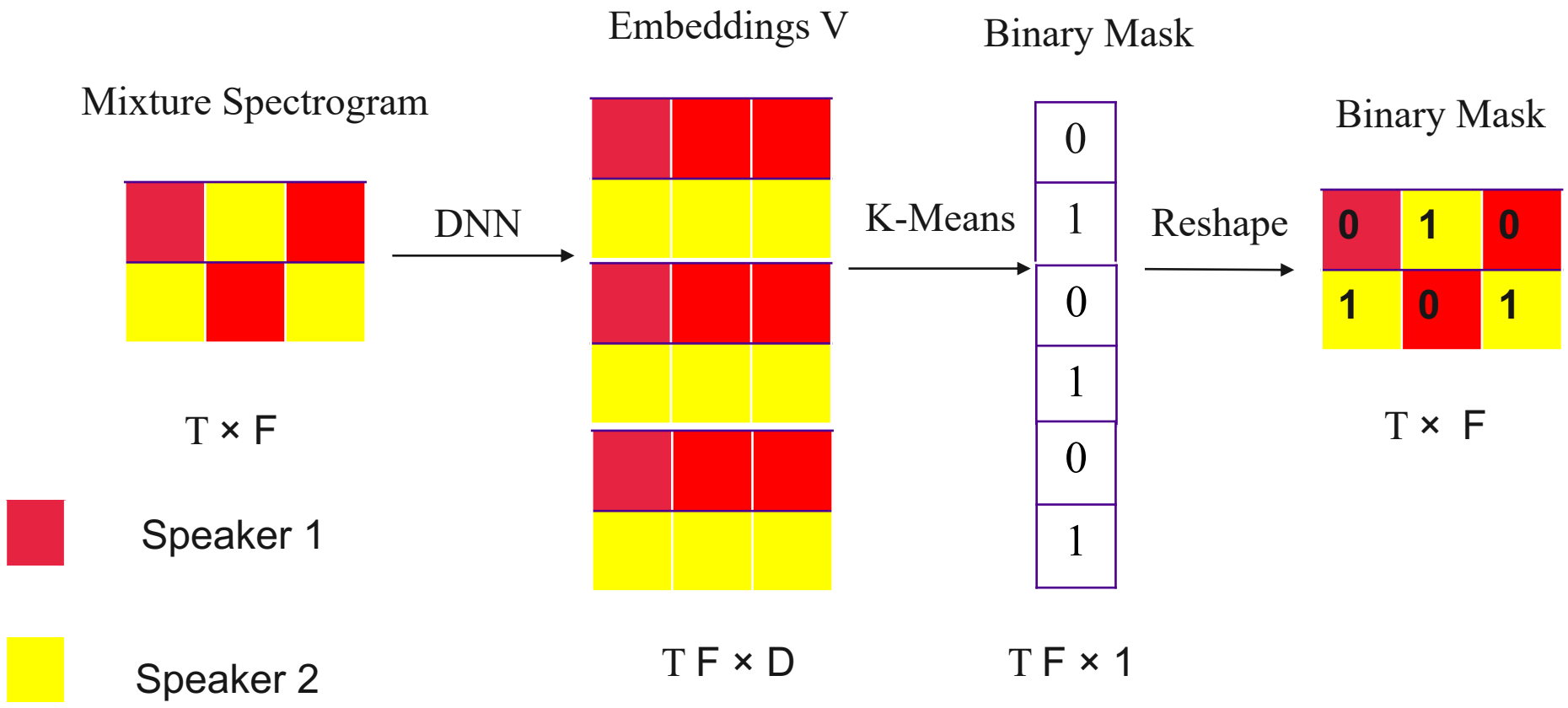
- **Offline Deep Clustering**
- Low-latency Deep Clustering
- Evaluation

Offline Deep Clustering



J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 31–35.

