Listening to Multi-talker Conversations

Modular and End-to-end Perspectives

Desh Raj

August 18, 2023

Motivation

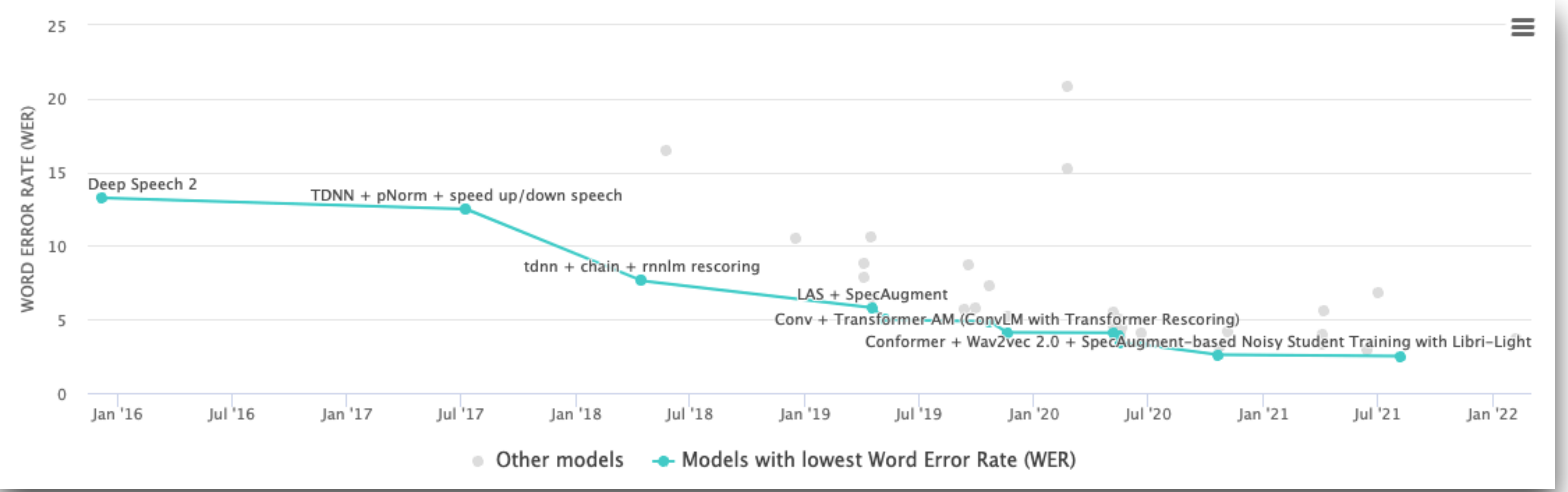
OCTOBER 20, 2020

AI outperforms humans in speech recognition

by Monika Landgraf, Karlsruhe Institute of Technology

Microsoft claims new speech recognition record, achieving a super-human 5.1% error rate

BY TODD BISHOP on August 20, 2017 at 7:44 pm



https://paperswithcode.com/sota/speech-recognition-on-librispeech-test-other

Motivation

Single-user applications

Smart Assistants

Language Learning

Customer Service

Voice-based Search

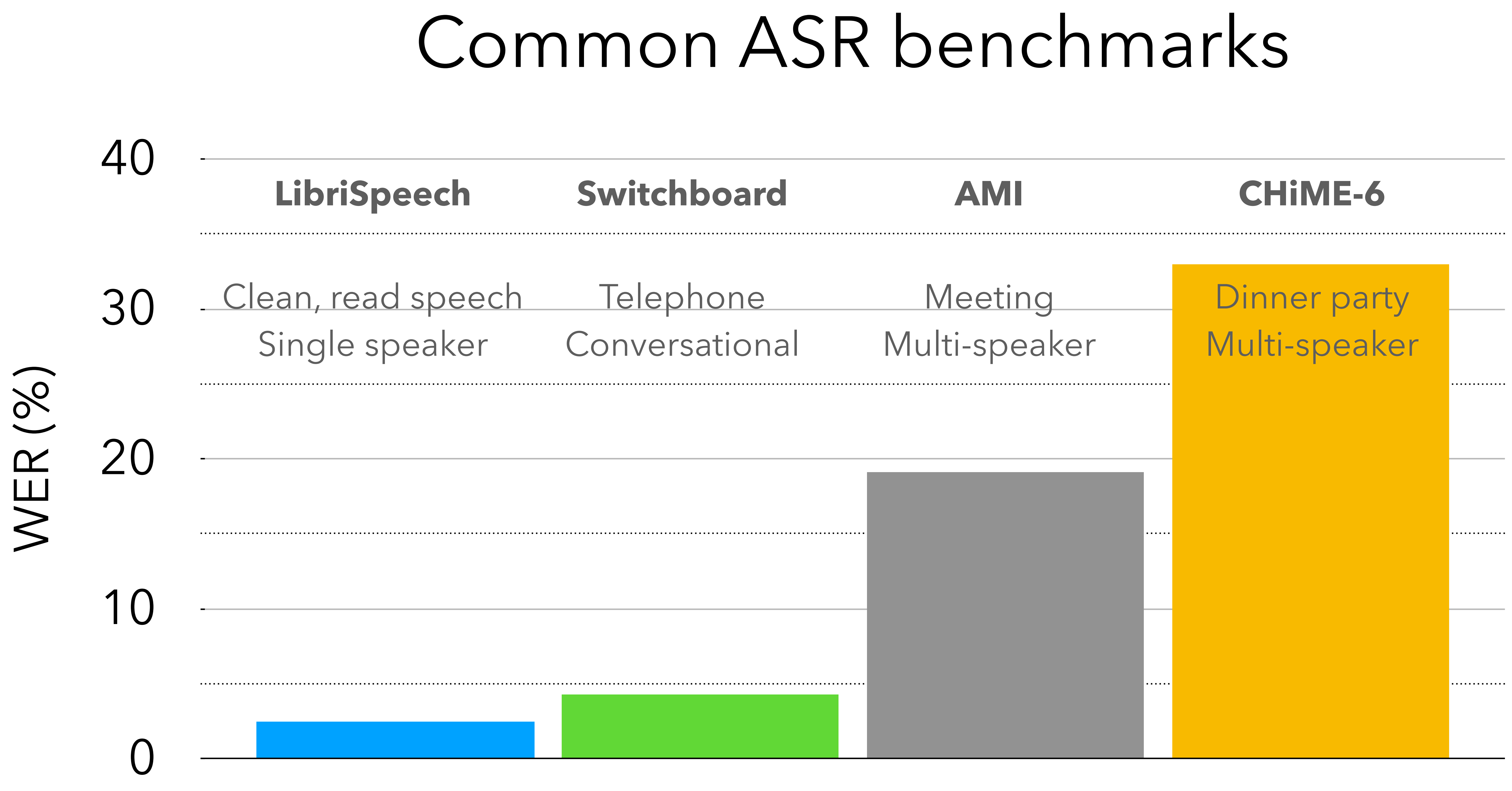
Multi-user applications

Meeting summaries

Collaborative Learning

Cocktail-party Problem

Motivation



What changed?

- Conversational speech
- Far-field audio: noise and reverberation
- Overlapping speakers

Problem Statement

Multi-talker speaker-attributed ASR

- **Input:** long unsegmented (possibly multi-channel) recording containing multiple speakers.
- **Output:**
    - Transcription of the recording (speech recognition)
    - Speaker attribution (diarization)
    - Additional constraints: streaming, i.e., real-time transcription
- We specifically look at “meetings”: LibriCSS, AMI, AliMeeting

Today's talk

"Modular" and "end-to-end" perspectives

1. Overlap-aware speaker diarization — 10 minutes
2. Target-speaker methods — 20 minutes
   (i) Extraction using guided source separation
   (ii) Recognition using neural transducers
3. Streaming Unmixing and Recognition Transducer (SURT) — 25 minutes

Overlap-aware Speaker Diarization

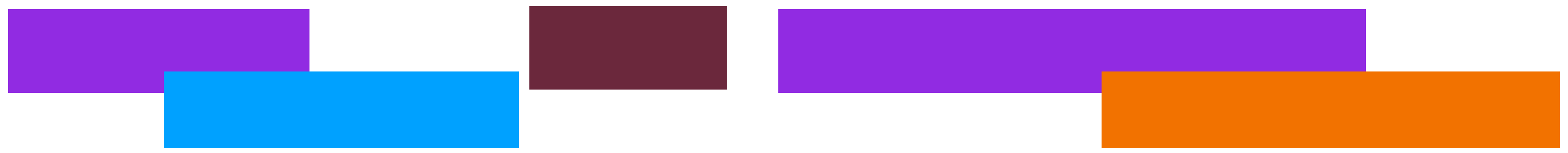
Background

What is speaker diarization?

Task of “who spoke when”

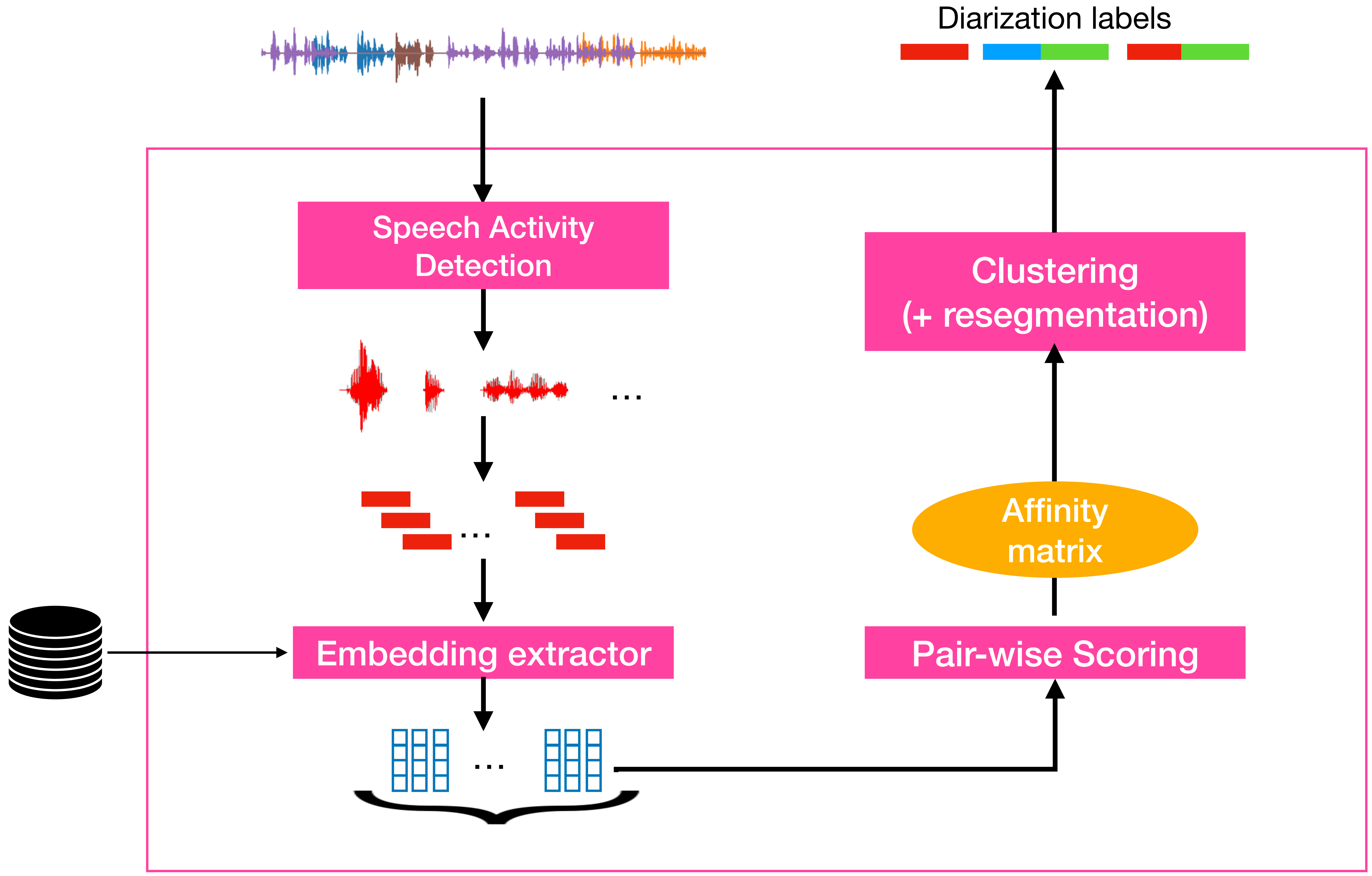
Input: ***recording containing multiple speakers***

Output: ***homogeneous speaker segments***



Clustering-based diarization

Overview of the process



Clustering paradigm assumes single-speaker segments

So overlapping speakers are completely ignored!

*"Roughly **8% of the absolute error** in our systems was from overlapping speech … it will likely require a **complete rethinking of the diarization process** … This is an important direction, but could not be addressed …"*
- JHU team (2018)

*"Given the current performance of the systems, the **overlapped speech gains more relevance** … **more than 50% of the DER** in our best systems … has to be addressed in the future …"*
- BUT team (2019)

Overlap-aware diarization

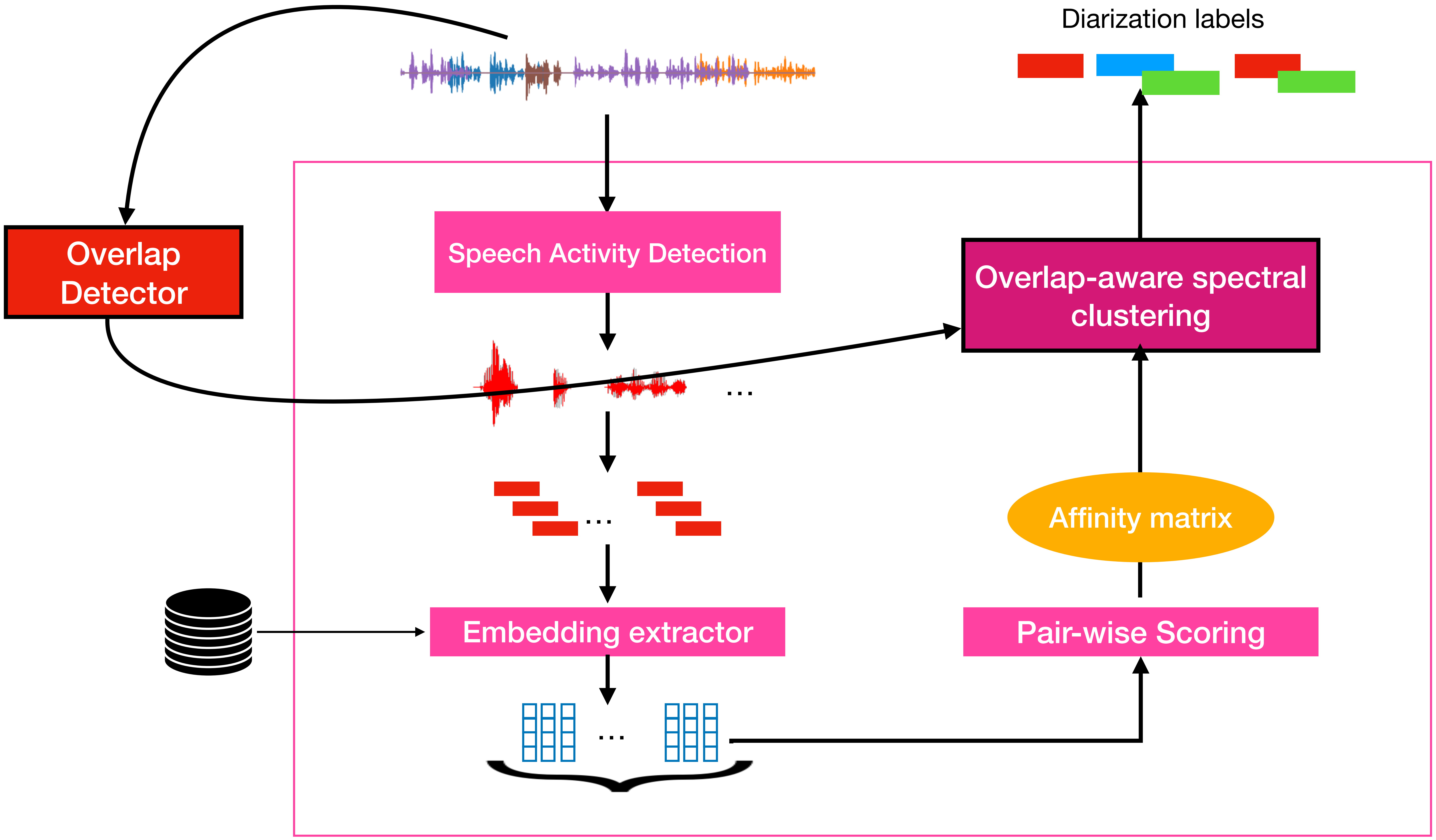
MULTI-CLASS SPECTRAL CLUSTERING WITH OVERLAPS FOR SPEAKER DIARIZATION

Desh Raj[1], *Zili Huang*[1], *Sanjeev Khudanpur*[1,2]

[1]Center for Language and Speech Processing & [2]Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD 21218, USA.