Offline Deep Clustering



Offline Deep Clustering

The neural network is trained to minimize the difference between the estimated affinity matrix $\boldsymbol{VV}^T$ and the target binary mask affinity matrix $\boldsymbol{YY}^T$.

$$\boldsymbol{L} = \|\boldsymbol{VV}^T - \boldsymbol{YY}^T\|_F^2,$$ where F denotes the Frobenius norm of the matrix

$\boldsymbol{V} \in \mathbb{R}^{TF \times D}$

		...	
		...	
		...	
		...	
		...	
		...	
		...	

$\boldsymbol{Y} \in \mathbb{R}^{TF \times C}, c = 2$

0	1
0	1
1 ...	0 ...
0	1
1	0
0	1

Pros: Speaker independent

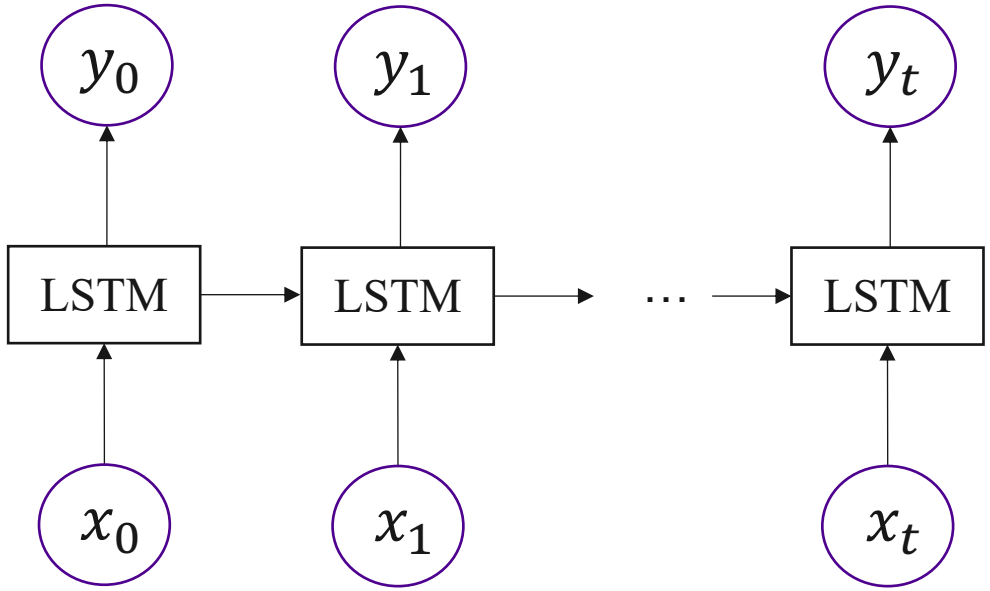
Cons: offline

Outline

- Deep Clustering
- **Low-latency Deep Clustering**
- Evaluation

Low-latency Deep Clustering

Network topology that allows online processing – **LSTM network**

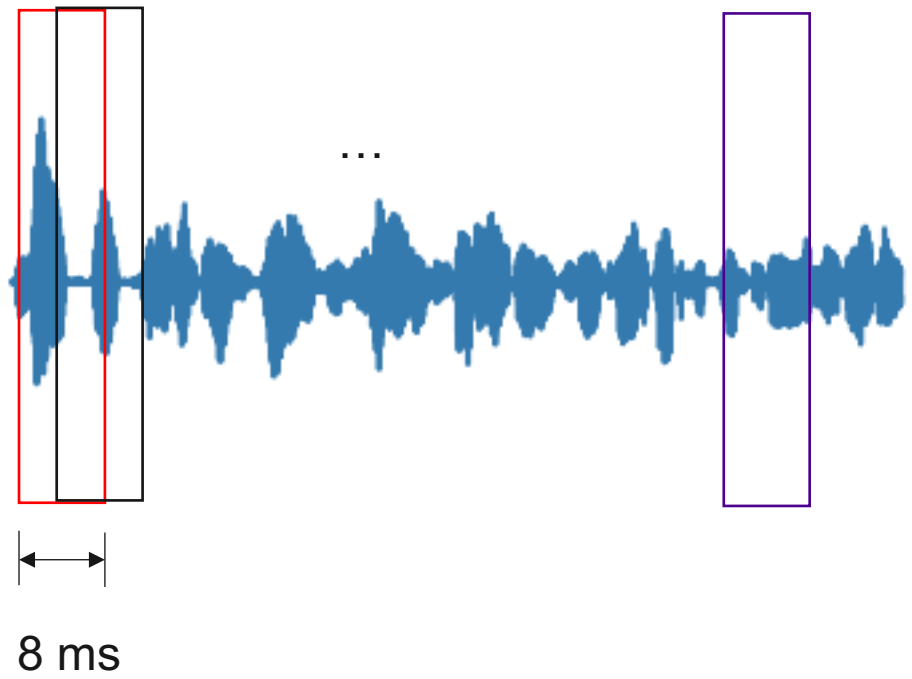


$x_i, \quad i \in 0,1, \ldots t$ Input

$y_i, \quad i \in 0,1, \ldots t$ Output

Low-latency Deep Clustering

Time-frequency representation that allows low latency – using **8 ms** window length

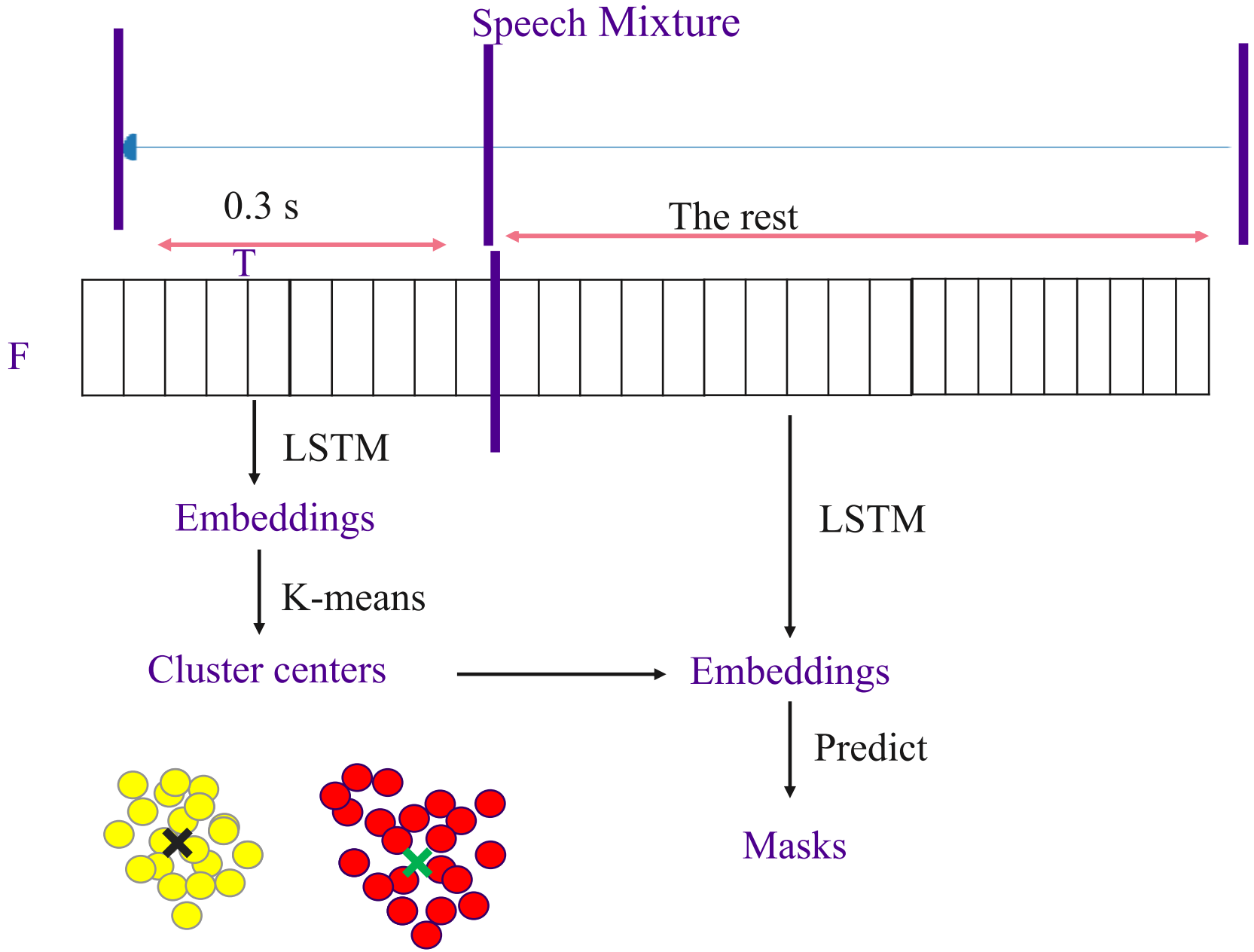


Low-latency Deep Clustering

Obtain speaker models (cluster of embeddings) in short time **- buffer**

Using only a certain length in the beginning of the mixture, to get the cluster centers,

and those centers can be used to predict the masks for the rest of the mixture.


Speech Mixture
0.3 s
The rest
T
F
LSTM
Embeddings
K-means
Cluster centers
LSTM
Embeddings
Predict
Masks