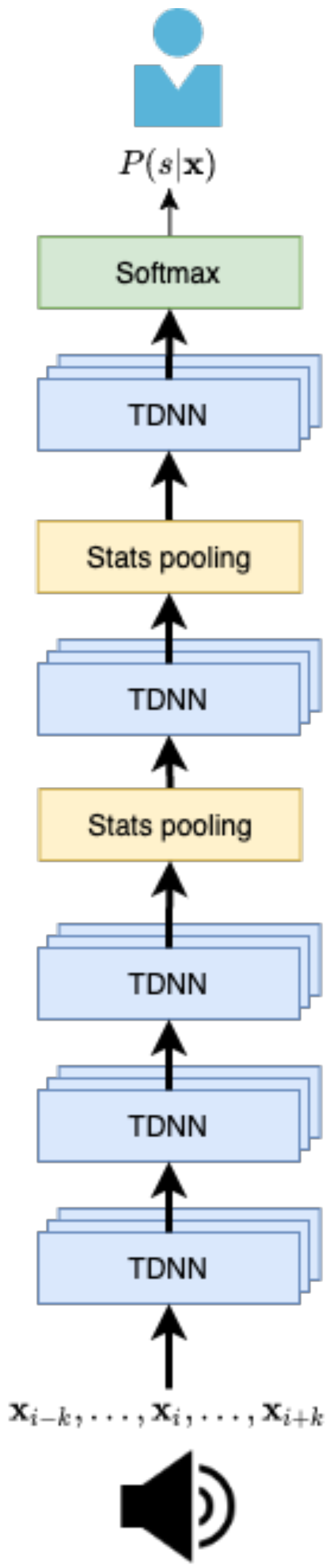
Published at IEEE SLT 2021

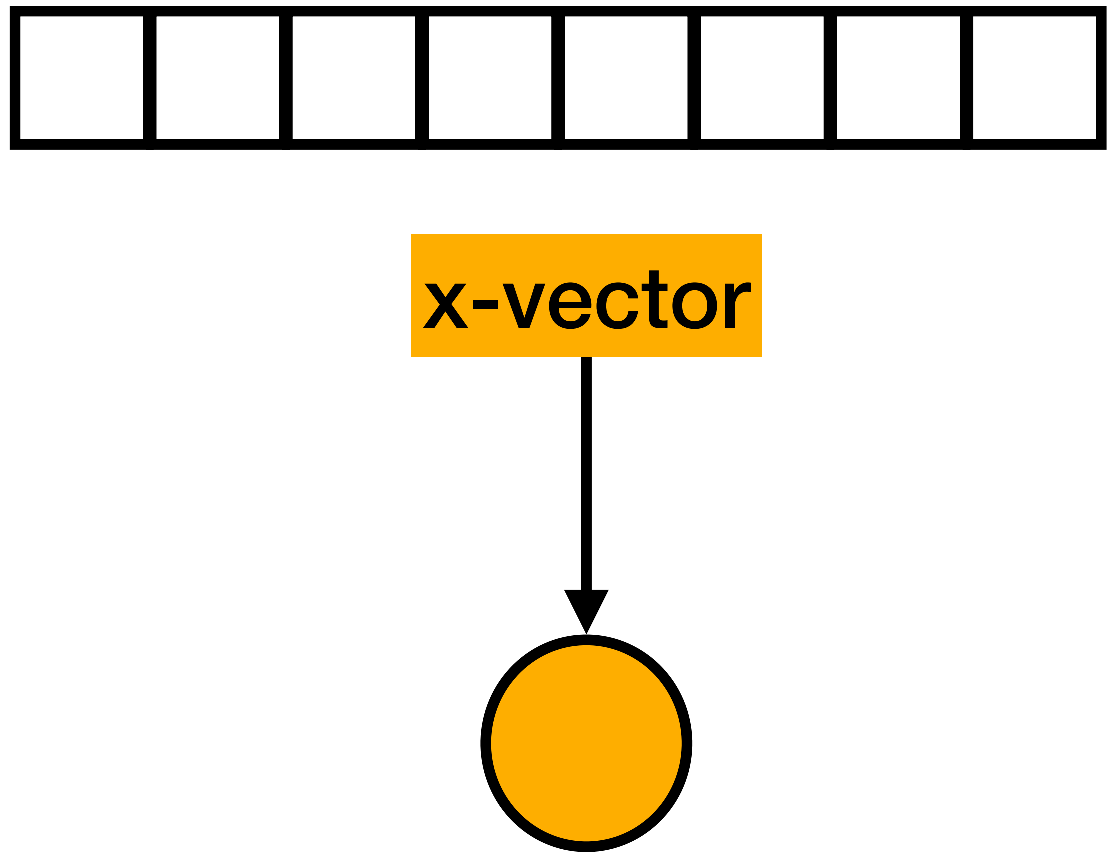
Overlap-aware spectral clustering



New formulation for spectral clustering

The basic clustering problem: a graph view

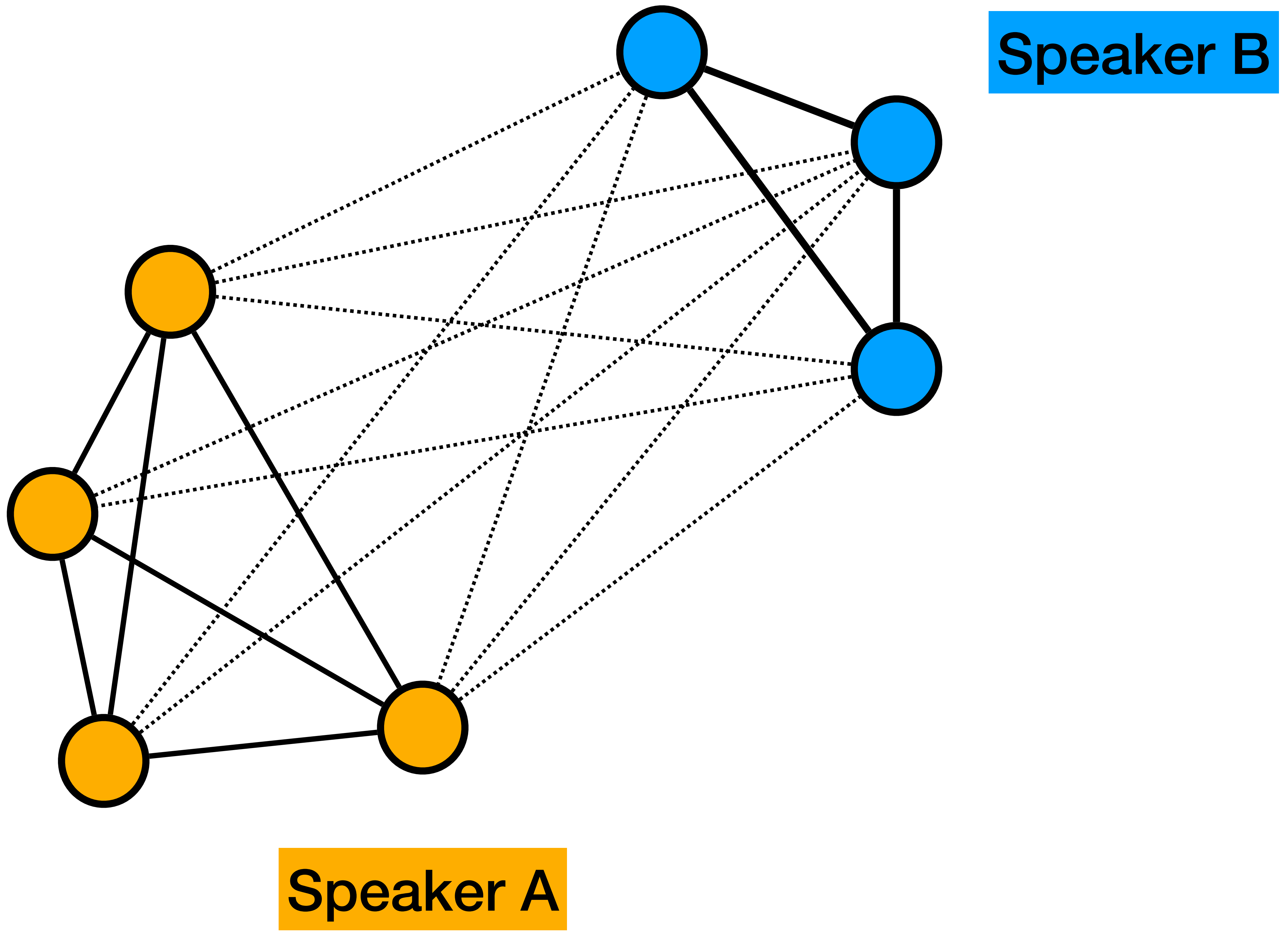




Cosine similarity

New formulation for spectral clustering

The basic clustering problem: a graph view

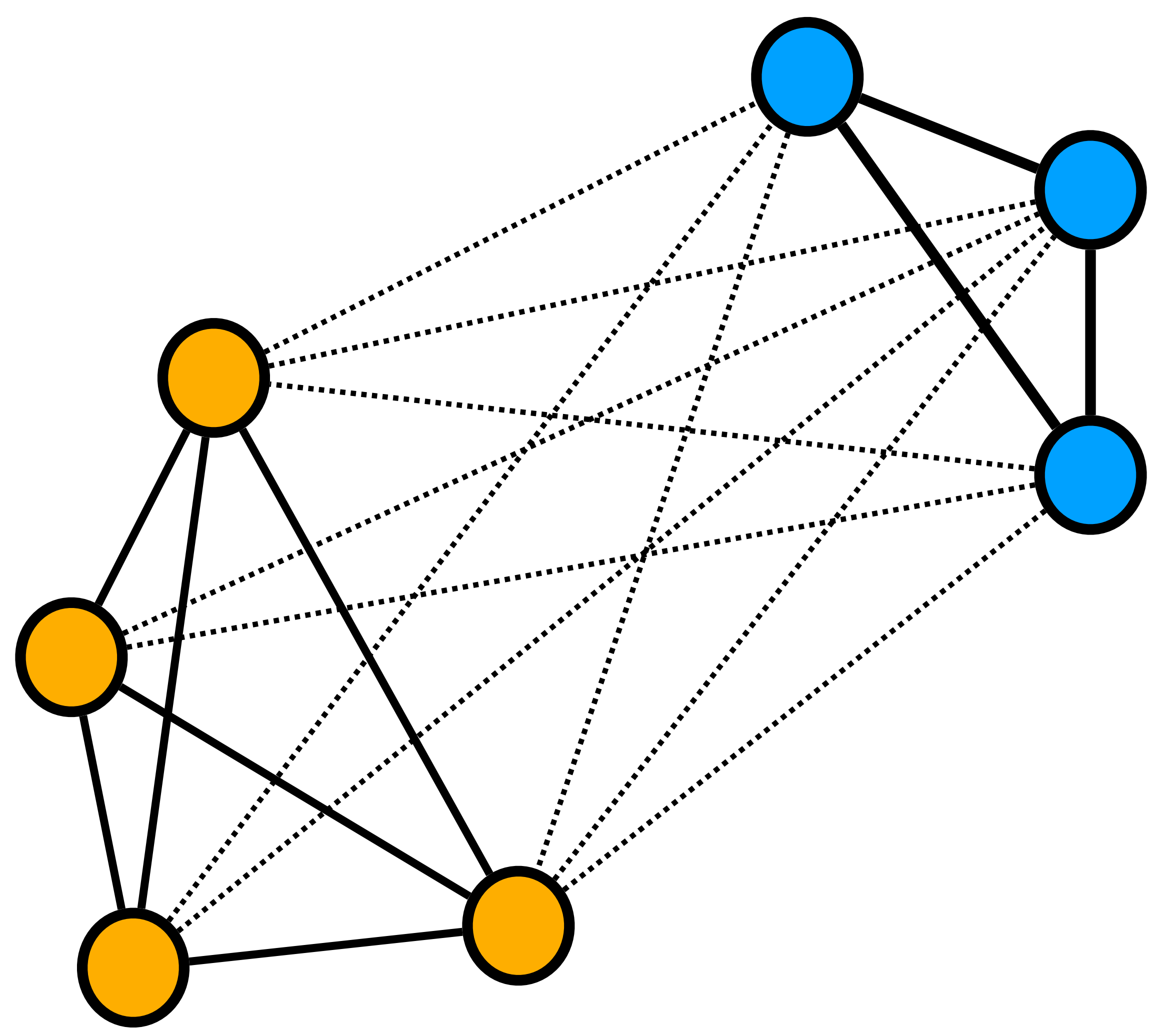


Edge weights within a group

Edge weights across groups

New formulation for spectral clustering

The basic clustering problem: a graph view



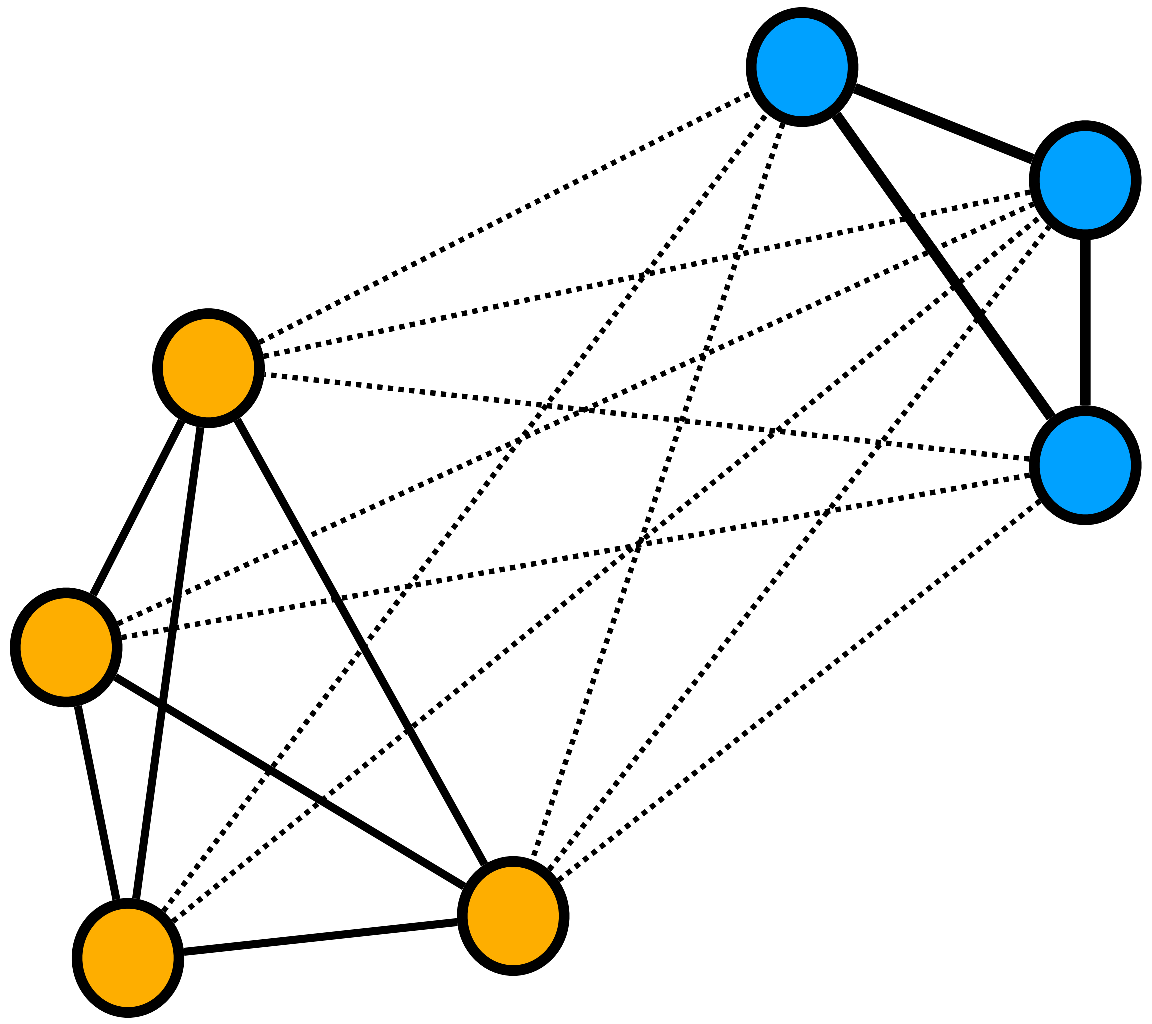
maximize $\dfrac{\text{Edge weights within a group}}{\text{Edge weights across groups}}$

$$\begin{aligned}\text{maximize} \quad & \epsilon(X) = \frac{1}{K}\sum_{k=1}^{K}\frac{X_k^T \mathbf{A} X_k}{X_k^T \mathbf{D} X_k} \\ \text{subject to} \quad & X \in \{0,1\}^{N\times K}, \\ & X\mathbf{1}_K = \mathbf{1}_N .\end{aligned}$$

K speakers, **N** segments

New formulation for spectral clustering

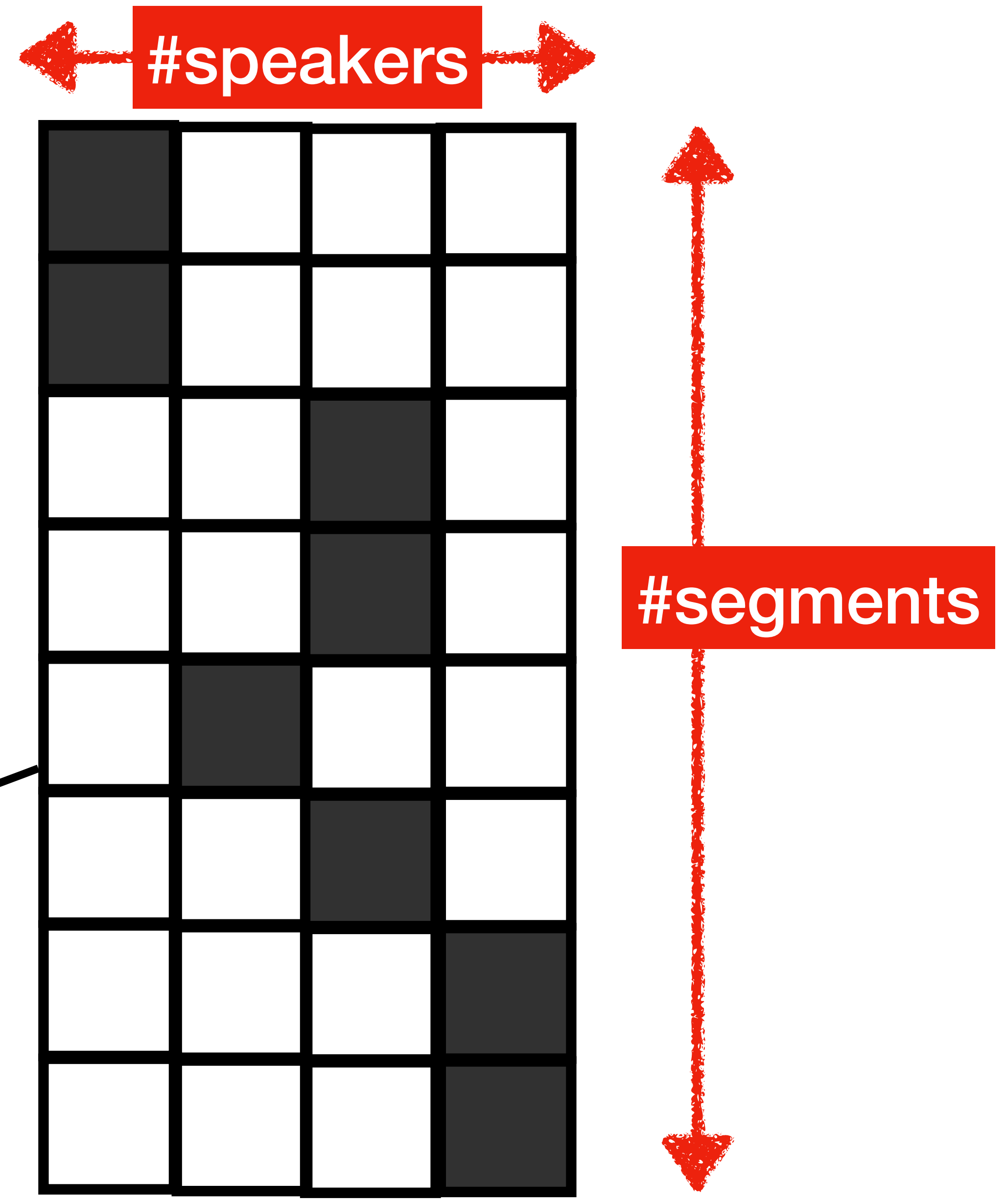
The basic clustering problem: a graph view



$$\text{maximize} \quad \epsilon(X) = \frac{1}{K}\sum_{k=1}^{K}\frac{X_k^T \mathbf{A} X_k}{X_k^T \mathbf{D} X_k}$$

$$\text{subject to} \quad X \in \{0,1\}^{N\times K}, \quad X\mathbf{1}_K = \mathbf{1}_N\,.$$

Final cluster assignment matrix



New formulation for spectral clustering

This problem is NP-hard!

$$
\begin{aligned}
\text{maximize} \quad & \epsilon(X) = \frac{1}{K} \sum_{k=1}^{K} \frac{X_k^T \mathbf{A} X_k}{X_k^T \mathbf{D} X_k} \\
\text{subject to} \quad & X \in \{0,1\}^{N \times K}, \\
& X\mathbf{1}_K = \mathbf{1}_N.
\end{aligned}
$$

Remove the discrete constraints to make the problem solvable

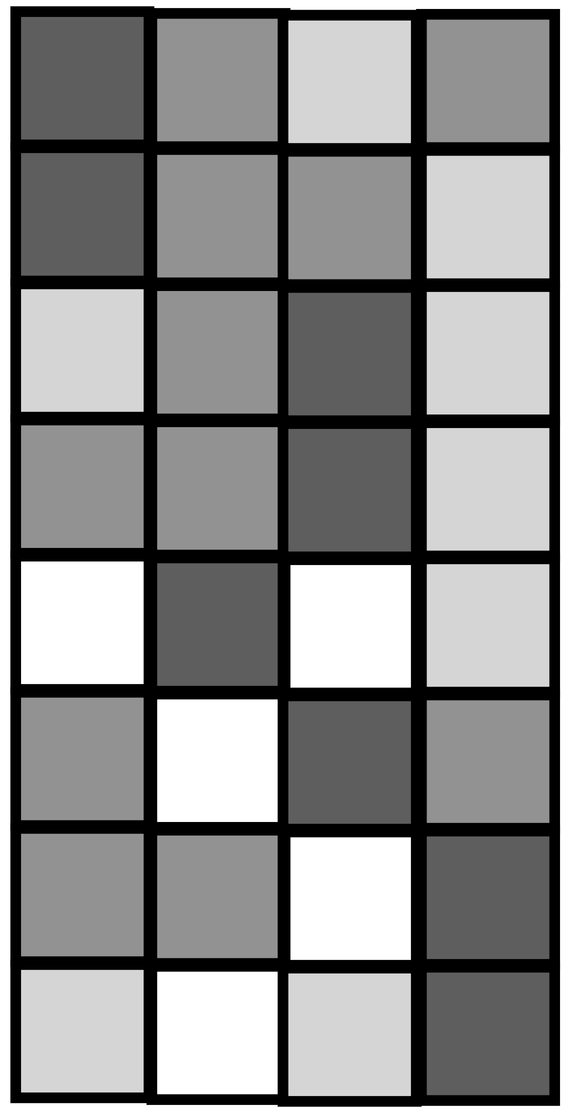
New formulation for spectral clustering

Relaxed problem has a set of solutions

$$\text{maximize} \quad \epsilon(X) = \frac{1}{K}\sum_{k=1}^{K}\frac{X_k^T\mathbf{A}X_k}{X_k^T\mathbf{D}X_k}$$

$$\text{subject to} \quad \cancel{X \in \{0,1\}^{N\times K}, \quad X\mathbf{1}_K = \mathbf{1}_N.}$$

Taking the Eigen-decomposition of $D^{-1}A$

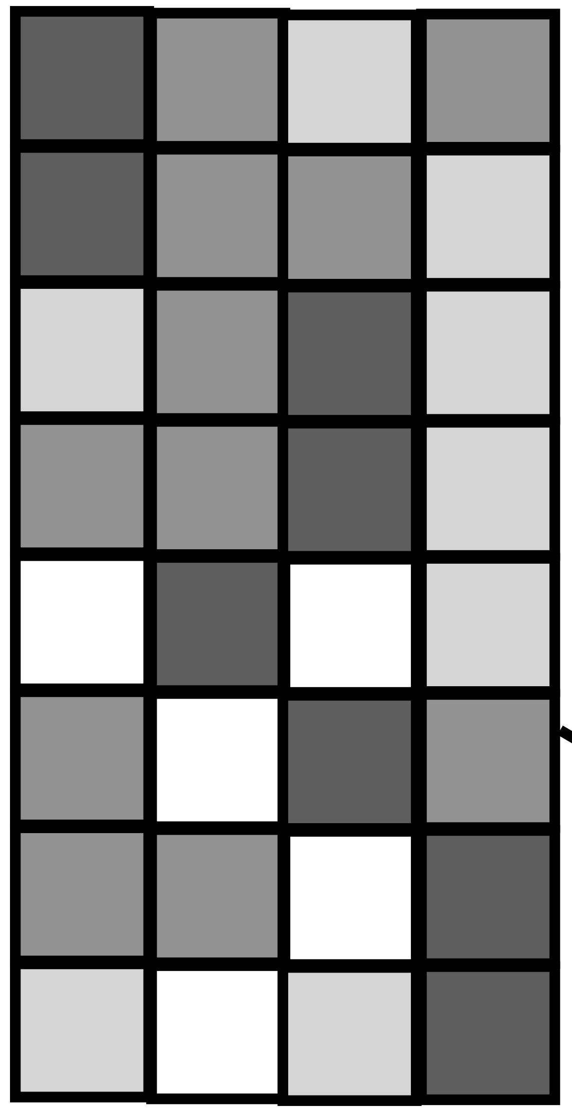


and its orthonormal transforms

Set of solutions to the **relaxed** problem

New formulation for spectral clustering

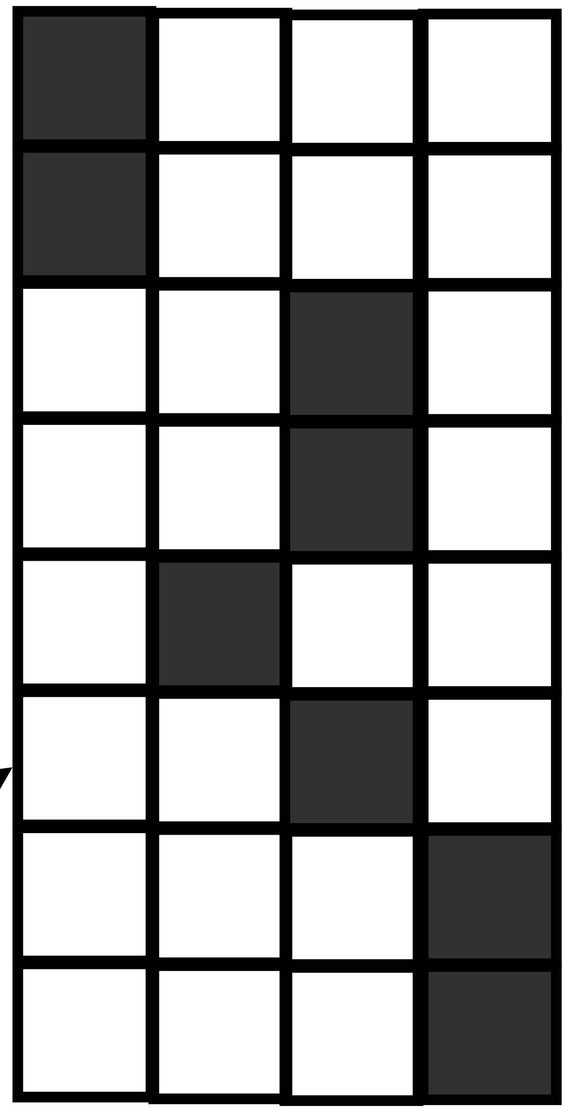
Now we need to discretize this solution!



and its orthonormal transforms

subject to $X \in \{0,1\}^{N \times K}$,

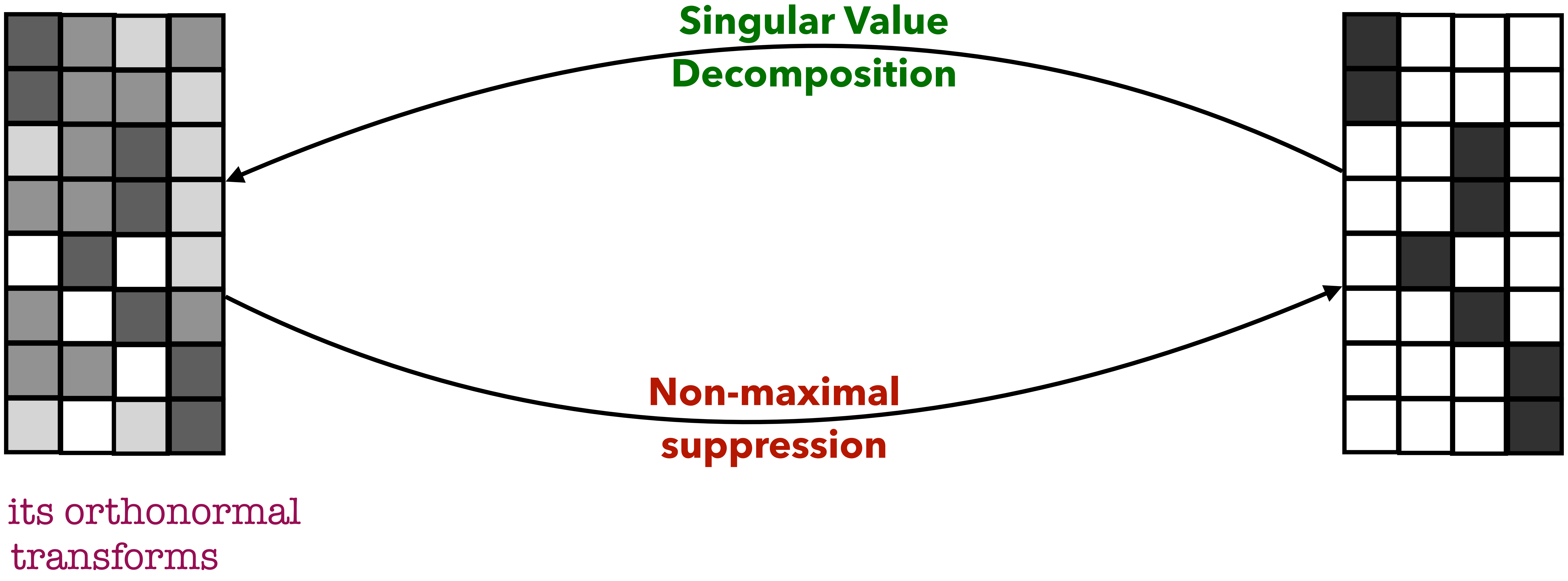
$$X\mathbf{1}_K = \mathbf{1}_N \,.$$



Find a matrix which is **discrete** and also close to any one of the **orthonormal transformations** of the relaxed solution

New formulation for spectral clustering

Now we need to discretize this solution!

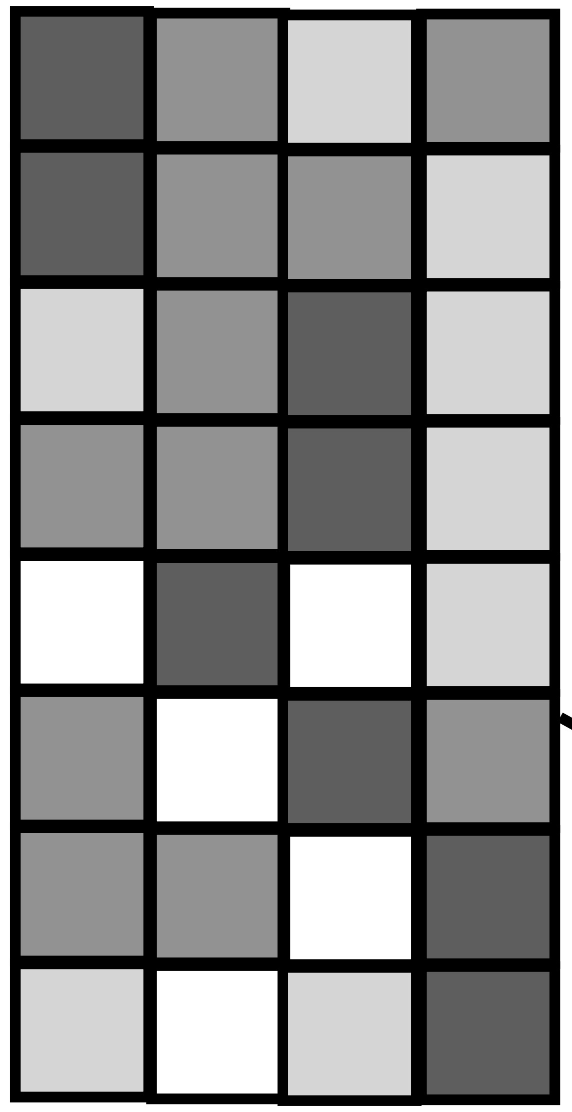


Iterate until convergence

Let us now make it overlap-aware

Suppose we have $\mathbf{v}_{OL}$

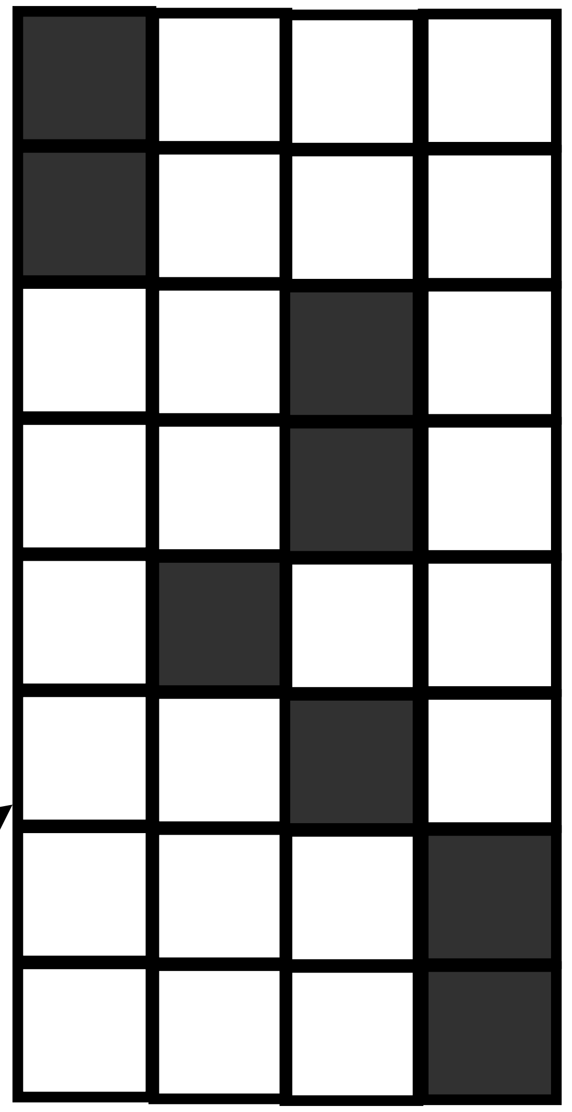
Overlap Detector



and its orthonormal transforms

subject to $X \in \{0,1\}^{N \times K}$,

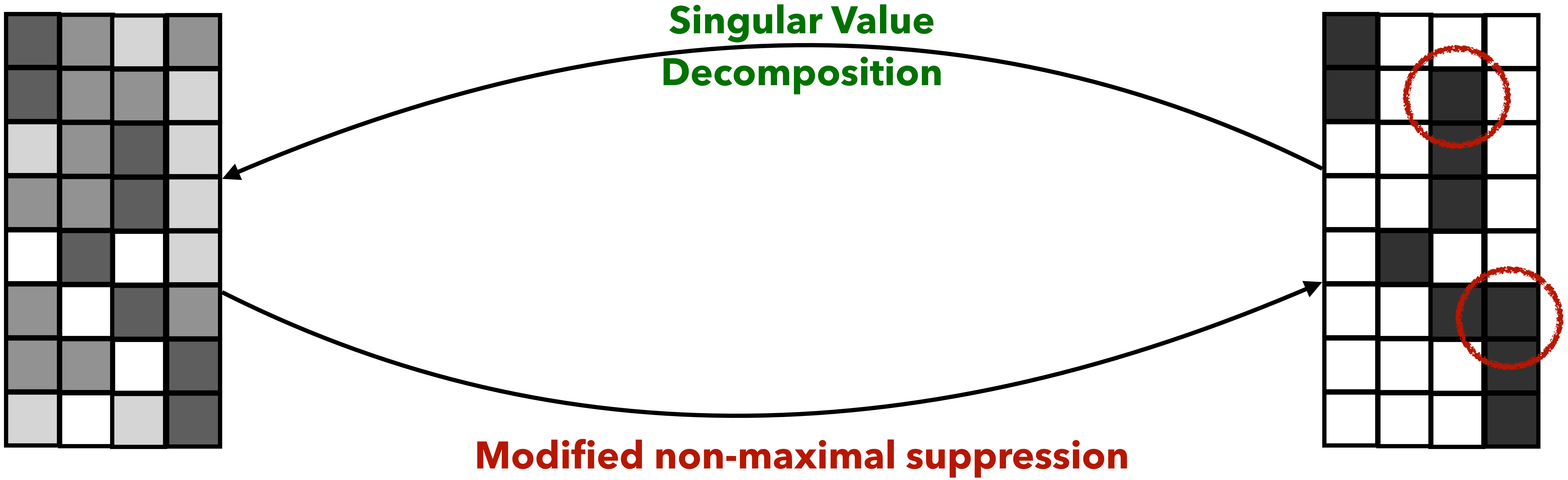
$$X\mathbf{1}_K = \mathbf{1}_N + \boldsymbol{v}_{OL}.$$



Discrete constraint is modified to include overlap detector output

Let us now make it overlap-aware

Modify non-maximal suppression to pick top 2 speakers

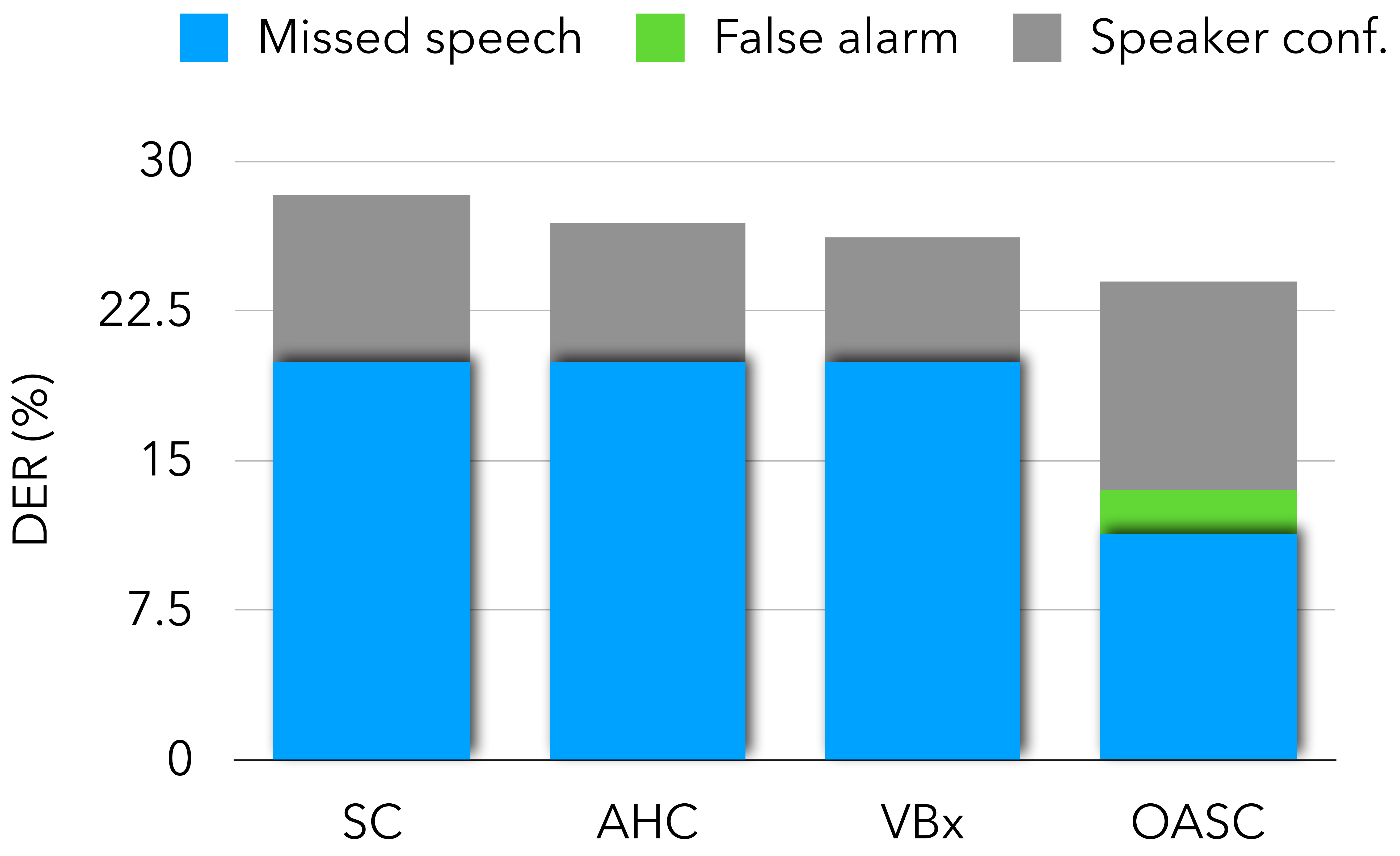


and its orthonormal transforms

Iterate until convergence

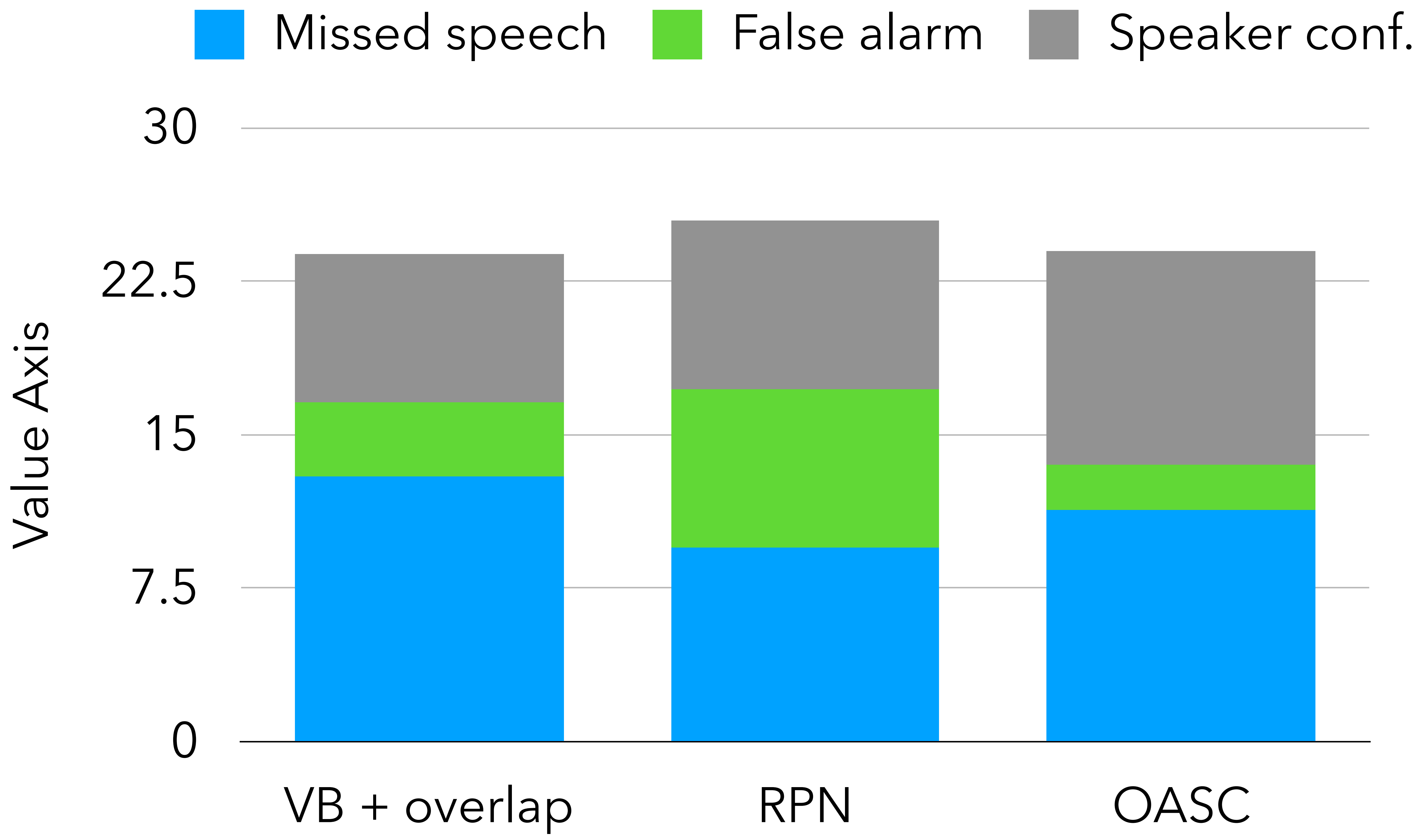
Results on AMI Mix-Headset eval

12.0% relative improvement over spectral clustering baseline



Results on AMI Mix-Headset eval

Comparable with other overlap-aware diarization methods



Does not require **matching training data** or **initialization** with other diarization systems.

Overlap-aware Diarization

Further reading

1. "Probing the infomation encoded in x-vectors." **D. Raj**, D. Snyder, D. Povey, S. Khudanpur. *IEEE ASRU 2019*.

2. "DOVER-Lap: A method for combining overlap-aware diarization outputs." **D. Raj**, P. Garcia, Z. Huang, S. Watanabe, D. Povey, A. Stolcke, S. Khudanpur. *IEEE SLT 2021*.

3. "The Hitachi-JHU DIHARD III system: Competitive end-to-end neural diarization and x-vector clustering systems combined by DOVER-Lap." S. Horiguchi, N. Yalta, P. Garcia, Y. Takashima, Y. Xue, **D. Raj**, Z. Huang, Y. Fujita, S. Watanabe, S. Khudanpur. *Third DIHARD Speech Diarization Challenge*.

4. "Target-speaker voice activity detection with improved i-vector estimation for unknown number of speaker." M. He, **D. Raj**, Z. Huang, J. Du, Z. Chen, S. Watanabe. *InterSpeech 2021*

5. "Reformulating DOVER-Lap label mapping as a graph partitioning problem." **D. Raj**, S. Khudanpur. *InterSpeech 2021*

6. "Low-Latency Speech Separation Guided Diarization for Telephone Conversations." G. Morrone, S. Cornell, **D. Raj**, Luca Serafini, Enrico Zovato, Alessio Brutti, Stefano Squartini. *IEEE SLT 2022*.

Remember DIHARD?

Top 2 teams used DOVER-Lap for system fusion in DIHARD III

All top teams at **VoxSRC-21** and **VoxSRC-22**, and **M2MeT** challenges, used DOVER-Lap!

```
$ pip install dover-lap

$ dover-lap <output-rttm> <input-rttms>
```

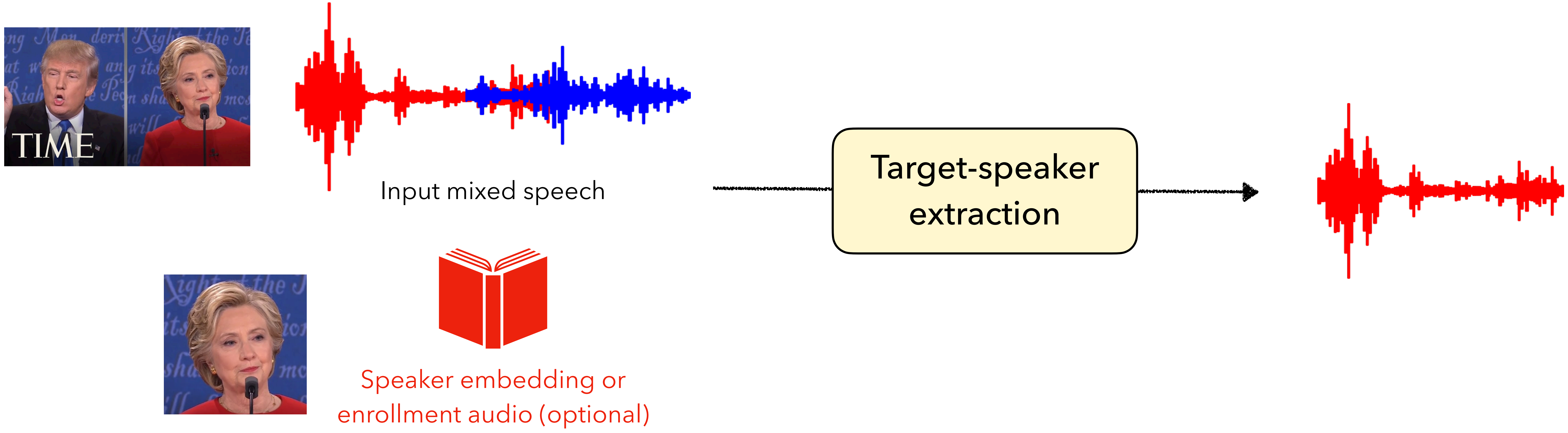
It’s easy to use!

Target-speaker Methods for ASR

What is target-speaker ASR?

Preliminary: Target speaker extraction

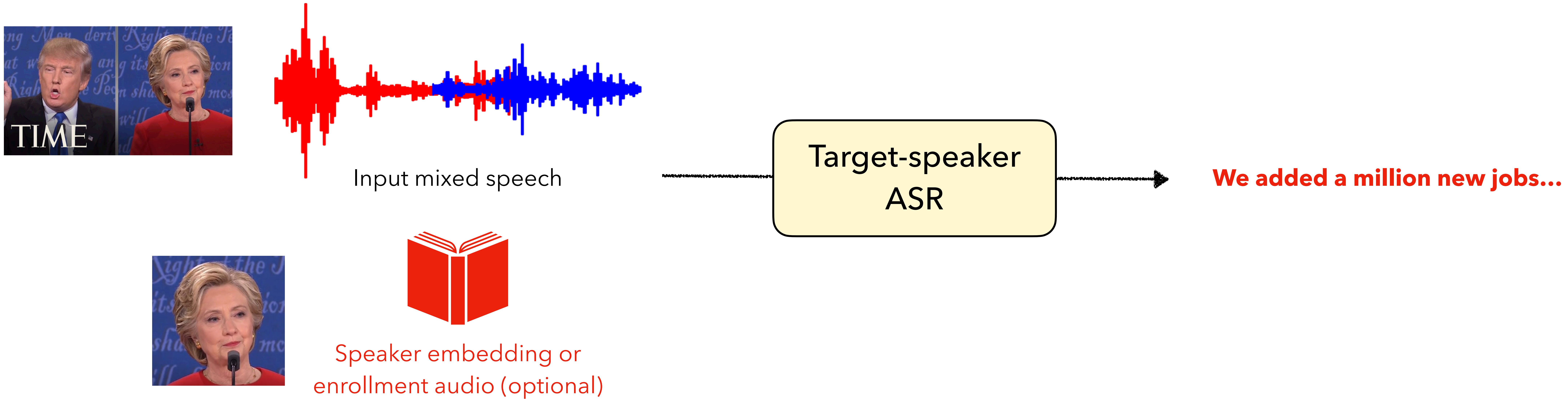
- Given an audio containing mixed speech, *extract* the speech of a **target speaker**
- Auxiliary information: enrollment audio or speaker embedding



What is target-speaker ASR?

Target speaker extraction + ASR

- Given an audio containing mixed speech, *transcribe* the speech of a **target speaker**
- Auxiliary information: enrollment audio or speaker embedding



What is target-speaker ASR?

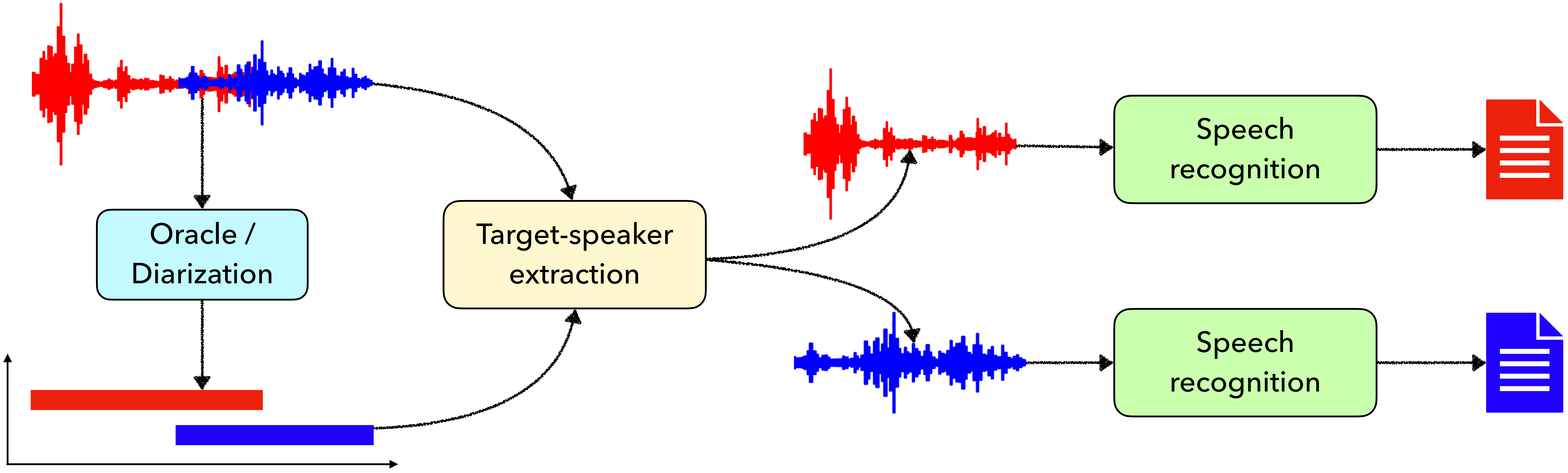
Methods

- Methods used for Target-speaker ASR depend on the application scenario.

Scenario	Meeting Transcription	Voice-based Assistant
Recording device	Multi-channel microphone array	Single microphone
Speakers	Multiple primary	1 primary + background
Wake-word	None	“Hey Siri”, “Alexa”, etc.
Real-time?	Optional	Required

Meeting Transcription

Approach using target speaker methods



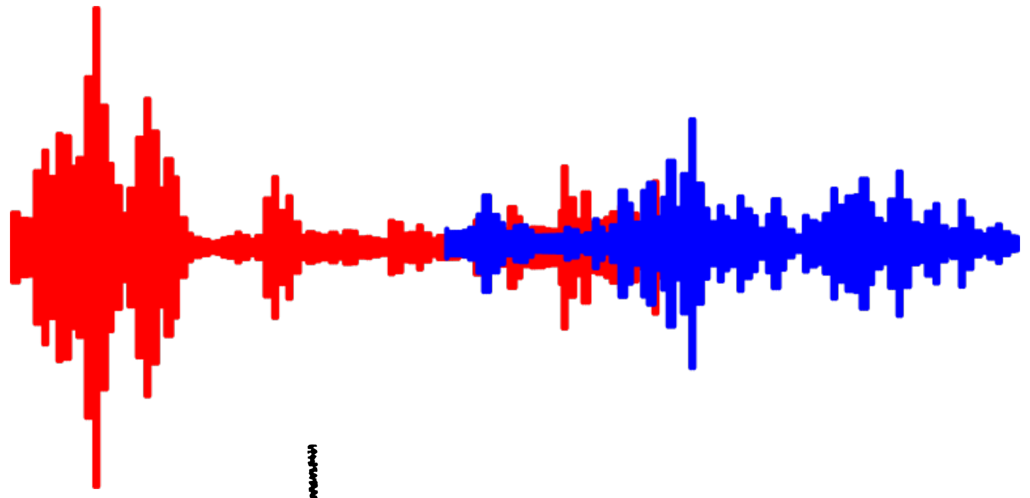
Guided source separation

Consists of 3 main steps

https://github.com/fgnt/pb_chime5

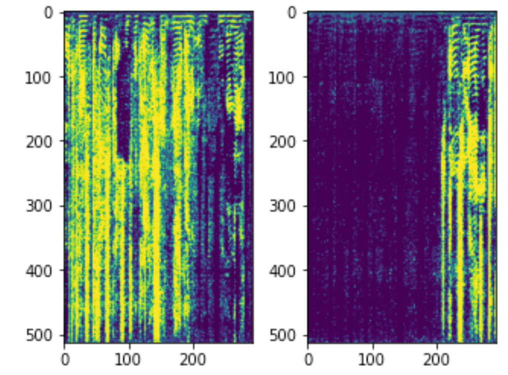
$$\mathbf{Y}_{t,f} = \underbrace{\sum_k \mathbf{X}^{\text{early}}_{t,f,k}}_{\text{Sum of speaker signals}} + \overbrace{\sum_k \mathbf{X}^{\text{tail}}_{t,f,k}}^{\text{Sum of reverb tails}} + \underbrace{\mathbf{N}_{t,f}}_{\text{Noise}}$$

$\mathbf{Y}_{t,f}$

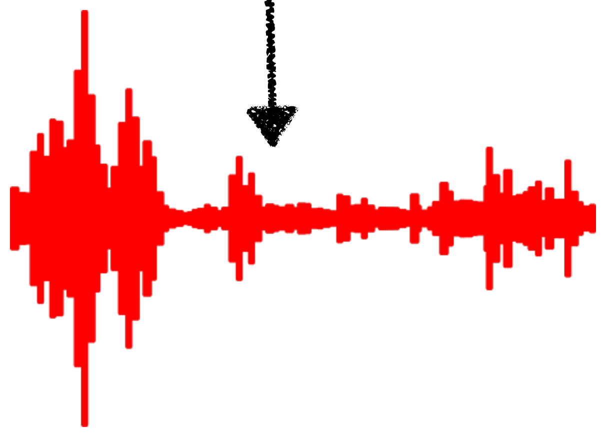


De-reverberation using Weighted Prediction Error (WPE) — Remove the late reverb

Mask estimation using mixture models — Estimate T-F masks for all speakers and noise



Mask-based MVDR beamforming — Use T-F masks to extract desired signal from input



Boeddeker, Christoph et al. "Front-end processing for the CHiME-5 dinner party scenario." *CHiME Workshop, 2018* .

Guided source separation

Limitations with original implementation

- Several iterative parts, e.g., mask estimation using complex angular GMMs.
- All implementation on CPU (with NumPy).
- Example: Applying GSS on CHiME-6 *dev* set takes ~20h with 80 jobs!

Guided source separation

GPU-acceleration + engineering tricks

GPU-accelerated Guided Source Separation for Meeting Transcription

Desh Raj[1], *Daniel Povey*[2], *Sanjeev Khudanpur*[1,3]

[1]CLSP & [3]HLTCOE, Johns Hopkins University, Baltimore, USA; [2]Xiaomi Corp., Beijing, China
draj@cs.jhu.edu, dpovey@gmail.com, khudanpur@jhu.edu

To appear at **InterSpeech 2023**

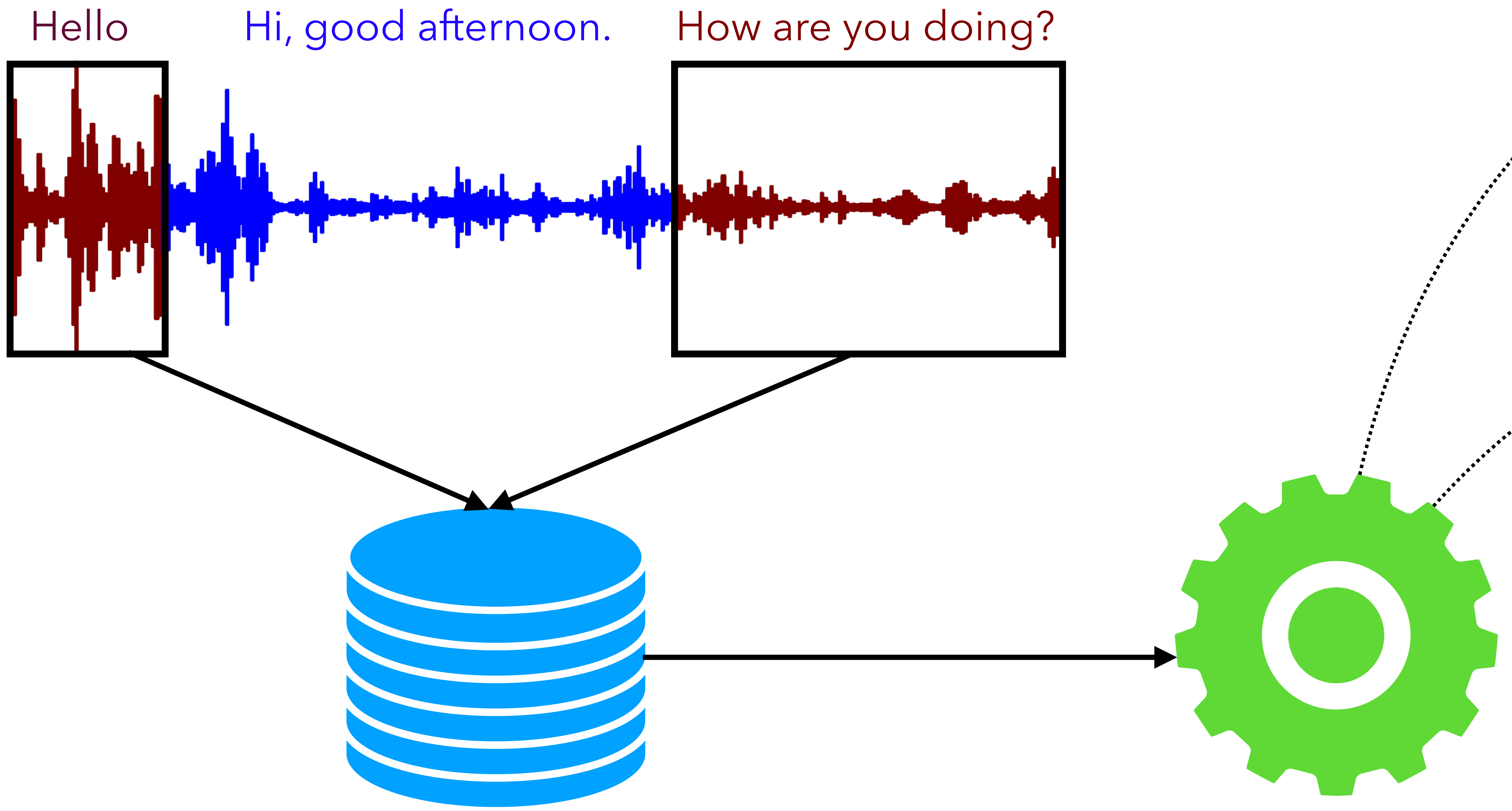
CENTER FOR LANGUAGE AND SPEECH PROCESSING

xiaomi

Guided source separation

GPU-acceleration + engineering tricks

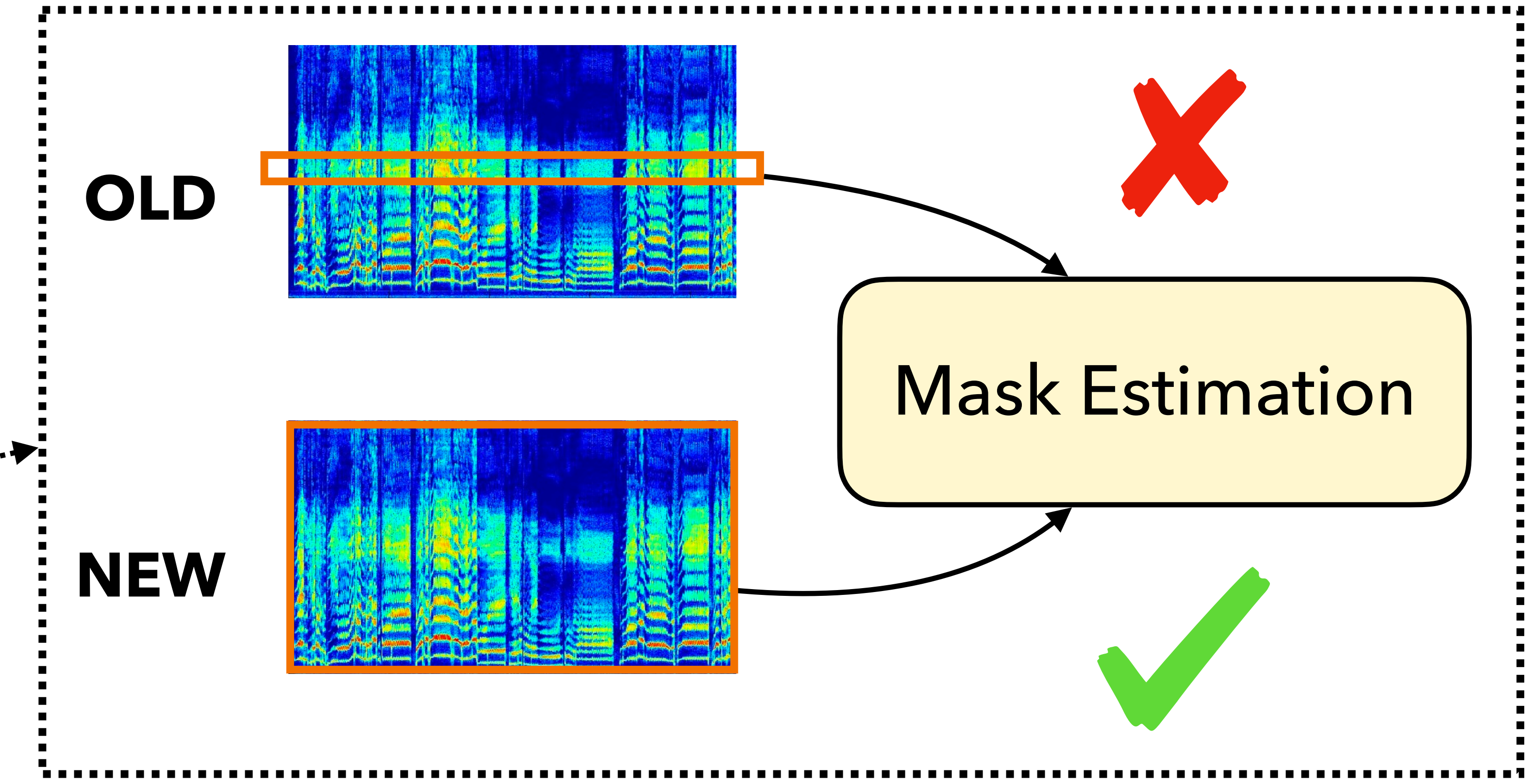
https://github.com/desh2608/gss



1. CPU-based data-loader performs smart batching of segments

2. STFT computation, WPE, mask estimation on GPU using CuPy

CuPy



3. Batched processing of STFT frequency bins

```
covariance = D * cp.einsum(
    "...dn,...Dn,...n->...dD",
    y,
    y.conj(),
    (saliency / quadratic_form),
    optimize=einsum_path,
)
```

Cache optimized path on first iteration.