Outline

- Deep Clustering
- Low-latency Deep Clustering
- **Evaluation**

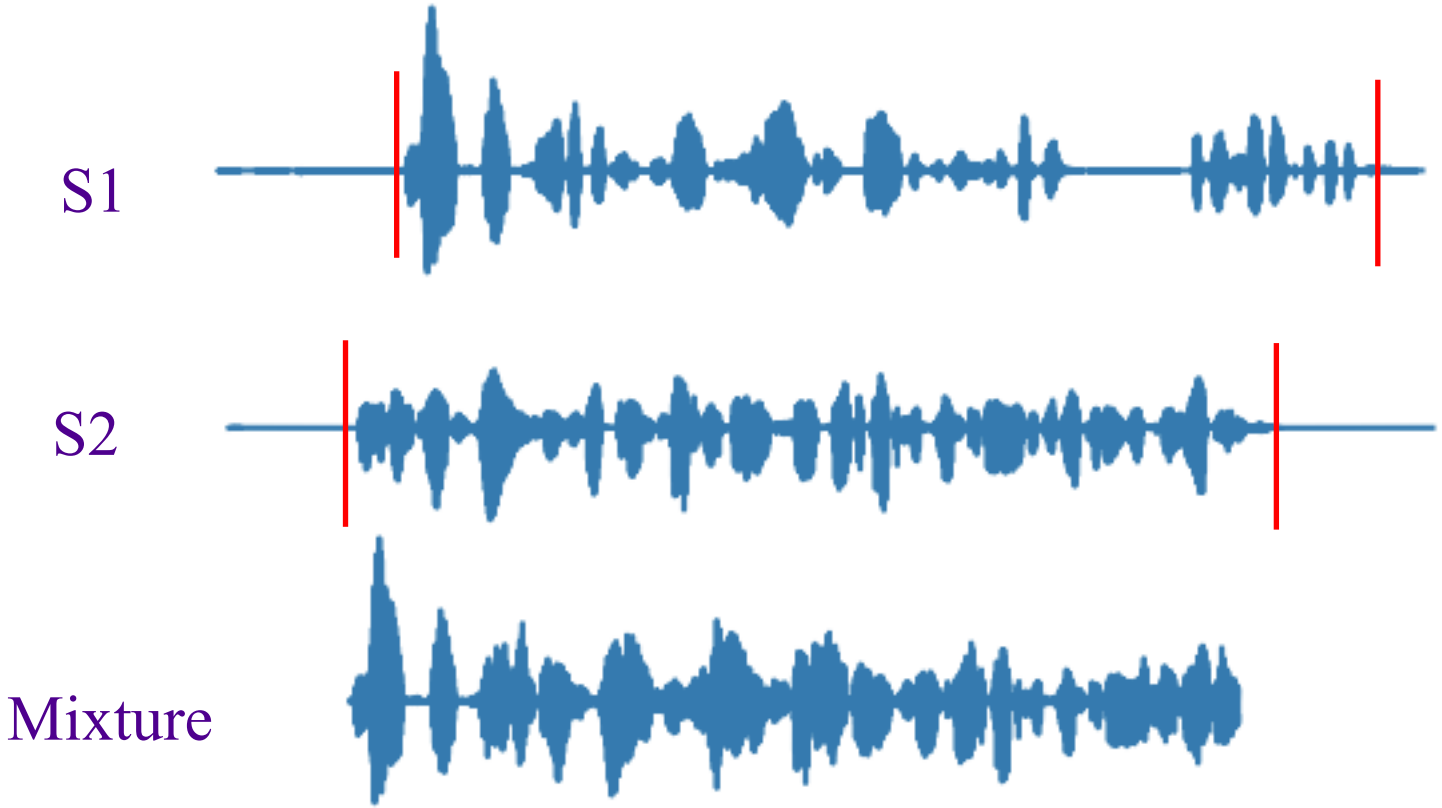
Low-latency DC Evaluation

Data Set: 2-speaker mixtures from Wall Street Journal corpus (wsj0)

	Training data	CV data	Test data
Duration	33 hours	8 hours	5 hours
# of speakers	110	-	18

Low-latency DC Evaluation

Leading and trailing silence are trimmed from test audio signals.



Ensuring that **both speakers are active** in the buffer.

Low-latency DC Evaluation

In order to **keep the test material the same** for different buffer length, test and cluster utterance are taken from different mixtures of the same speaker pair.