Use same path on subsequent iterations.

4. einsum path caching

Guided source separation

Speed-up

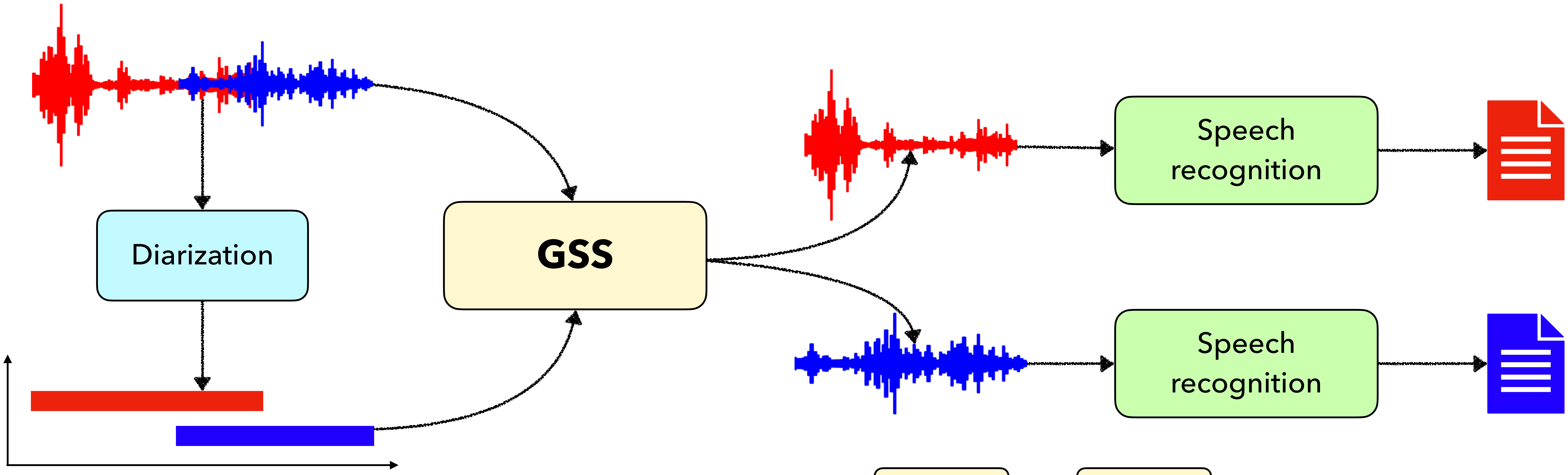
- Comparison on CHiME-6 dev set
- Old GSS: Takes **19.3** hours using 80 jobs
- New GSS: Takes **1.3** hours using 4 GPUs

CHiME-7 DASR Baseline

- Part of the official baseline in CHiME-7 DASR challenge: https://www.chimechallenge.org/current/task1/index

Meeting Transcription

Results on AMI with GSS

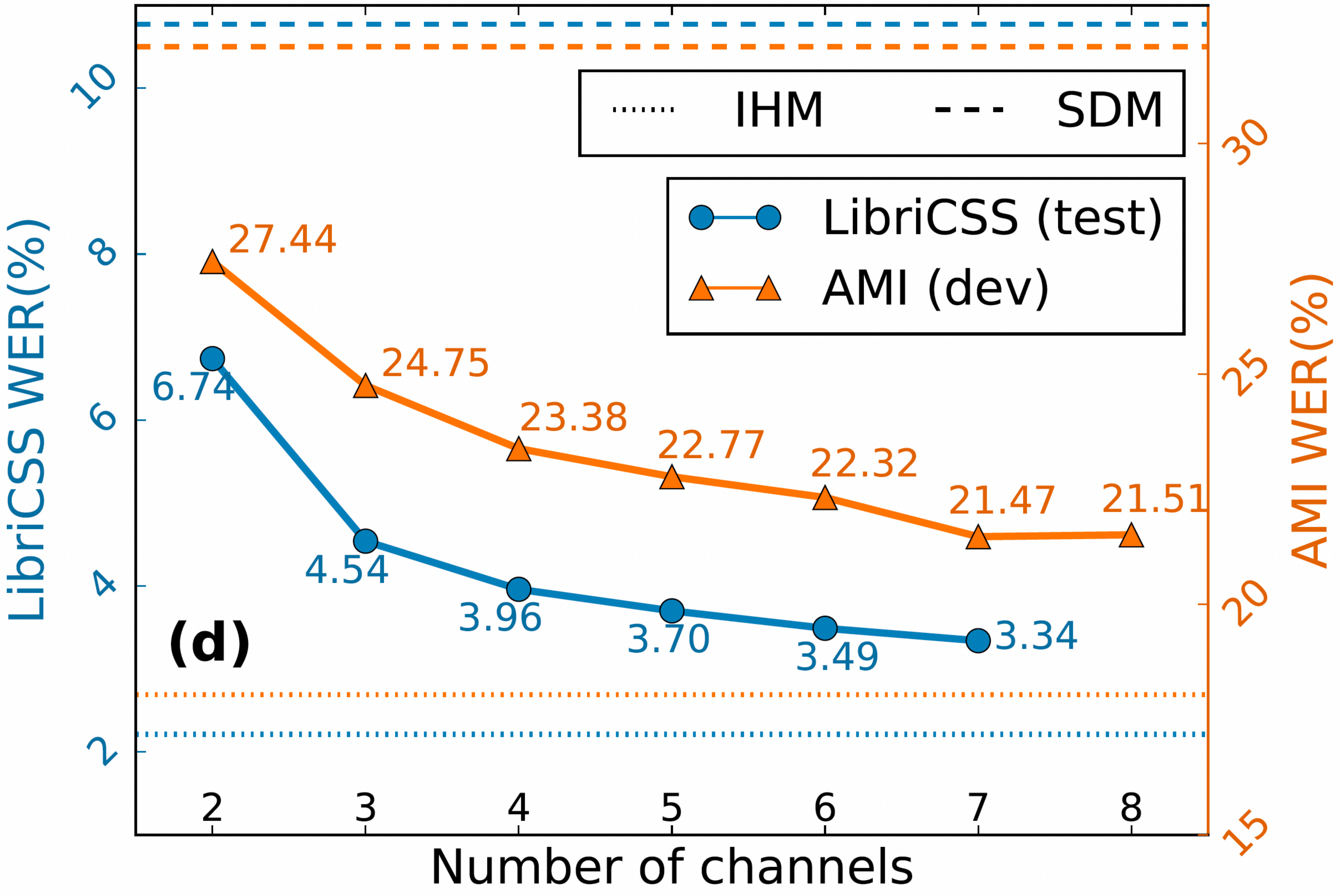


		No GSS	GSS
Diarizer	**DER (%)**	**WER (%)**	**WER (%)**
Oracle	0.0	32.1	22.8
Clustering	23.7	38.5	31.0

29.0% ↓
19.5% ↓

Guided source separation

Effect of number of channels



LibriCSS example

	REFERENCE:
No GSS	Paul declares that the false apostles were called or sent neither by men nor by man
2 channels	All declares of the false apostles [were] recalled or sent neither by men [nor by man]
7 channels	All declares that the false apostles were called or sent neither by men nor by man

Recall from earlier...

Voice assistant is very different from meeting transcription

- Methods used for Target-speaker ASR depend on the application scenario.

Scenario	Meeting Transcription	Voice-based Assistant
Recording device	Multi-channel microphone array	Single microphone
Speakers	Multiple primary	1 primary + background
Wake-word	None	"Hey Siri", "Alexa", etc.
Real-time?	Optional	Required

Voice-based Assistant

Approach using target speaker methods

ANCHORED SPEECH RECOGNITION WITH NEURAL TRANSDUCERS

Desh Raj[*1], *Junteng Jia*[2], *Jay Mahadeokar*[2], *Chunyang Wu*[2], *Niko Moritz*[2], *Xiaohui Zhang*[2], *Ozlem Kalinli*[2]

[1]Center for Language and Speech Processing, Johns Hopkins University, USA, [2]Meta AI, USA

Published at IEEE ICASSP 2023

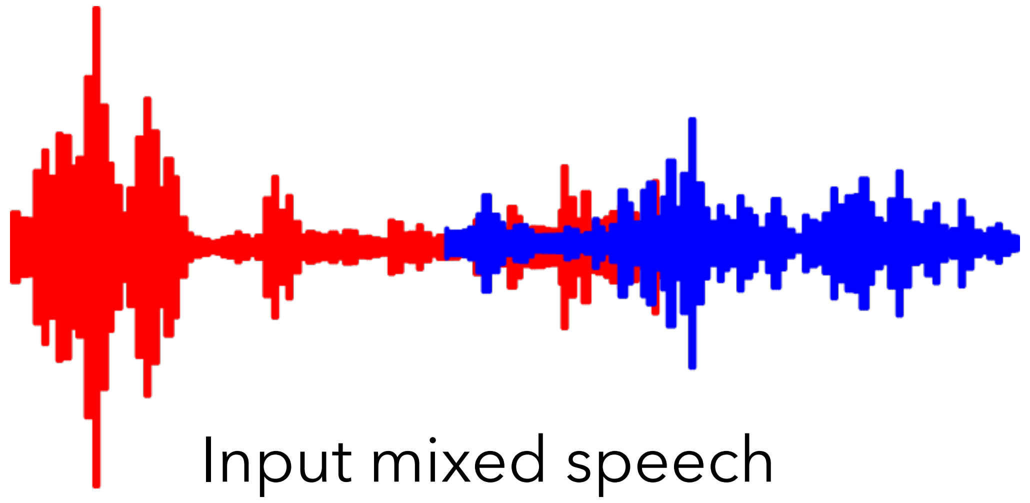
CENTER FOR LANGUAGE AND SPEECH PROCESSING

Meta

From GSS to anchored speech recognition

"Anchor" = wake-word

- **"Alexa,** play my favorite song."
- Auxiliary information: "Alexa"



Input mixed speech

"Alexa"

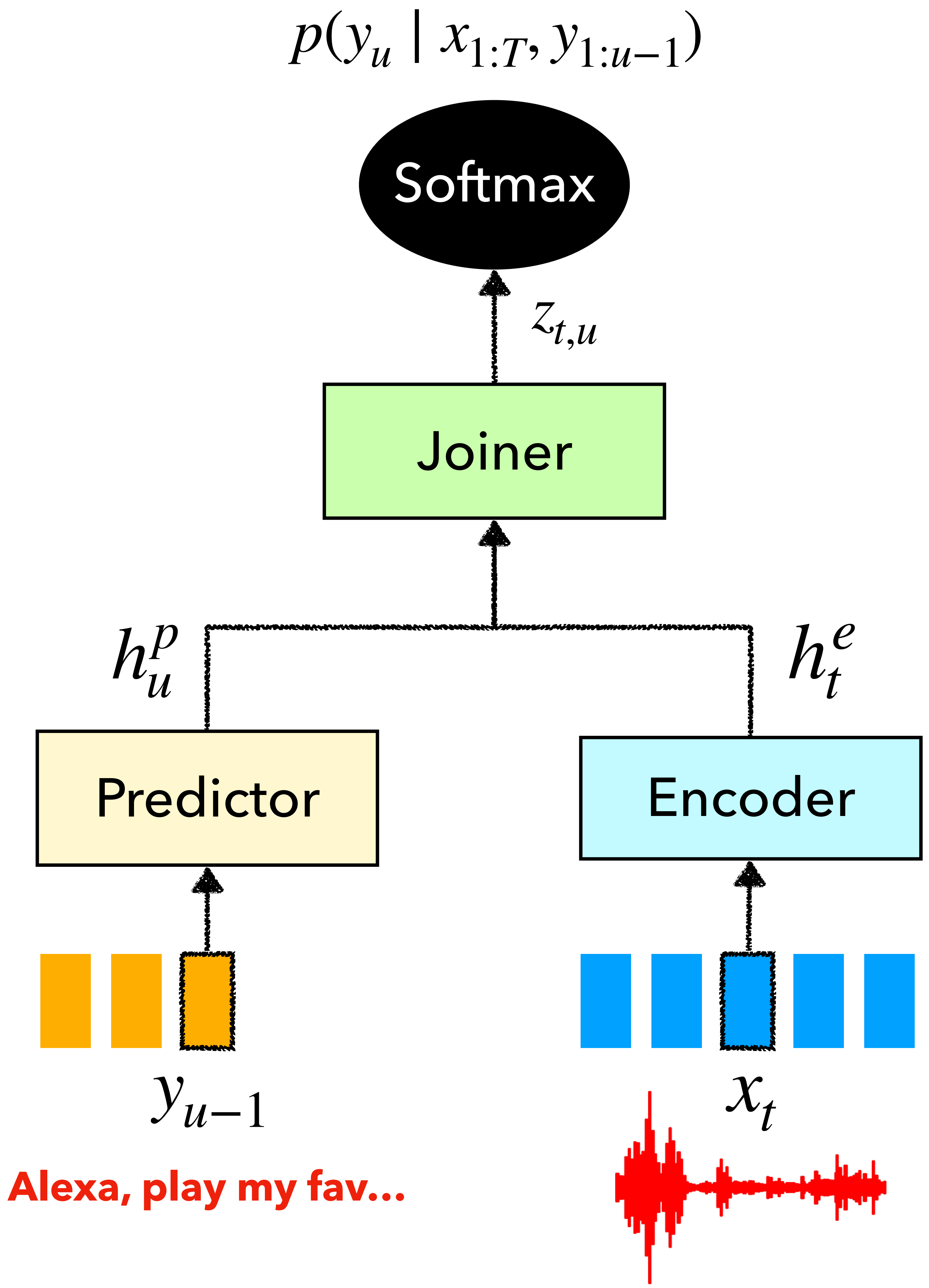
Anchored Speech Recognition

Alexa, play my favorite song...

Wang, Yiming et al. "End-to-end Anchored Speech Recognition." IEEE ICASSP, 2019.

Voice-based Assistant

The basic ASR system: Neural transducer

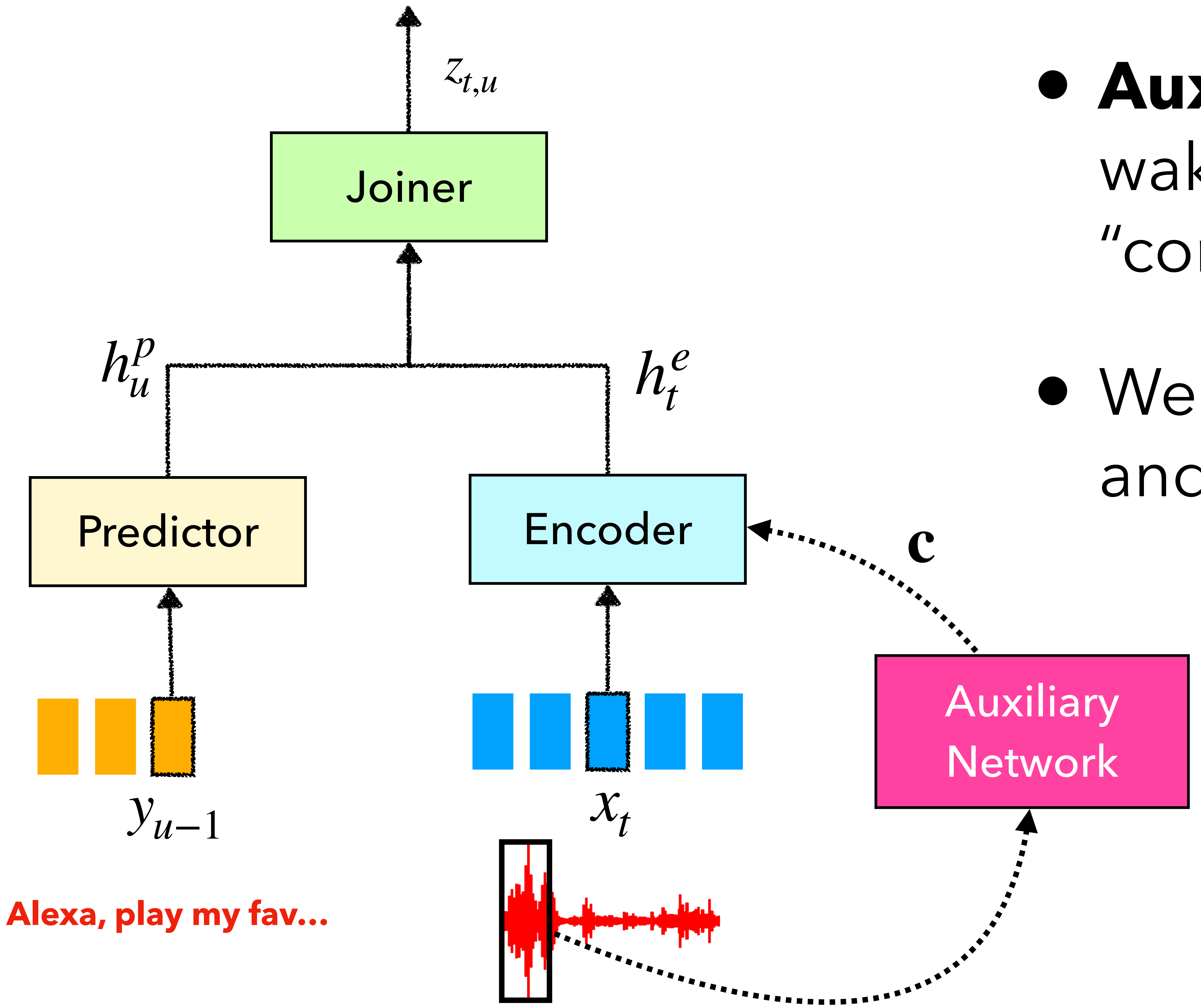


- **Encoder** converts input *audio* to high-dimensional representation
- **Predictor** is an autoregressive model that encodes input *text*
- **Joiner** combines audio and text representations to predict next token

Voice-based Assistant

1. Biasing the encoder with context

⚡ **Encoder can use context embedding to suppress background speech.**



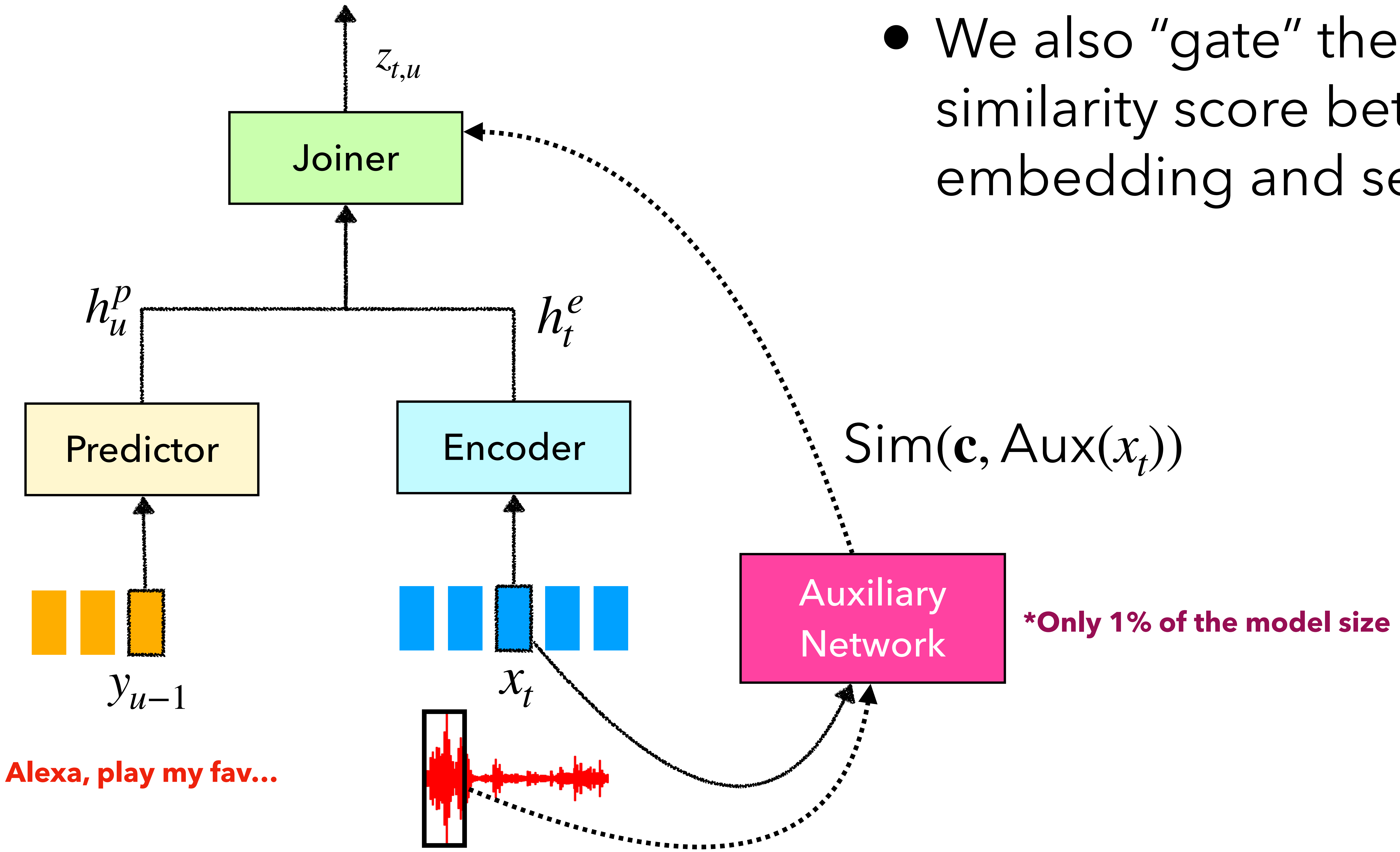
*Only 1% of the model size

- **Auxiliary network** encodes the wake-word segment into a "context" embedding
- We concat this to input features and project to original dimension

Voice-based Assistant

2. Gating the joiner

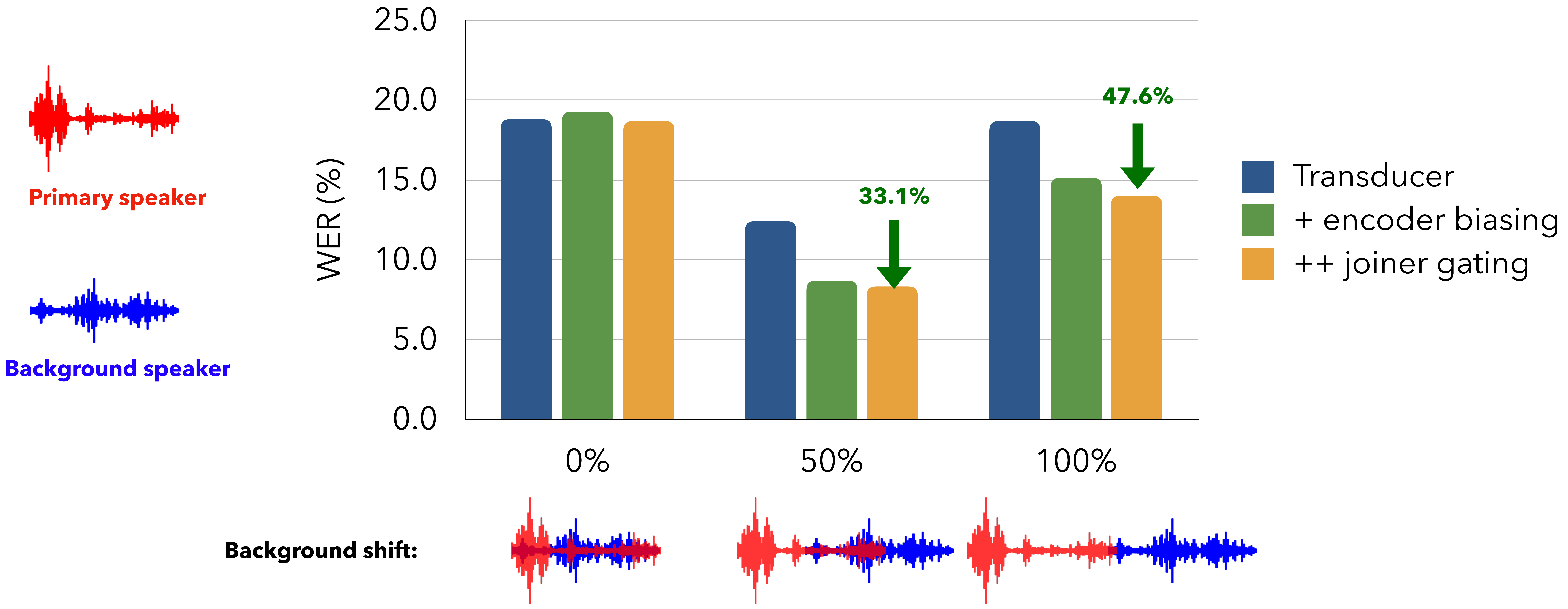
Boost the logits for blank tokens when speaker is different from wake-word segment.



- We also "gate" the logits with the similarity score between context embedding and segments.

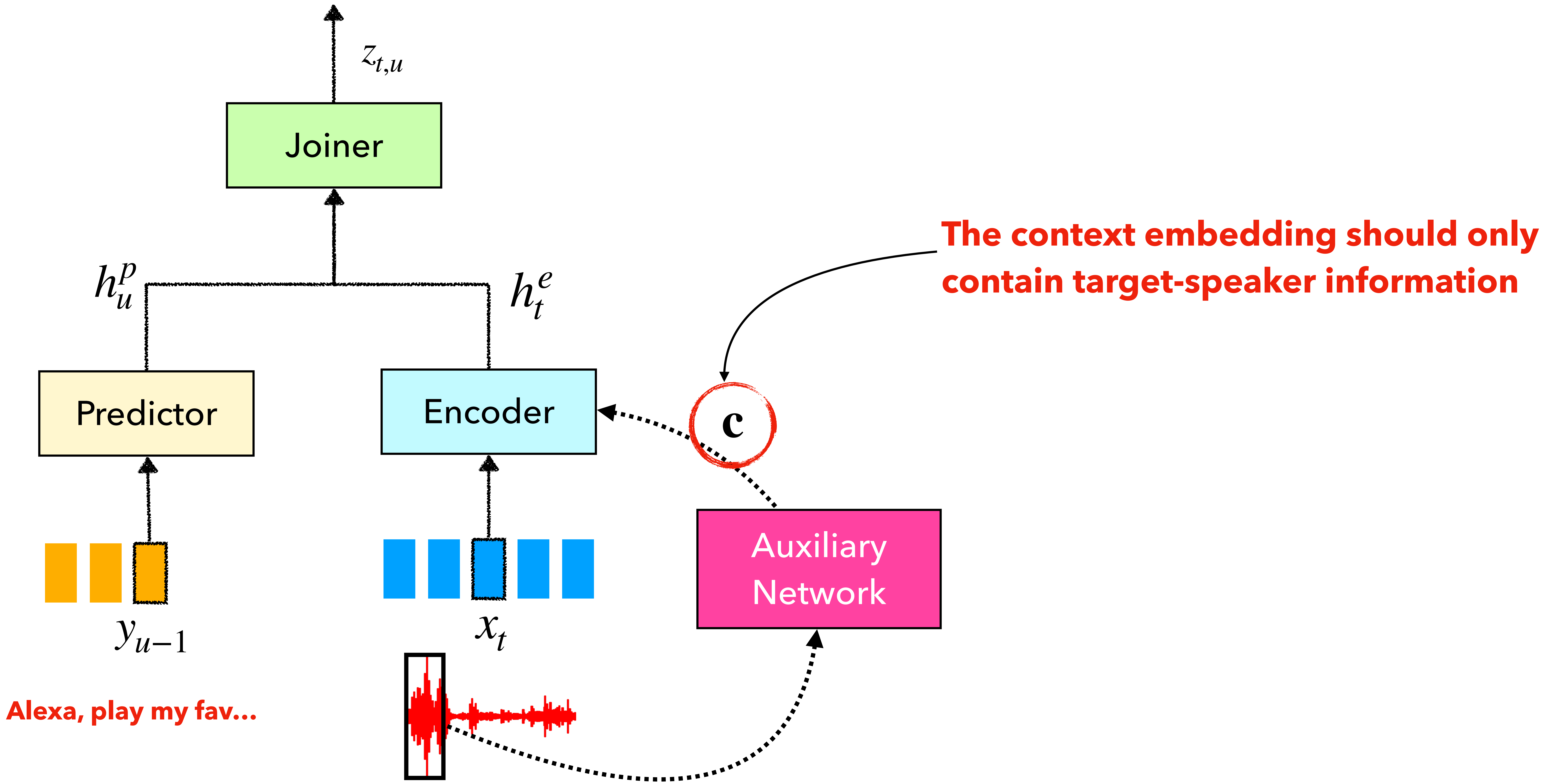
Effect of TS-ASR

WER on LibriSpeech mixtures (average over SNRs 1~20 dB)



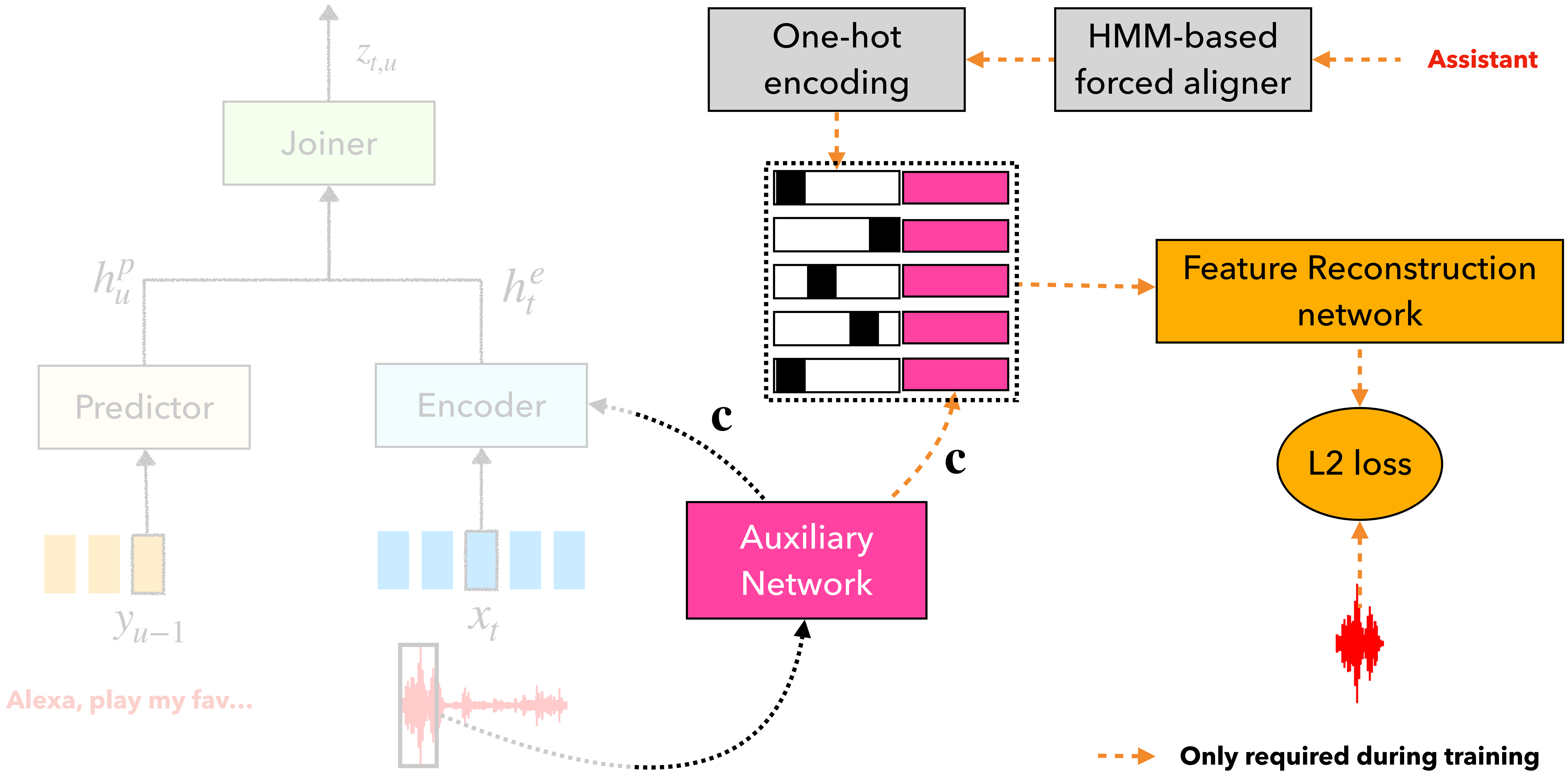
Let’s think about the context embedding

We want to disentangle “style” from “content”



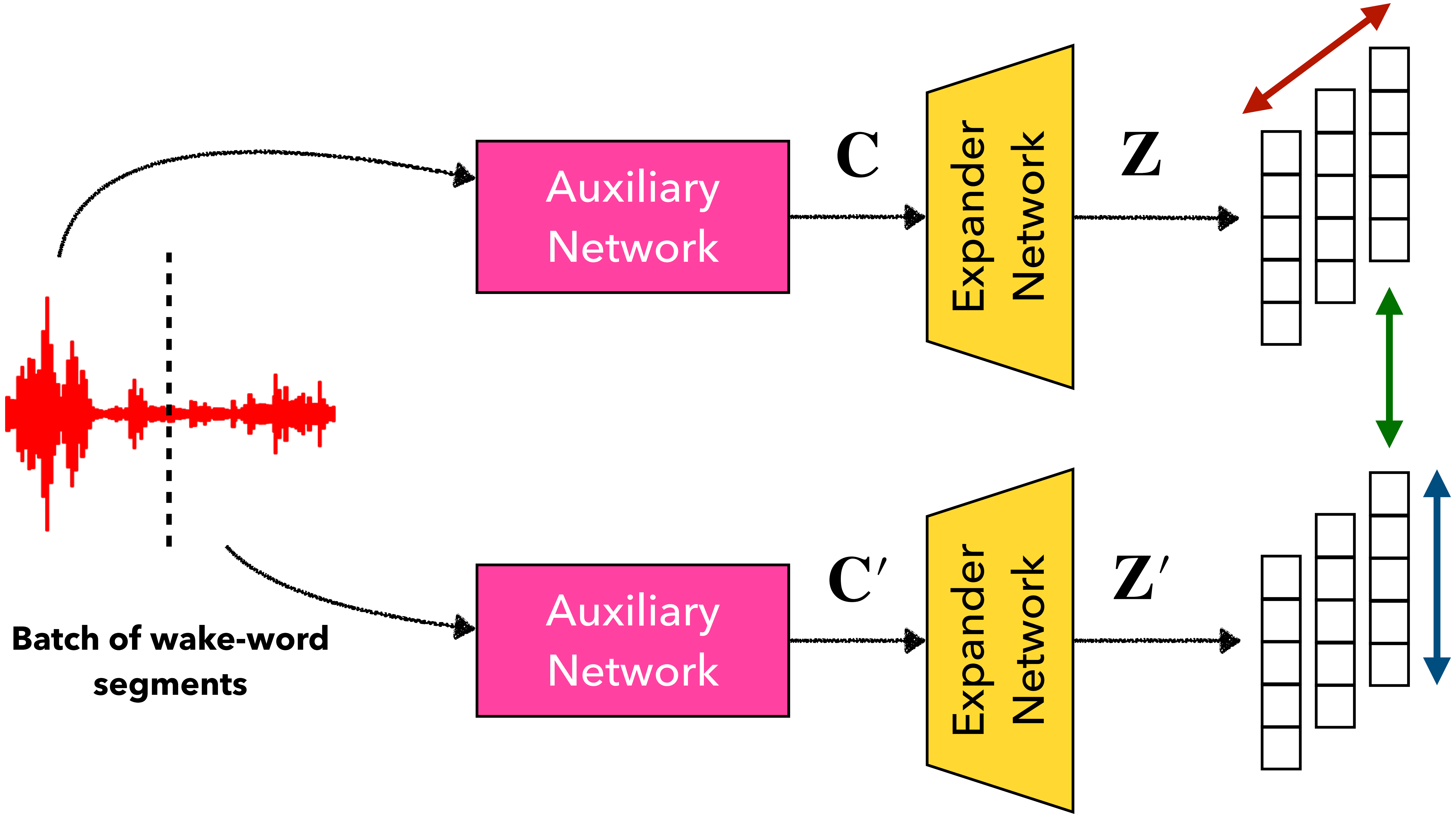
Disentangling "style" from "content"

Method 1: Feature Reconstruction



Disentangling "style" from "content"

Method 2: VIC Regularization



Variance prevents representations from collapsing to a constant

$$v(\mathbf{Z}) = \frac{1}{\mathfrak{D}} \sum_{j=1}^{\mathfrak{D}} \max\left(0, 1 - \sqrt{\mathrm{Var}(\mathbf{z}^j)}\right)$$

Invariance ensures that representations are similar for the two halves

$$s(\mathbf{Z}, \mathbf{Z}') = \mathrm{MSE}(\mathbf{Z}, \mathbf{Z}')$$

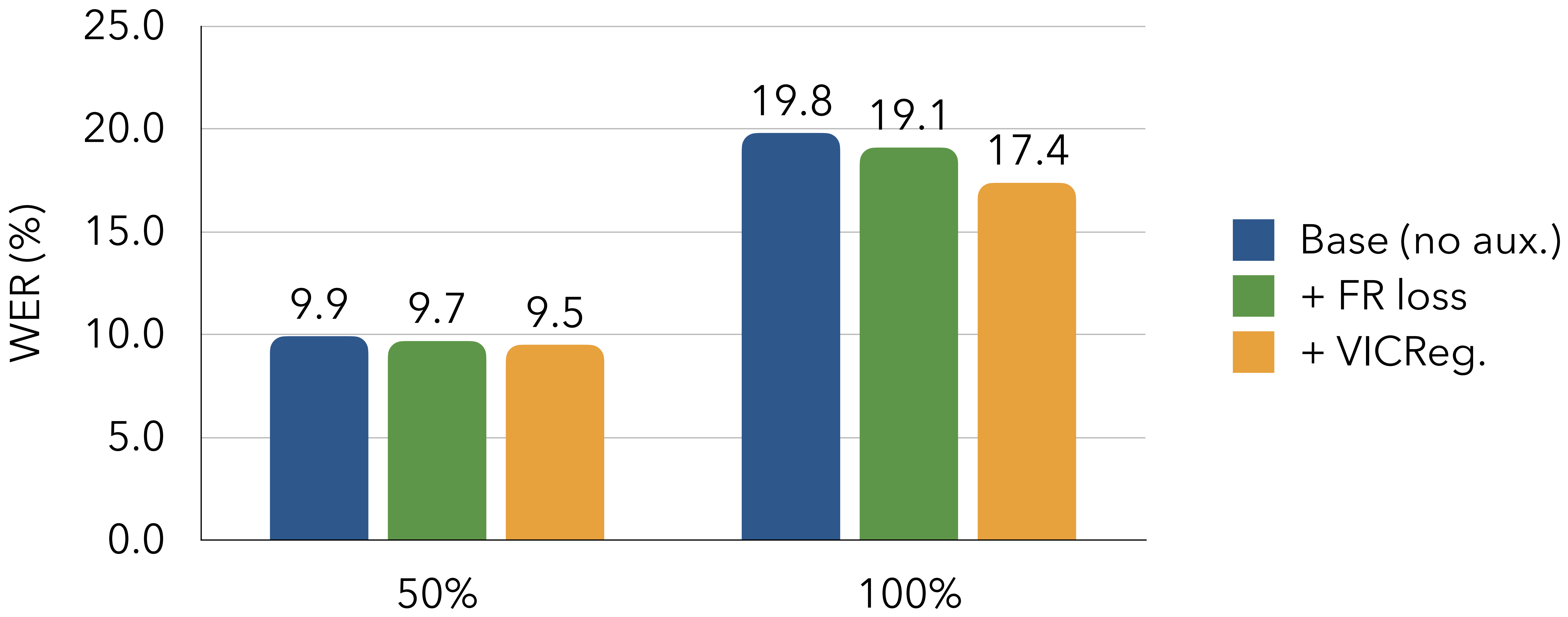
Covariance reduces correlation among dimensions to increase information content

$$c(\mathbf{Z}) = \frac{1}{\mathfrak{D}} \sum_{i \neq j} [\mathrm{cov}(\mathbf{Z})]_{i,j}^2$$

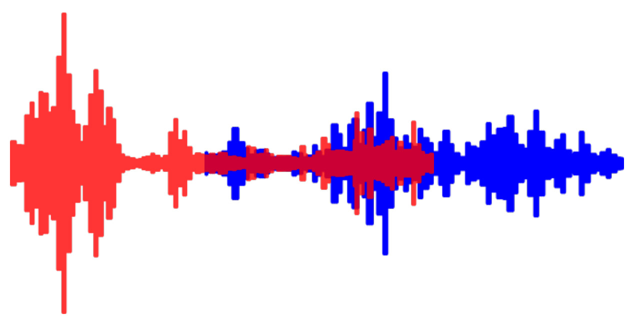
Bardes, Adrien et al. "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning." *ICLR, 2022.*

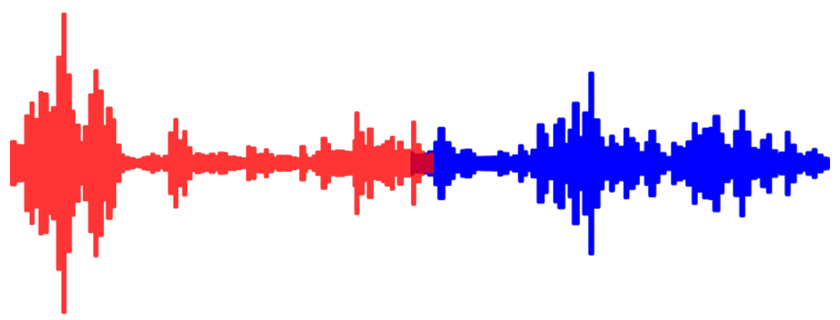
Effect of auxiliary objectives

WER on LibriSpeech mixtures (average over SNRs 1~10 dB)



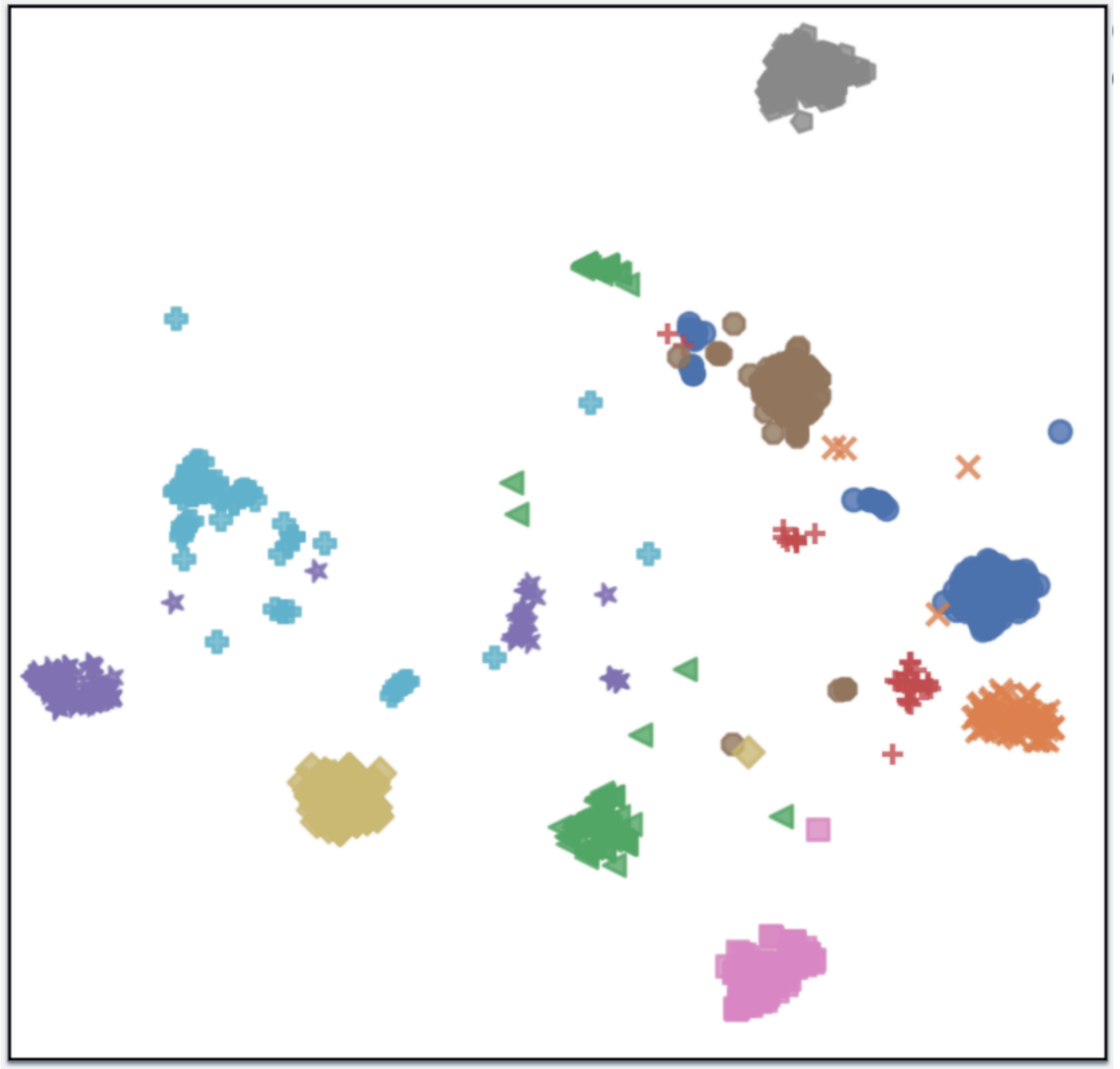
Background shift:





Context embeddings capture speaker characteristics

T-SNE clustering of context embeddings



Target-speaker methods
Further reading

1. "Auxiliary loss function for target speech extraction and recognition with weak supervision based on speaker characteristics." K. Zmolikova, M. Delcroix, **D. Raj**, S. Watanabe, J. Černocký. *InterSpeech 2021*.

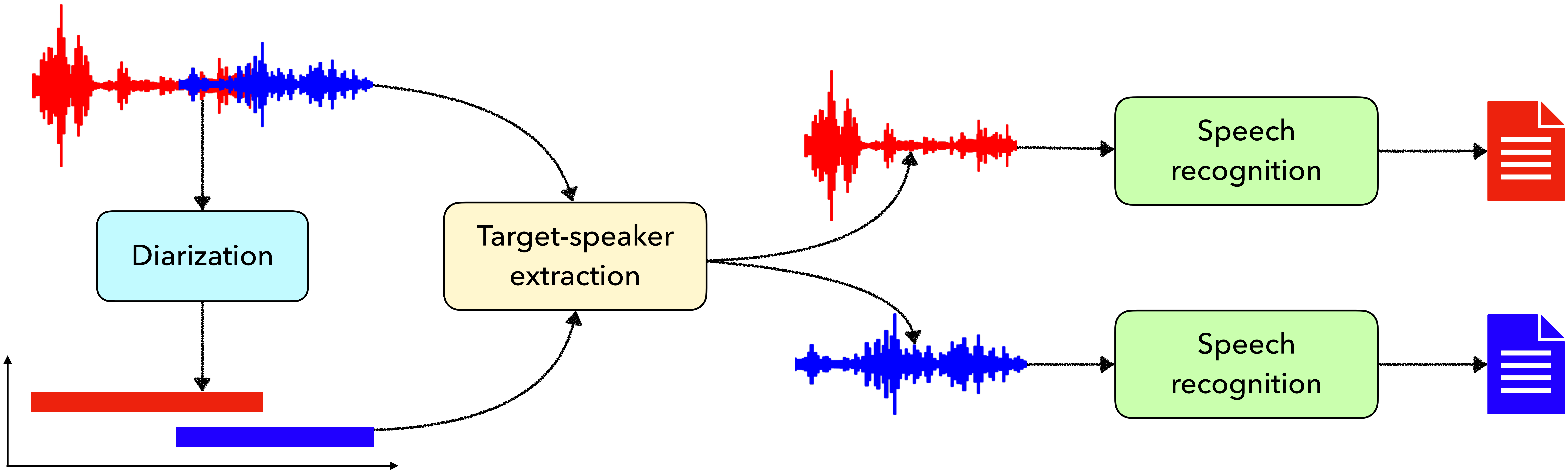
2. "Adapting self-supervised models to multi-talker speech recognition using speaker embeddings." Z. Huang, **D. Raj**, P. Garcia, S. Khudanpur. *IEEE ICASSP 2023*.

End-to-end Multi-talker ASR

Motivation

The problem with modular systems

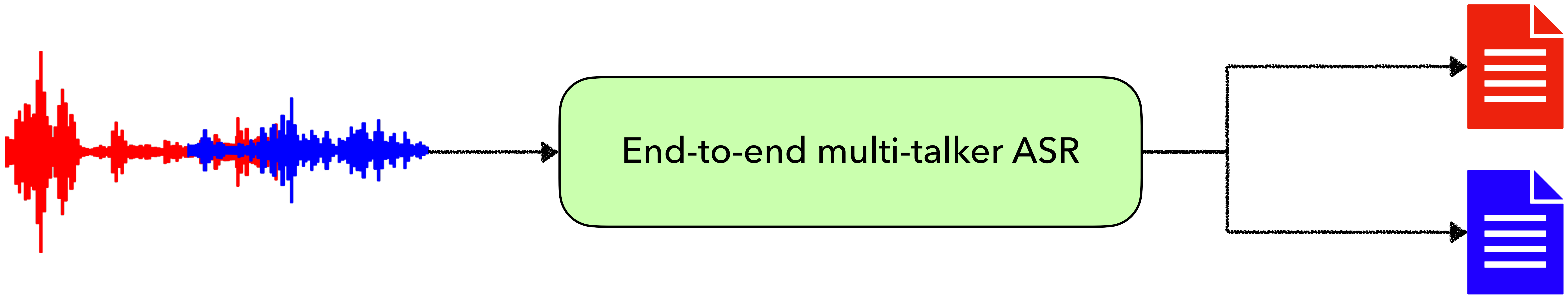
- Modules are independently optimized
- Higher accumulated latency
- Requires engineering efforts to maintain



Motivation

Towards end-to-end multi-talker ASR

- Optimized for end objective
- Single model –> Lower latency
- Easy to maintain and extend



Multi-talker ASR

Streaming Unmixing and Recognition Transducers (SURT)

CONTINUOUS STREAMING MULTI-TALKER ASR WITH DUAL-PATH TRANSDUCERS

Desh Raj[*1], *Liang Lu*[2], *Zhuo Chen*[2], *Yashesh Gaur*[2], *Jinyu Li*[2]

[1]Center for Language and Speech Processing, Johns Hopkins University, USA, [2]Microsoft Corp., USA

Published at IEEE ICASSP 2022

SURT 2.0: Advances in Transducer-based Multi-talker Speech Recognition

Desh Raj, *Student Member, IEEE*, Daniel Povey, *Fellow, IEEE*, and Sanjeev Khudanpur, *Member, IEEE*

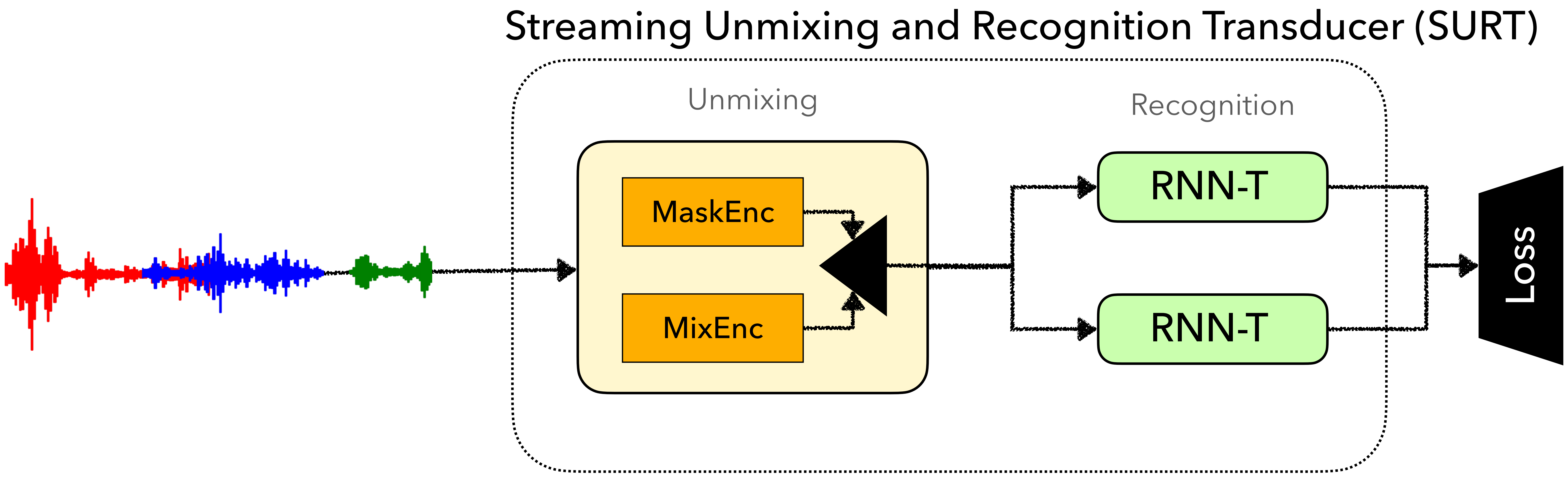
Under review at IEEE TASLP

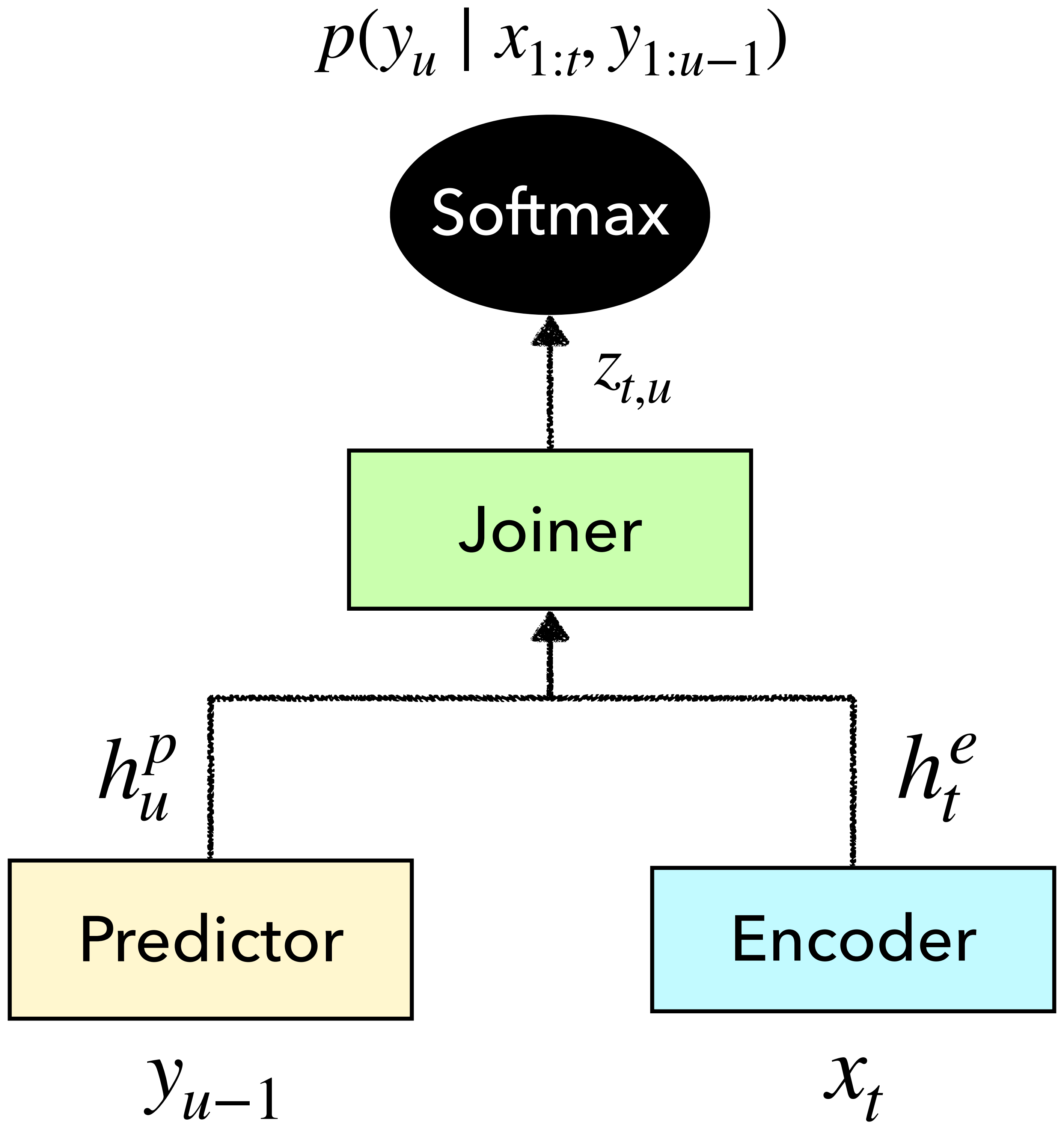
Microsoft

CENTER FOR LANGUAGE AND SPEECH PROCESSING

The original SURT

Basics





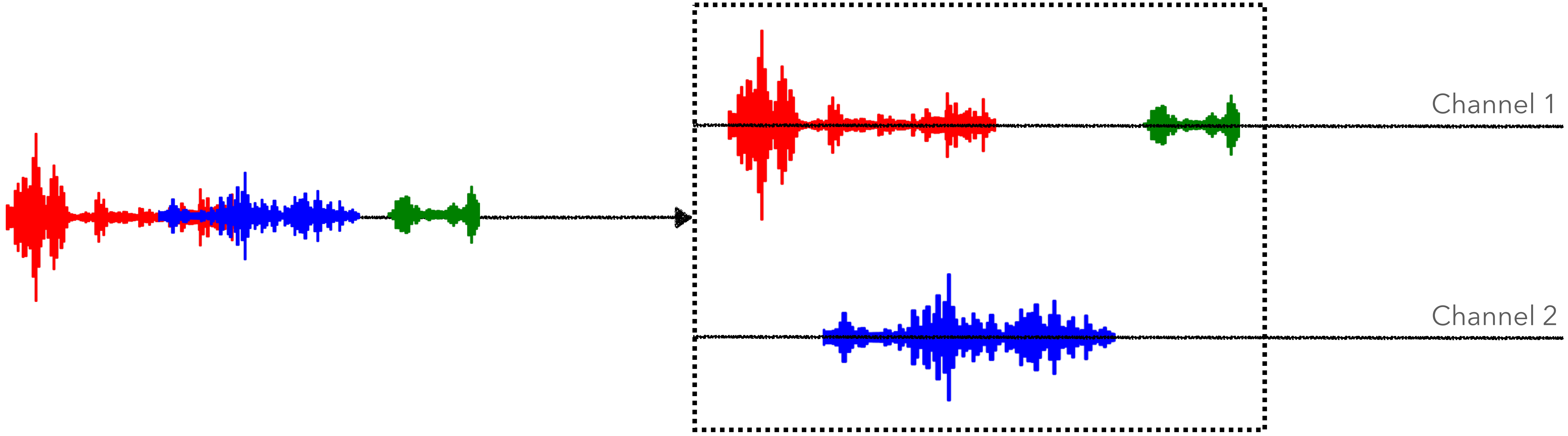
- Unmixing part separates mixed speech into non-overlapping features
- Made of convolutional layers