Cluster utterance

Test utterance

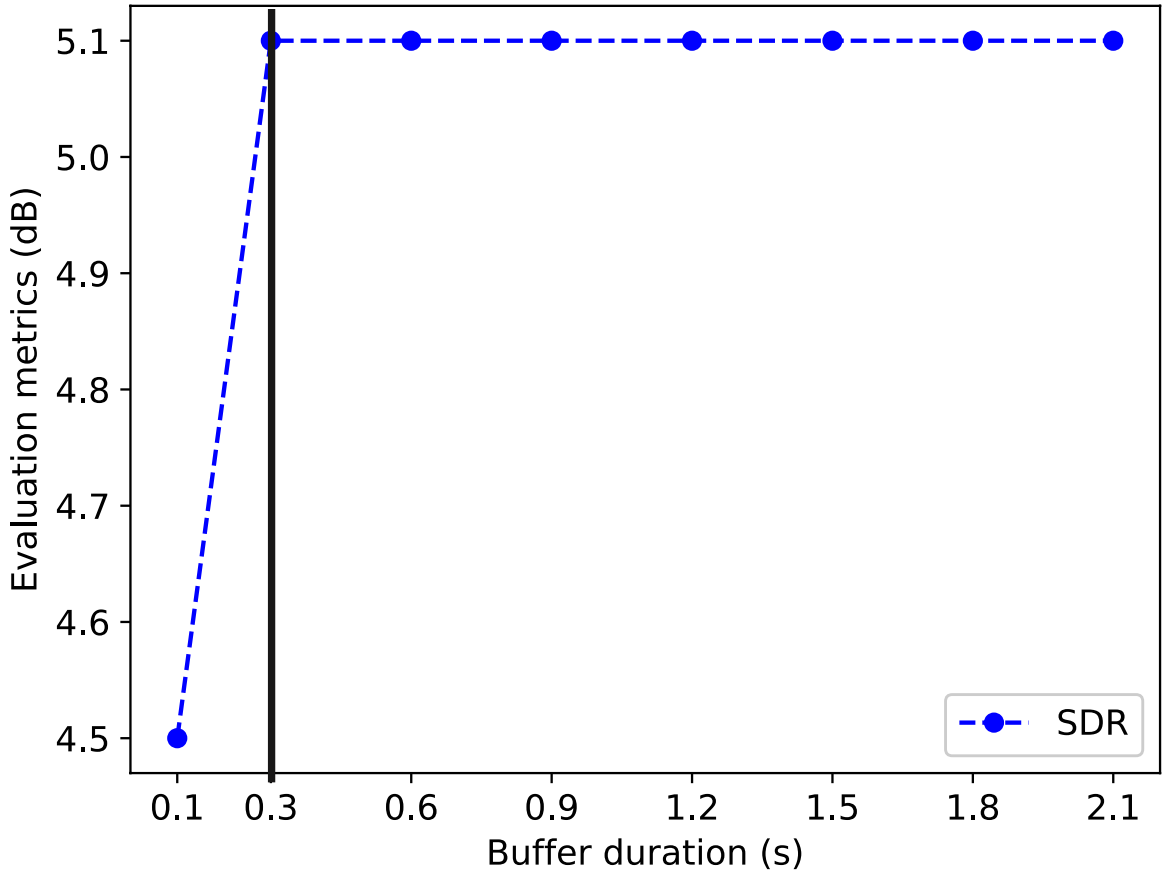
Networks and Parameters

	Offline DC	Low-latency DC
Window length	32 ms	8 ms
Hop length	8 ms	4 ms
Sequence length	100	200
Network	BLSTM	LSTM
Number of layers	4	
Number of LSTM units	600	
Embedding dimension	40	

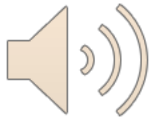
Offline DC & Online DC result

		Window length	SDR (dB)
Offline DC	*BLSTM	32 ms	7.9
	LSTM	32 ms	6.9
	LSTM	8 ms	5.8
Online DC	LSTM (0.3s buffer)	8 ms	5.1

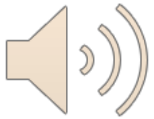
The Effect of Buffer Length