- Transducer model is used as the recognizer
- Output of unmixing component is fed into encoder
- Use HEAT loss over the transducer loss

SURT objective function

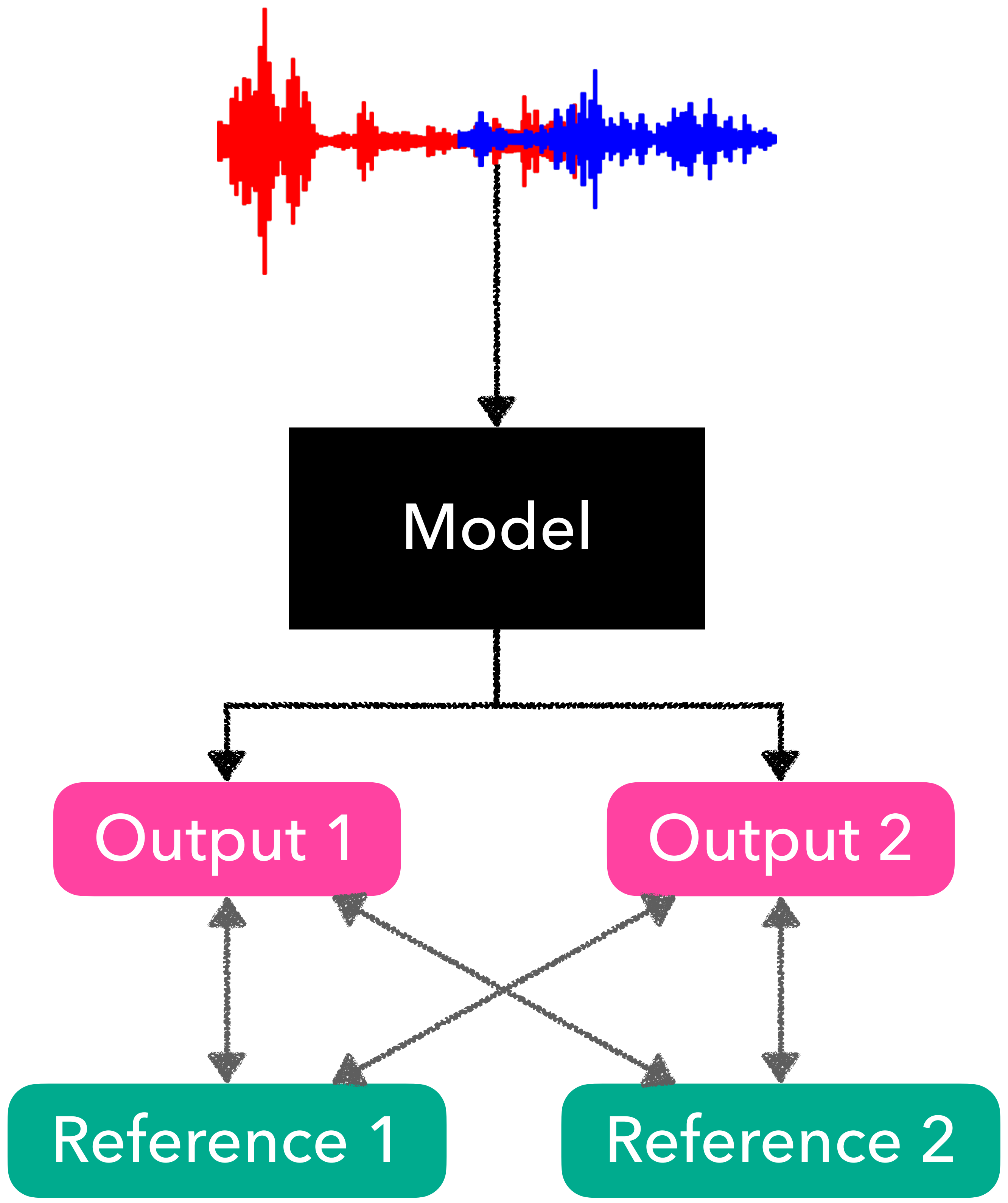
Heuristic error assignment training (HEAT)

- Assign utterances to output channels in order of start time.

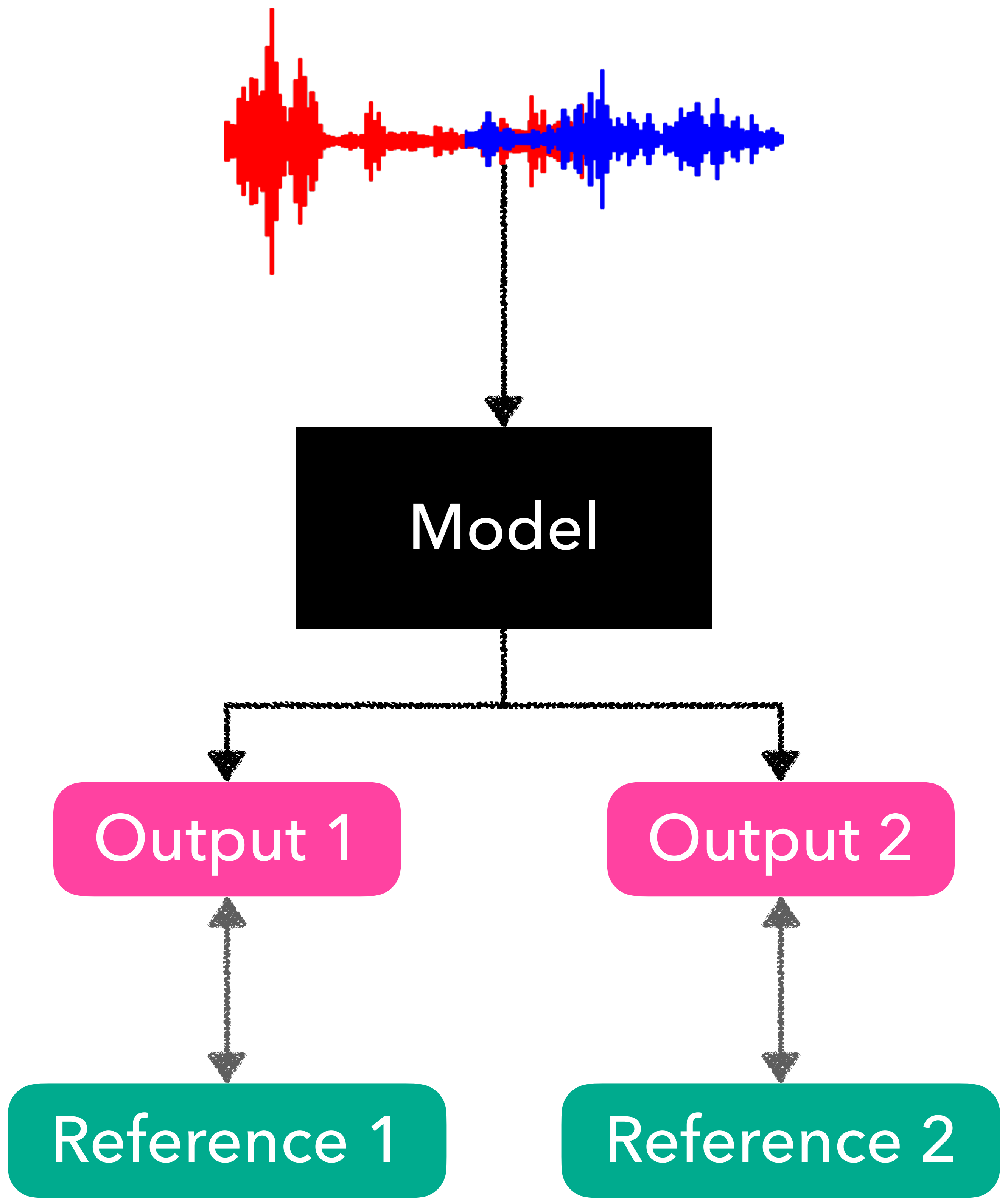


SURT objective function

HEAT vs. PIT



Permutation invariant training (PIT)



Heuristic error assignment training (HEAT)

SURT objective function

HEAT vs. PIT

Permutation invariant training (PIT)

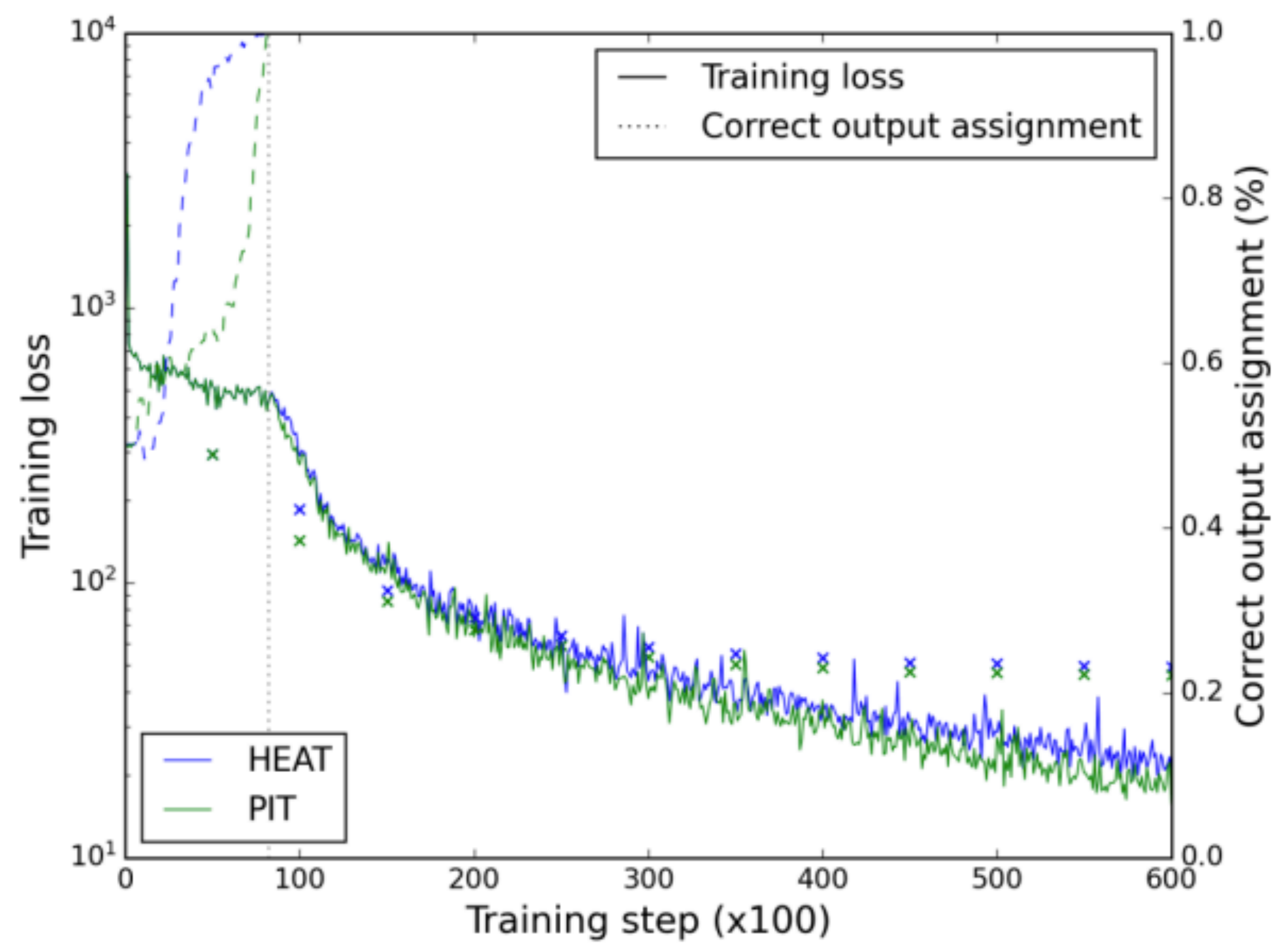
Requires computing all permutations of outputs and references

Can be prohibitively slow when N >> 2 (exponential in N)

Heuristic error assignment training (HEAT)

Requires computing only 1 permutation of output and reference

Complexity increases linearly with N

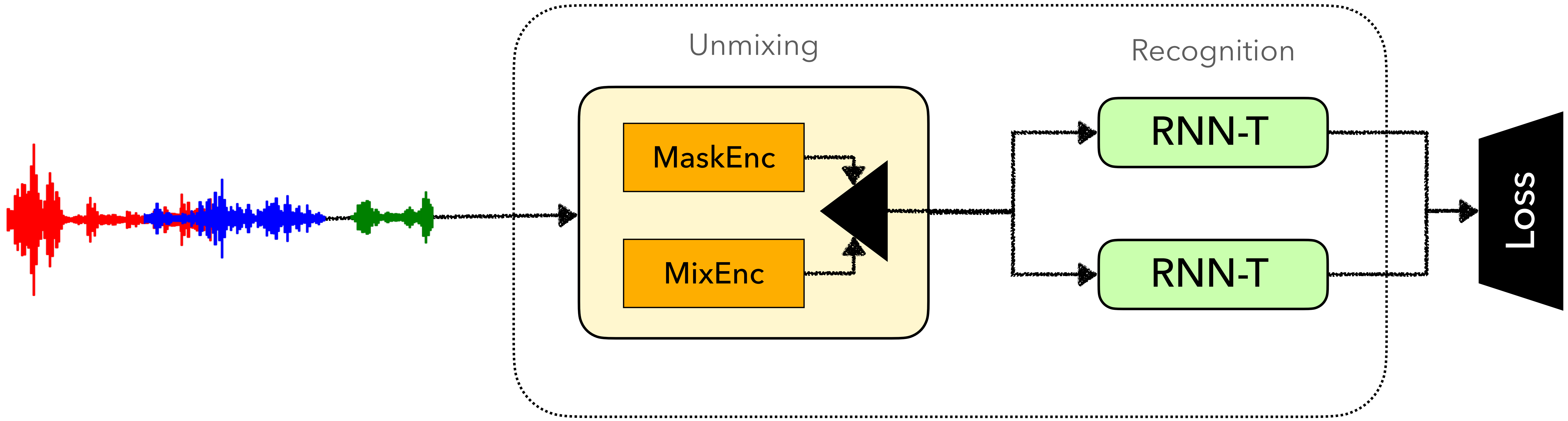


For utterances with non-zero delay, PIT learns the same heuristic as HEAT

The original SURT

Problems

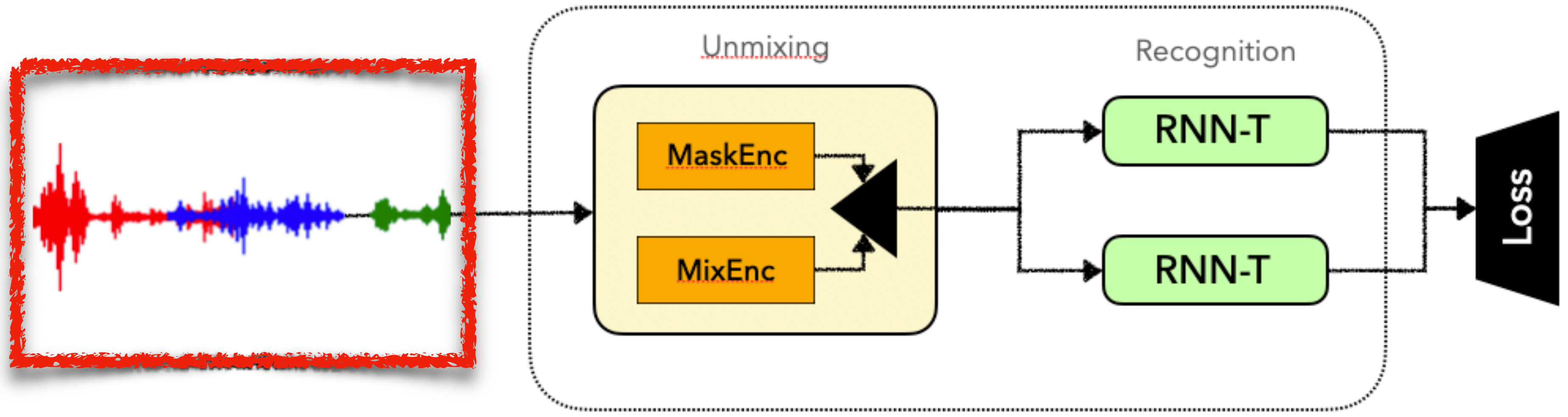
1. Infeasible to train on an academic cluster.
2. Suffers from *omission* and *leakage* errors.
3. How to make it work on real meetings?

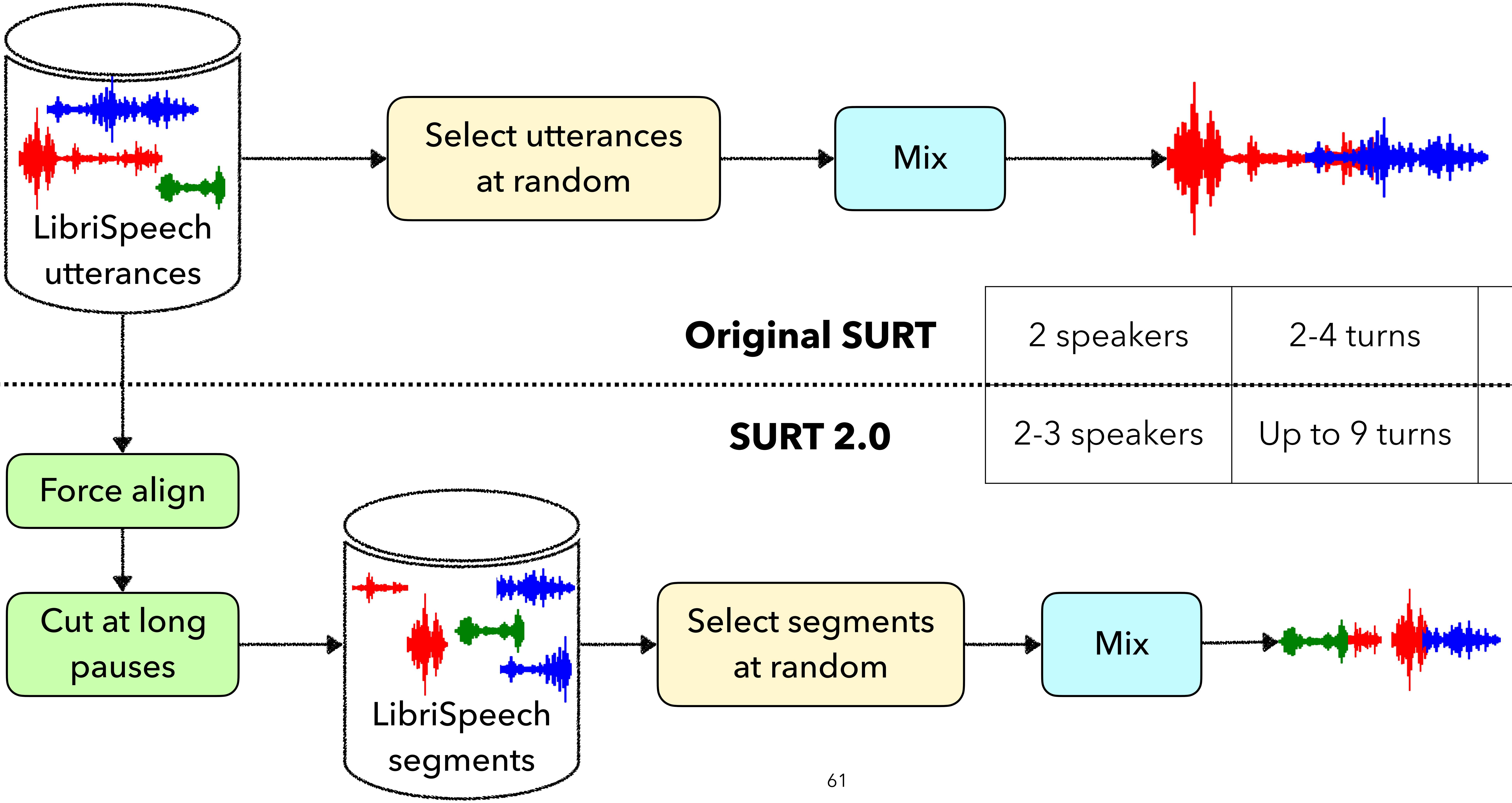


Problem 1

Making training efficient

#1: Shorter training mixtures



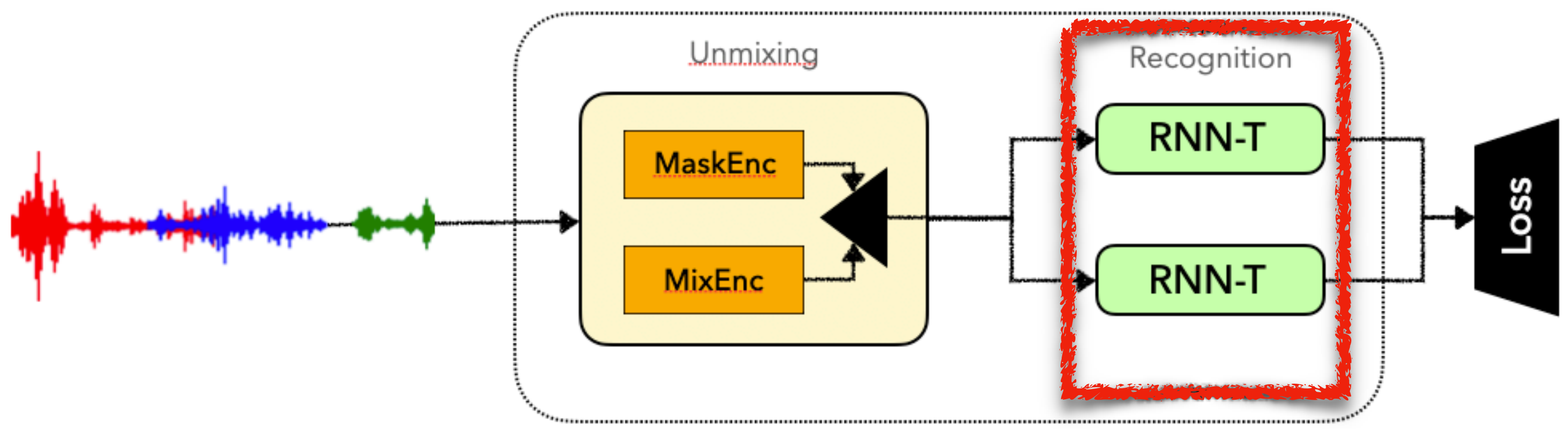


Original SURT	2 speakers	2-4 turns	25.5s
SURT 2.0	2-3 speakers	Up to 9 turns	16.2s

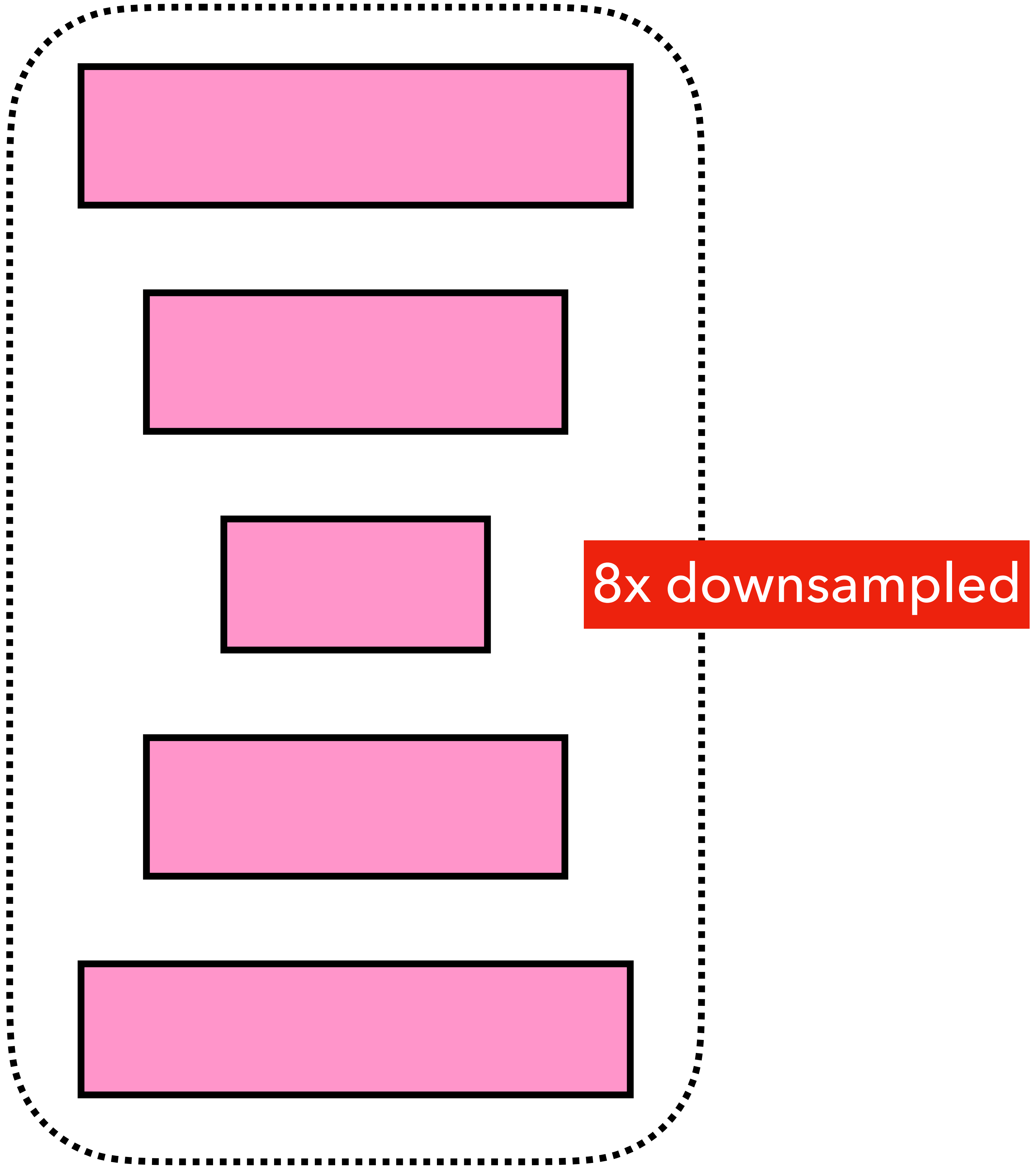
Problem 1

Making training efficient

#2: Zipformer encoder



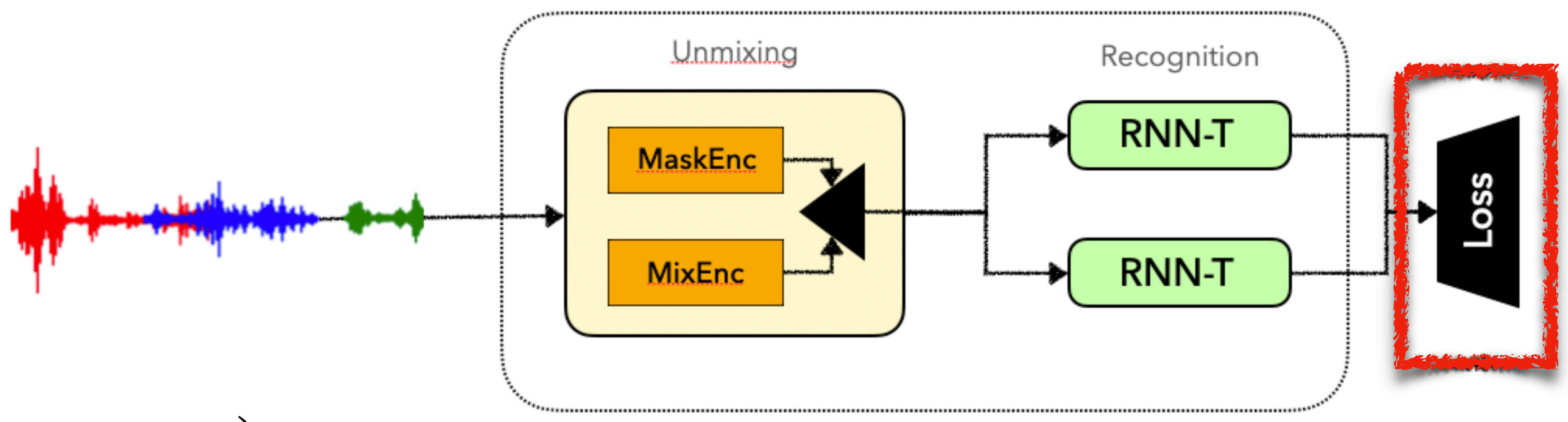
1. Subsampling in intermediate layers
2. Shared self-attention weights in each zipformer "block"
3. Other things (e.g., ScaledAdam)



Problem 1

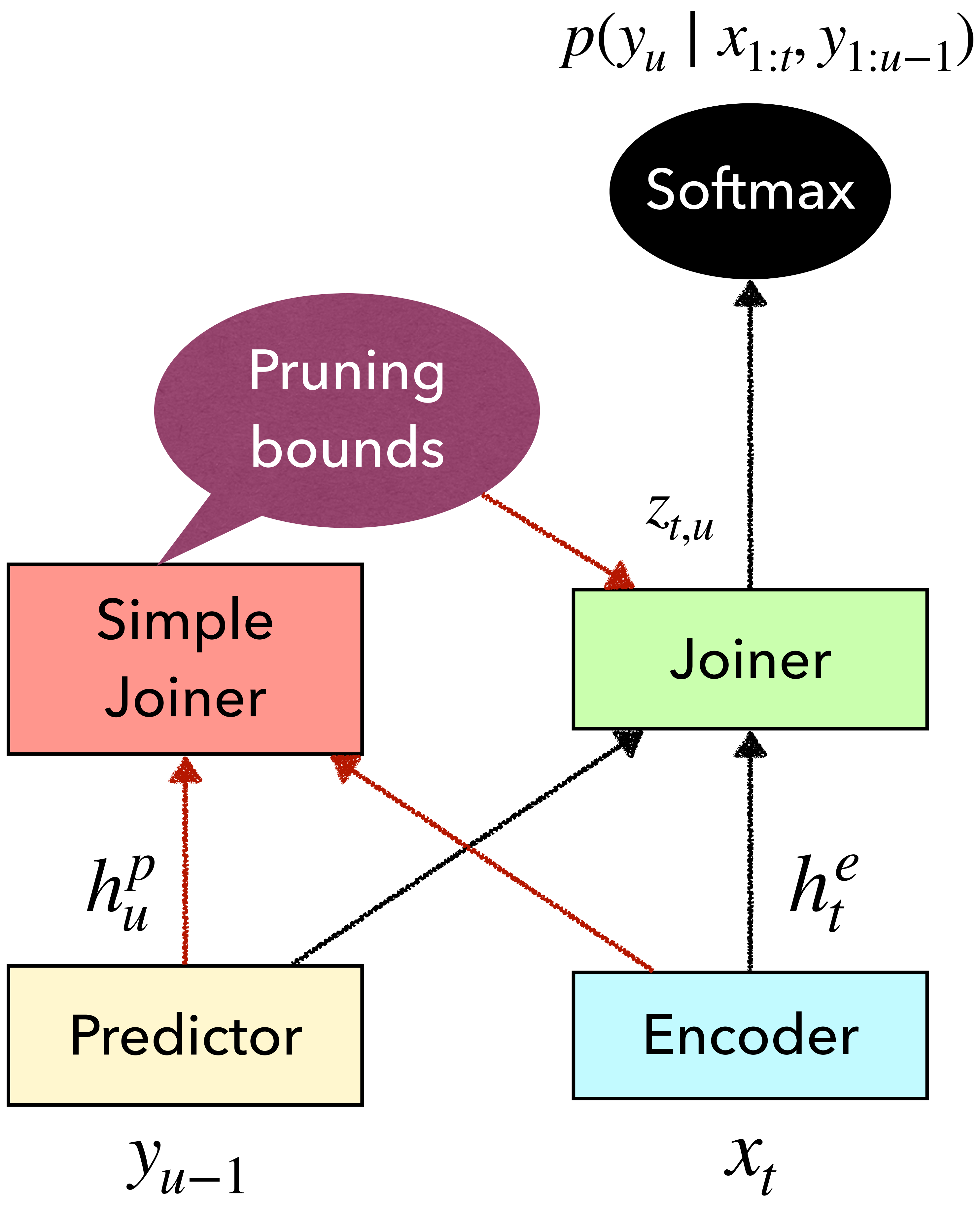
Making training efficient

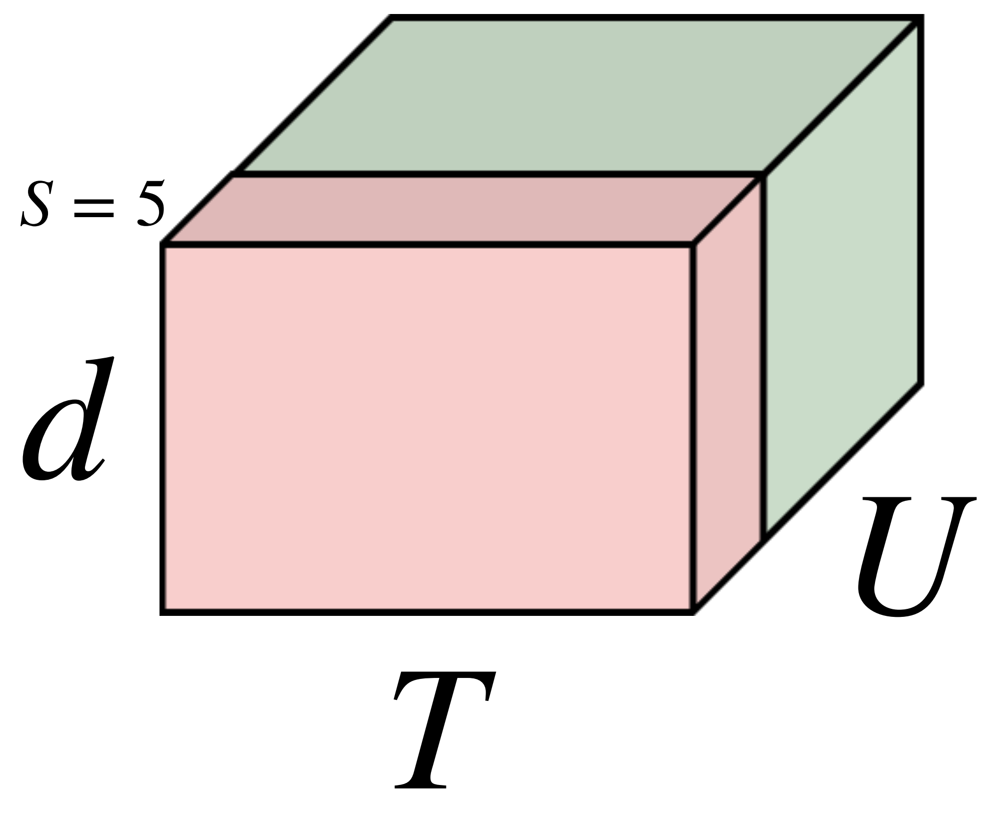
#3: Pruned transducer loss

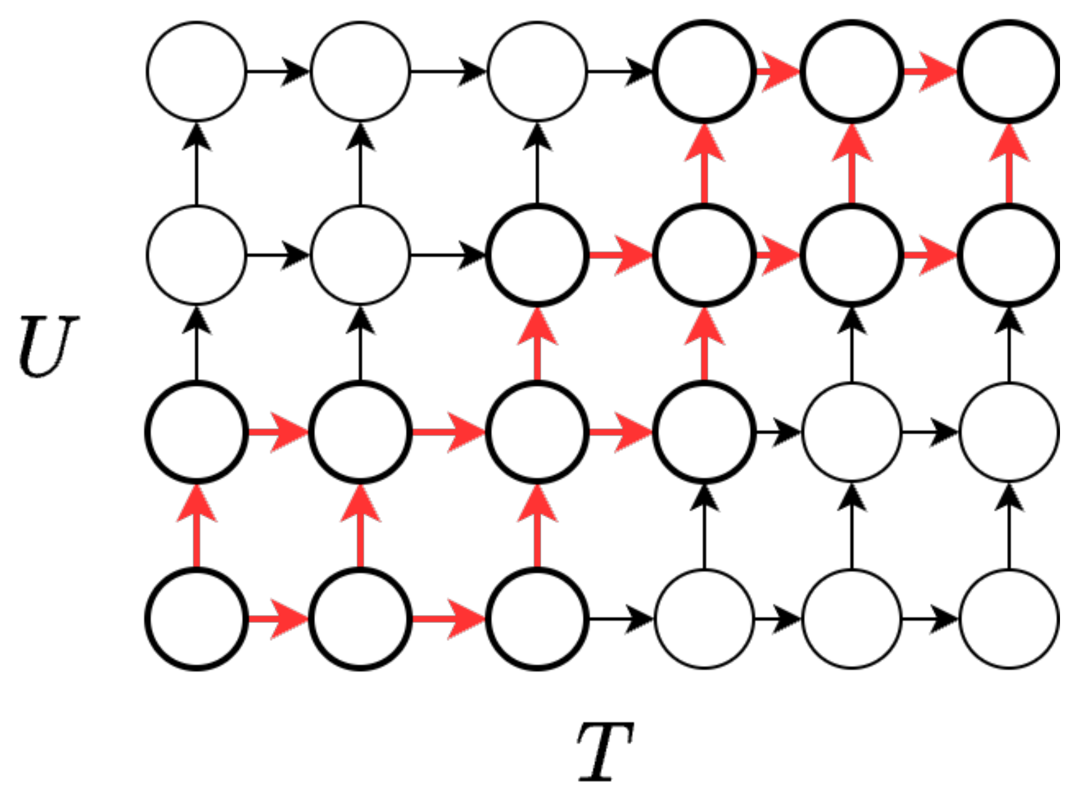


1. Compute pruning range using a "simple" joiner

2. Compute full loss on pruned alignments





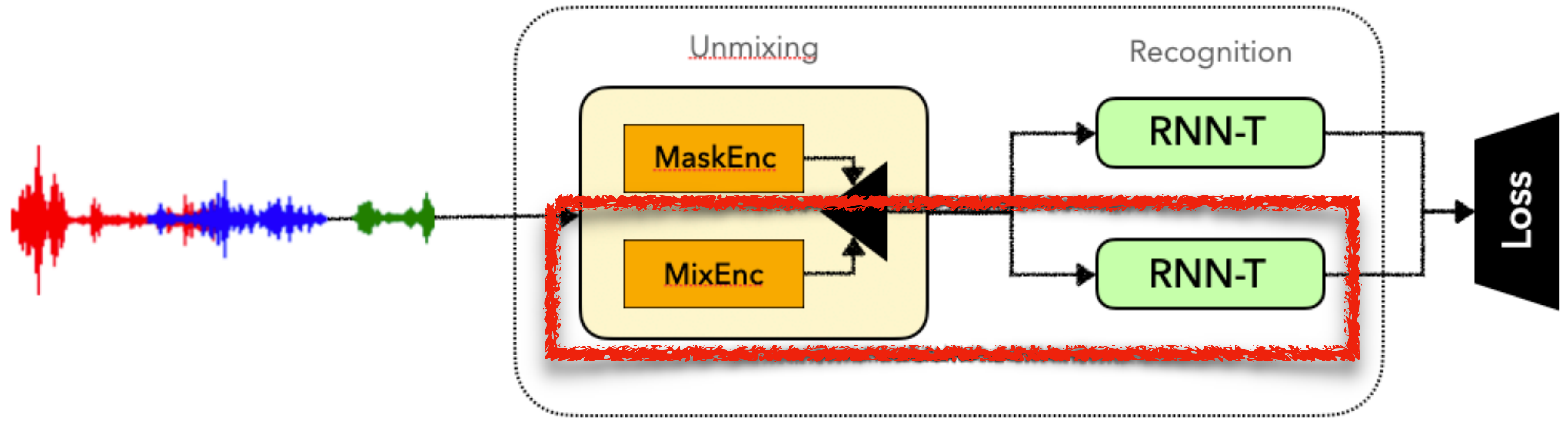


Kuang, F., Guo, L., Kang, W., Lin, L., Luo, M., Yao, Z., & Povey, D. Pruned RNN-T for fast, memory-efficient ASR training. *Interspeech 2022*.

Problem 1

Making training efficient

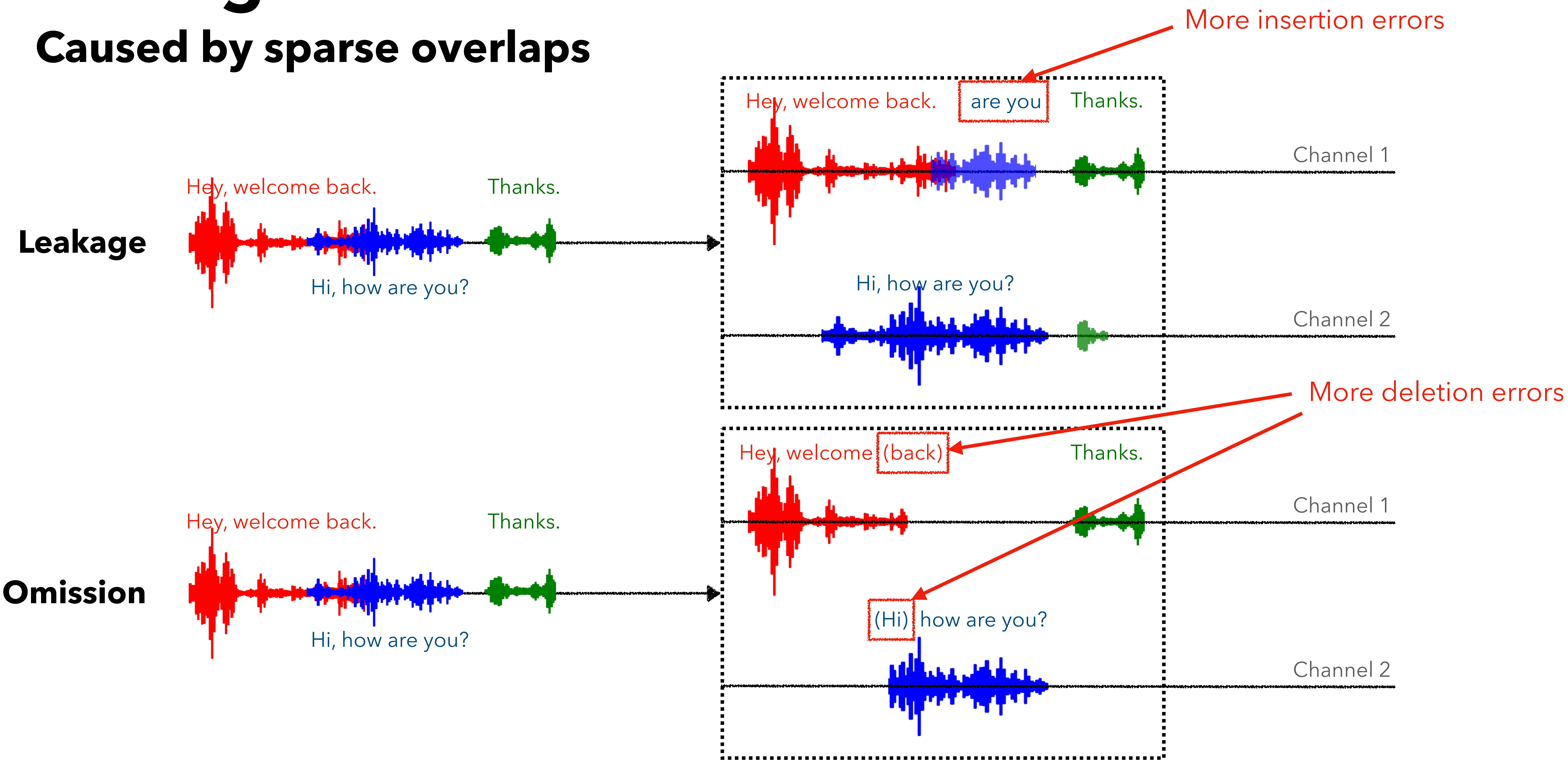
#4: Single-speaker pre-training



Problem 2

Leakage and omission errors

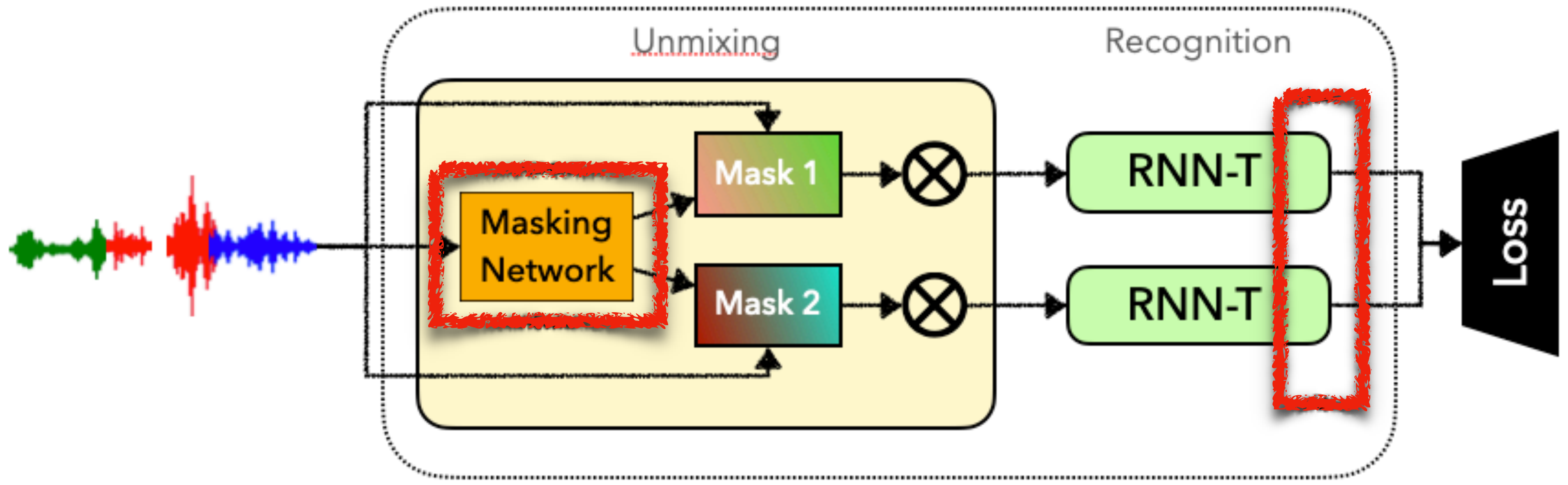
Caused by sparse overlaps



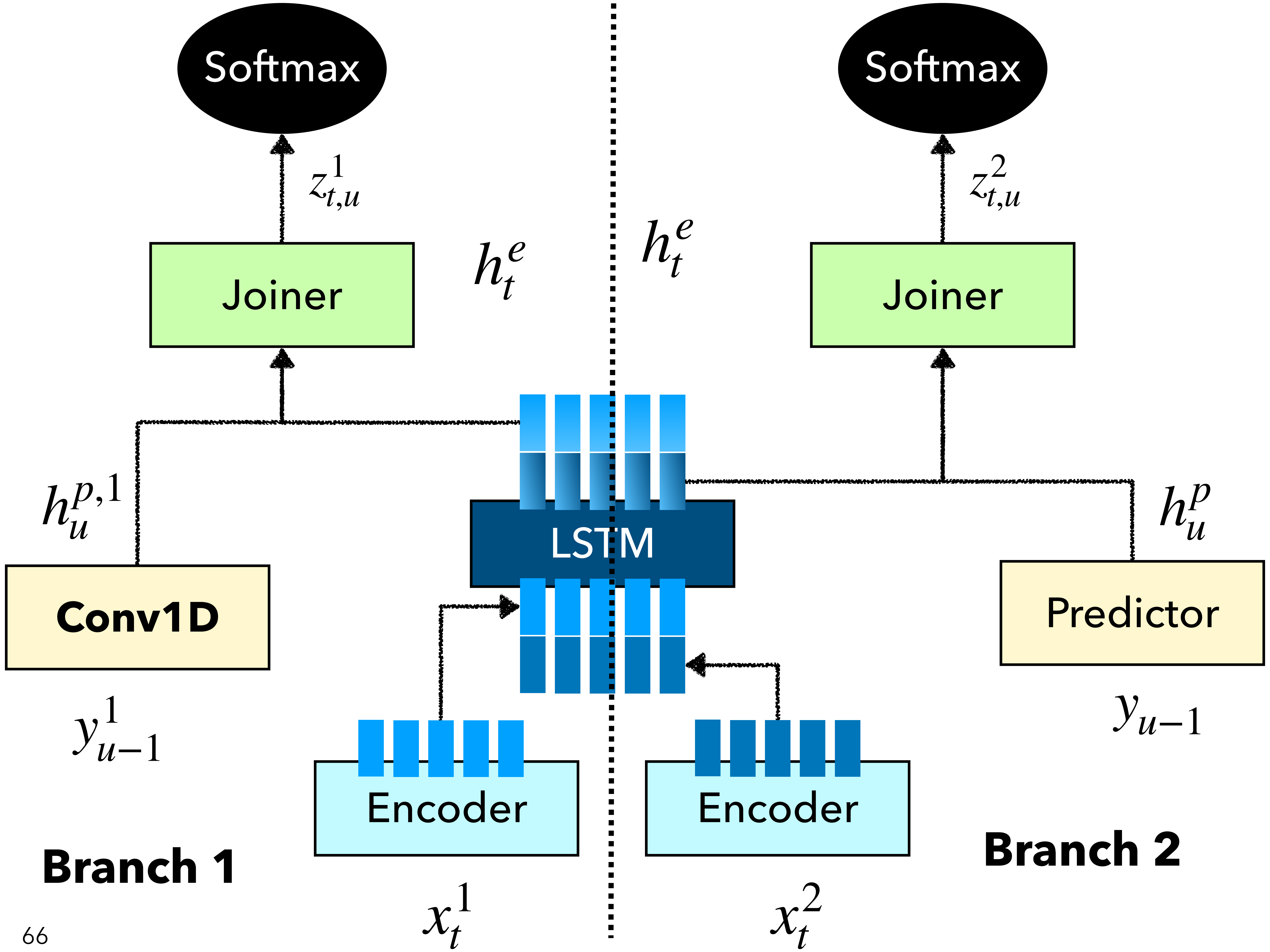
Problem 2

Leakage and omission errors

#1: DP-LSTM, branch tying, stateless decoder

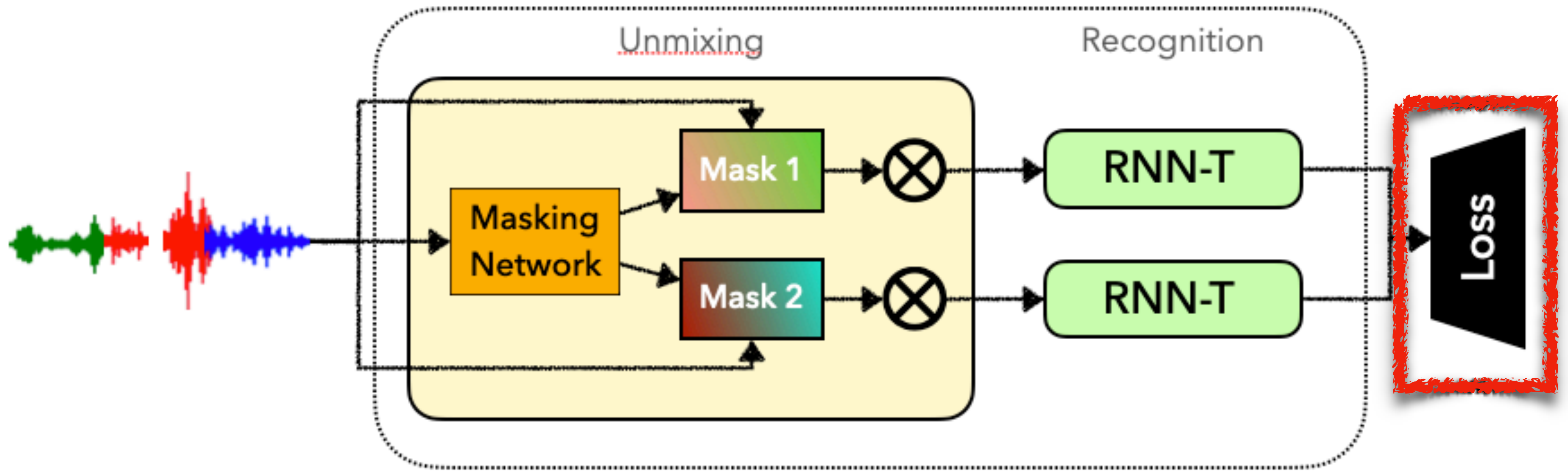


1. Use **dual-path LSTM** instead of Conv2D in masking network
2. Encoder branches are "tied" at the output
3. "Stateless" decoder to improve short turn-taking