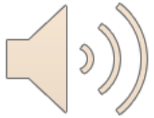
Audio Samples



Mixture (female + male)

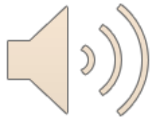


Estimated speaker 1



Estimated speaker 2

Audio Samples



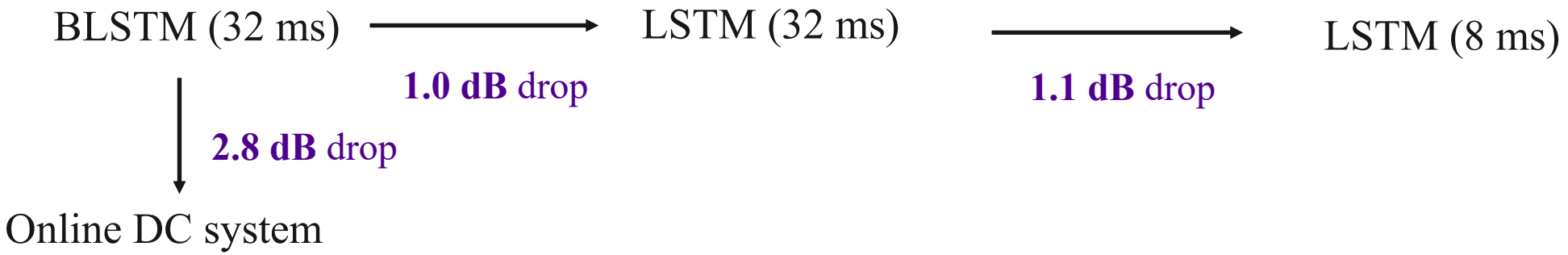
Mixture (male + male)

Estimated speaker 1

Estimated speaker 2

Conclusion

A **low algorithmic latency** adaptation of the deep clustering approach for speaker-independent speech separation.



More importantly, we found that even with **0.3 s** buffer duration, it is sufficient to estimate reasonable clusters for separation.