Problem 2

Leakage and omission errors



#2: Masking loss and encoder CTC loss

We use 2 auxiliary loss functions:

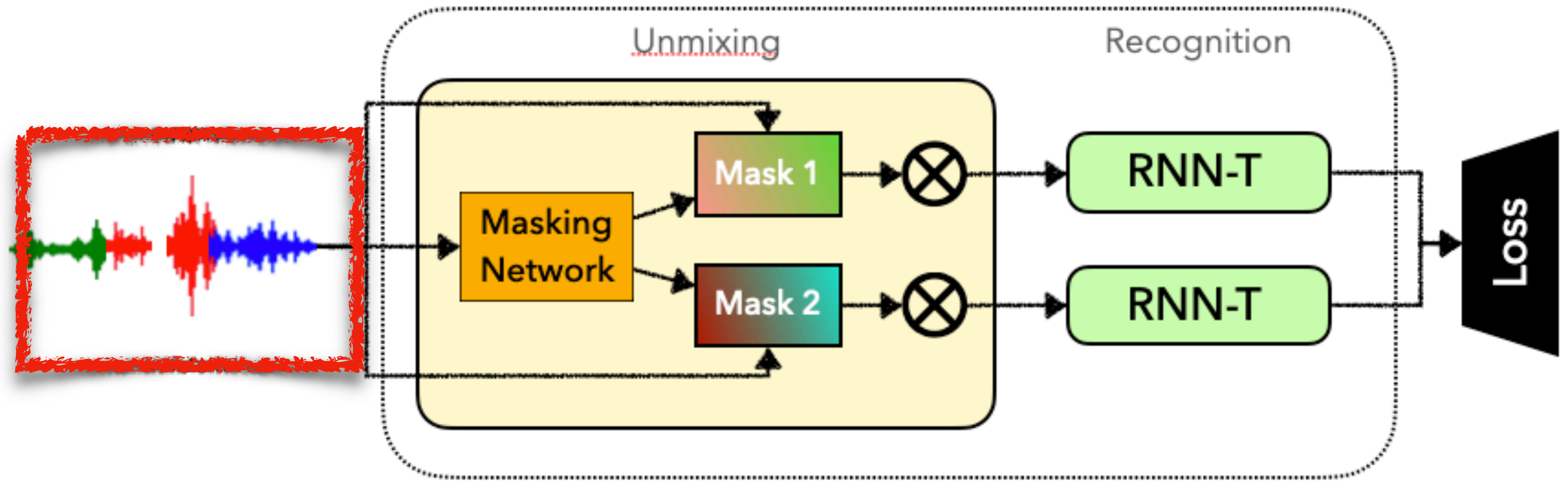
1. **CTC loss** at the output of the encoder (for better alignment)
2. **MSE loss** on the masked filterbanks (for better separation)

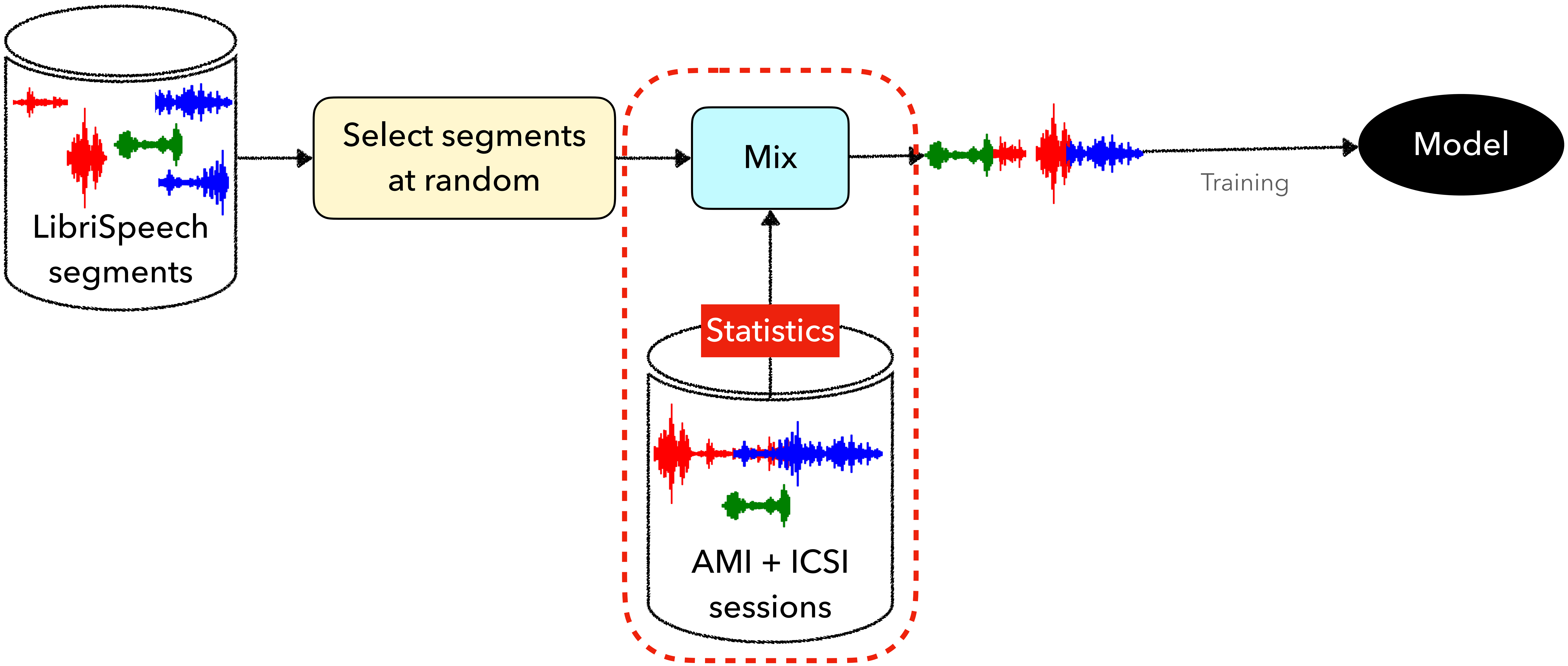
$$\mathscr{L} = \mathscr{L}'_{\mathrm{rnnt}} + \lambda_{\mathrm{ctc}}\mathscr{L}_{\mathrm{ctc}} + \lambda_{\mathrm{mask}}\mathscr{L}_{\mathrm{mask}}$$

Problem 3

Performance on real meetings

#1: Simulation using real meeting statistics

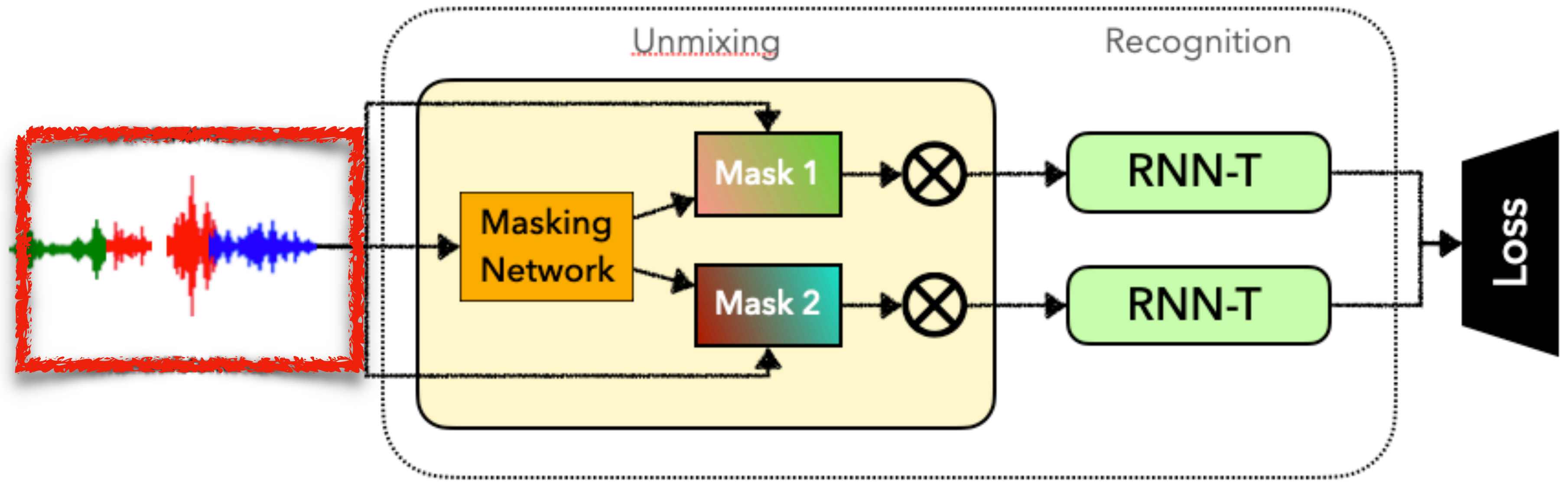


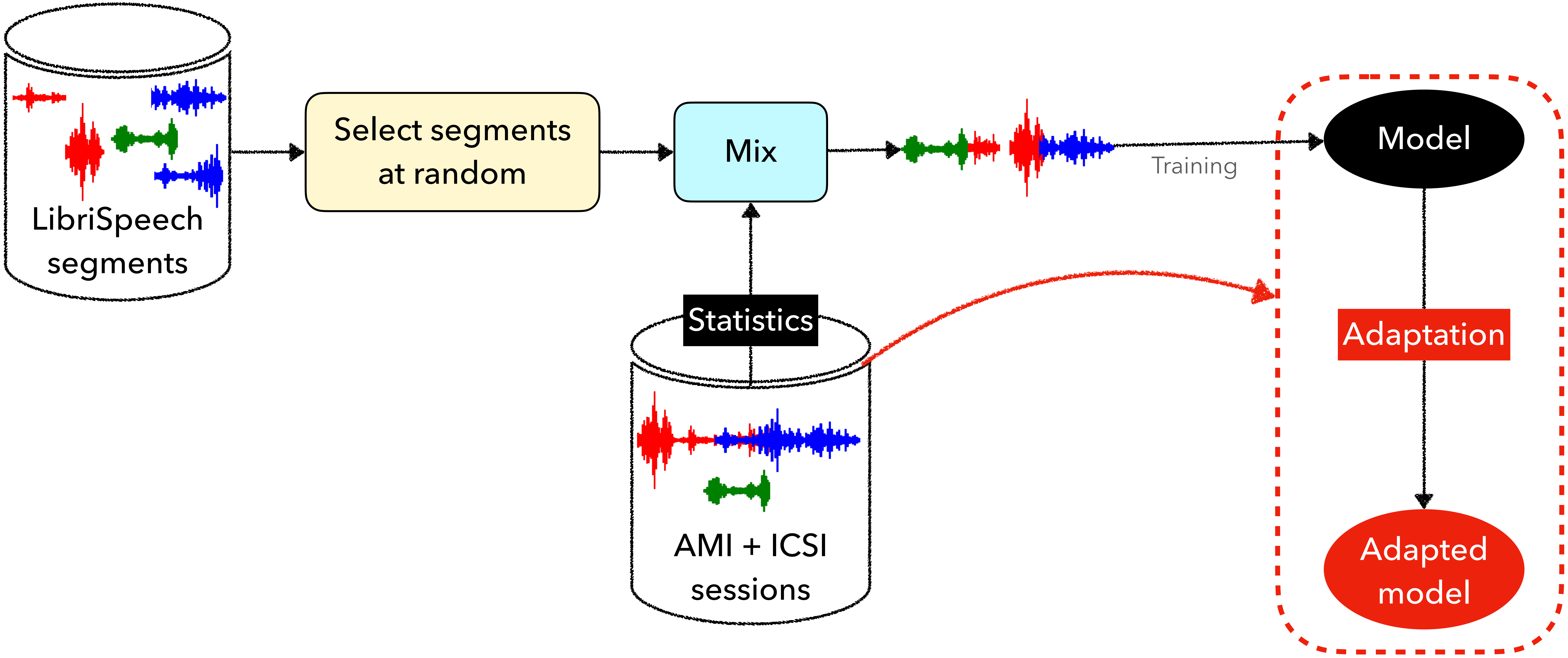


Problem 3

Performance on real meetings

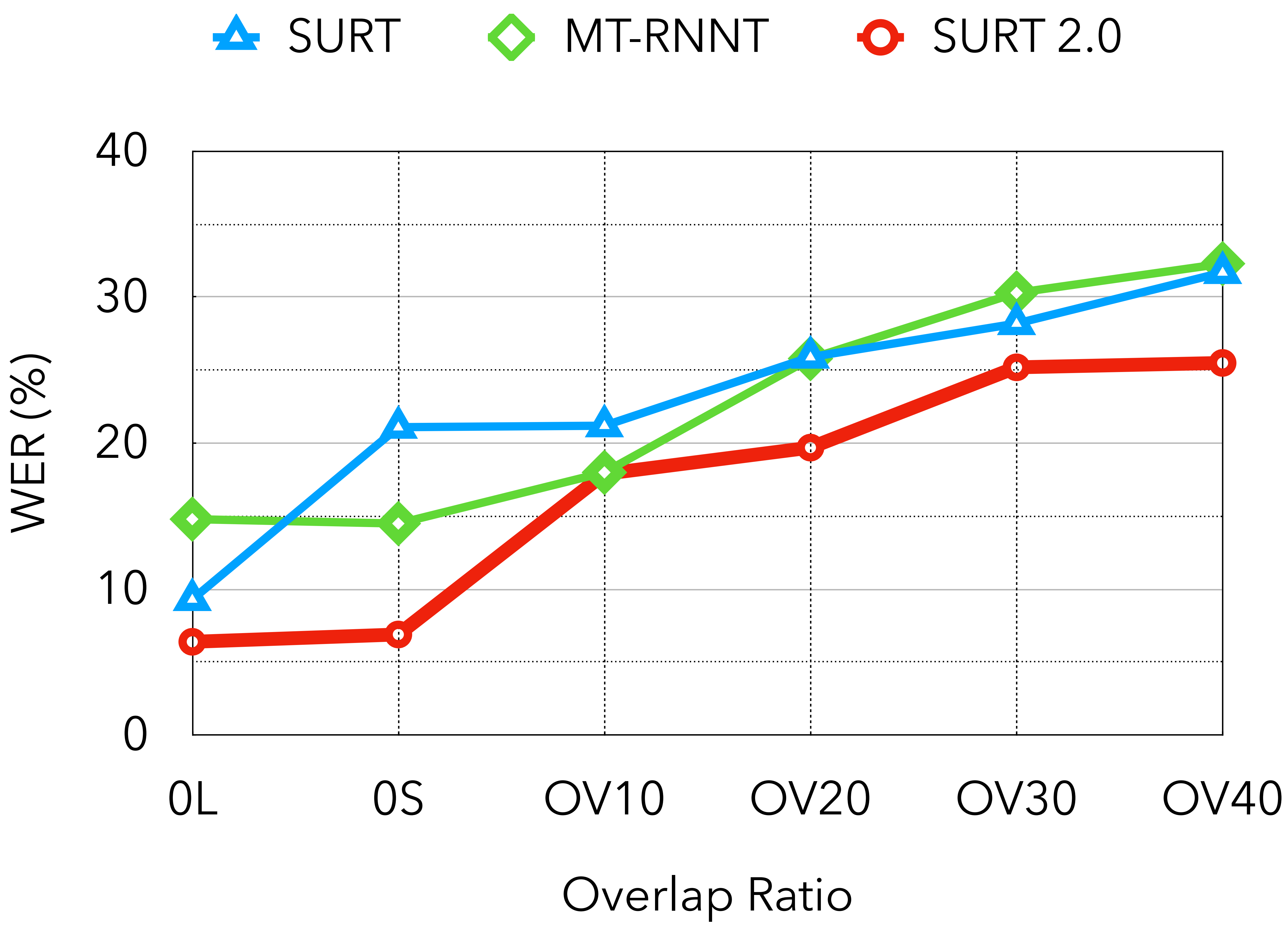
#2: Domain adaptation





Results on LibriCSS

#1: SURT 2.0 outperforms original SURT

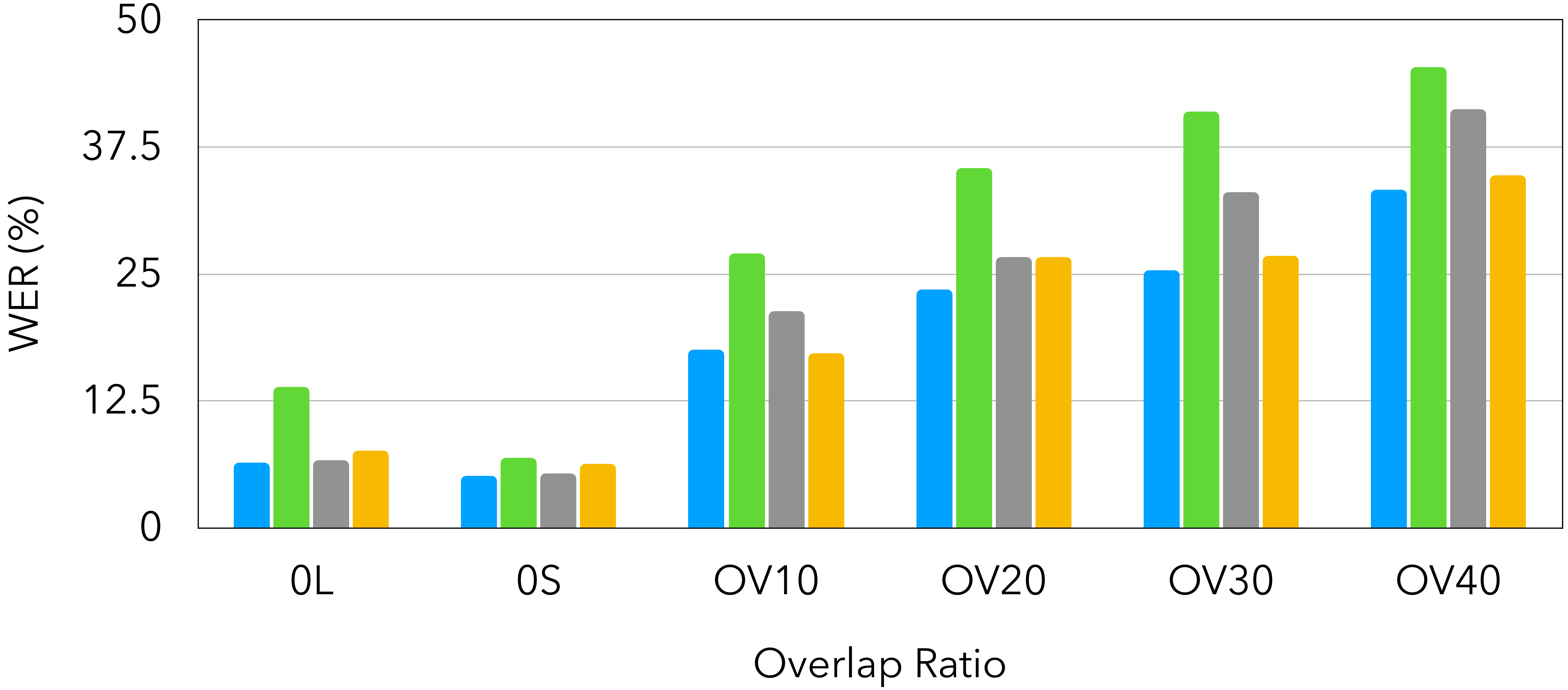


Model	# params (M)	WER (%)	
SURT	42.9	22.9	16 x V100
MT-RNNT	81.0	22.6	
SURT 2.0	**37.9**	**16.9**	4 x V100

Results on LibriCSS

#2: Effect of architectural changes

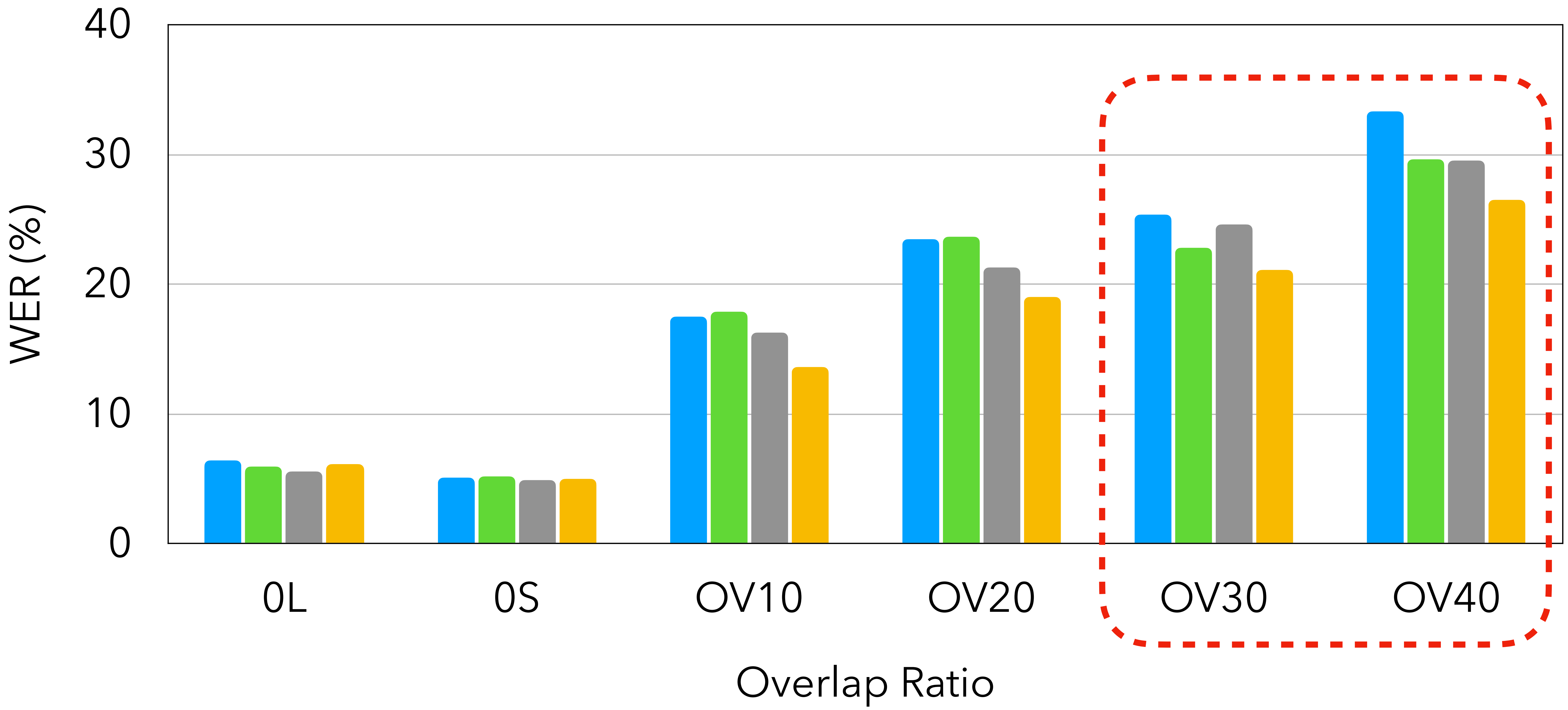
- Most improvement comes from using DP-LSTM in masking network.



	WER (%)
SURT 2.0	18.5
w/o DP-LSTM	28.3
w/o branch tying	22.4
w/o stateless decoder	19.9

Results on LibriCSS

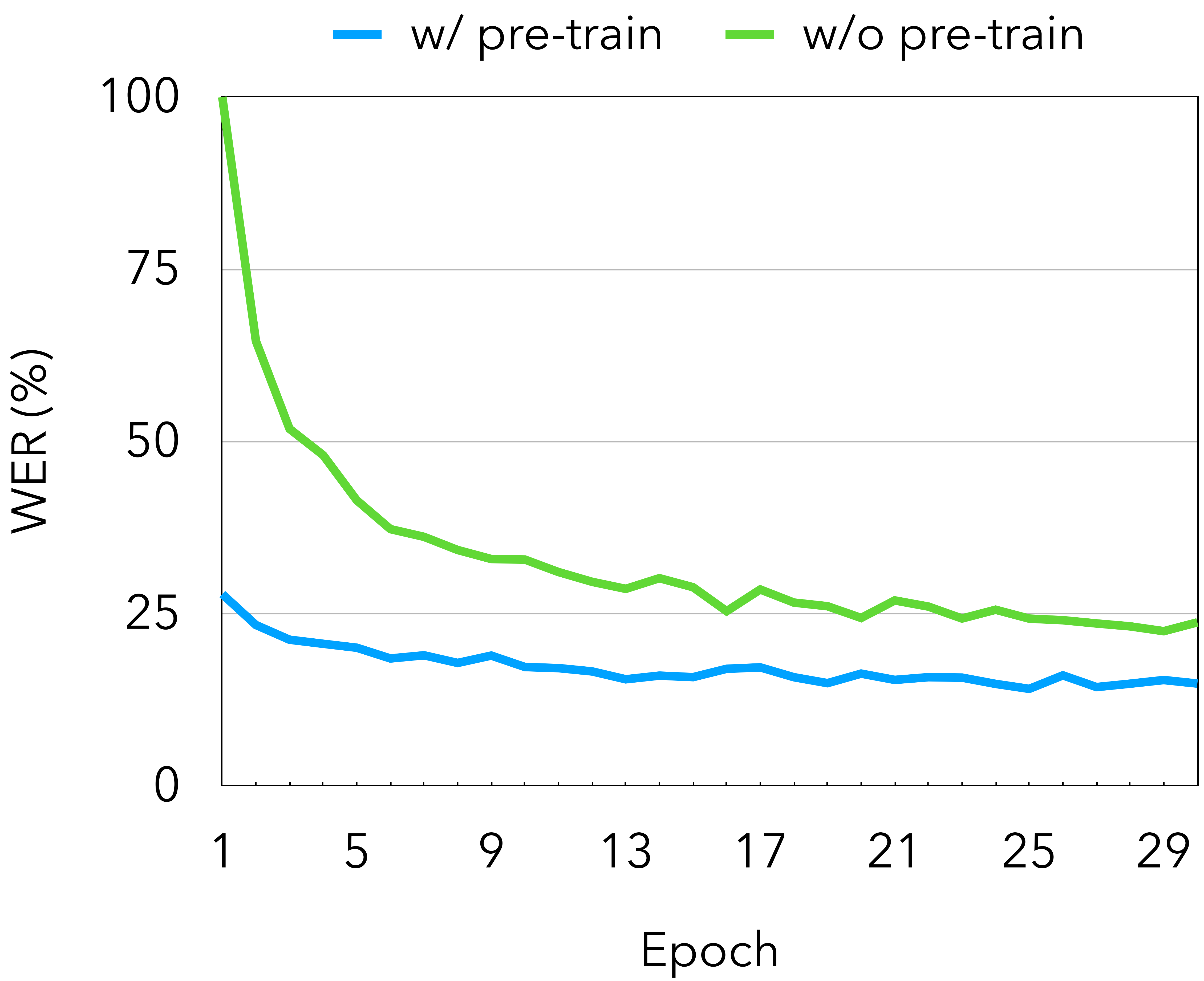
#3: Effect of auxiliary objectives



	WER (%)
No aux. loss	18.5
+ CTC loss	17.5
+ Mask loss	17.1
+ CTC + Mask	15.2

Results on LibriCSS

#4: Single speaker pre-training is critical



Results on real meetings

AMI and ICSI

AMI

	IHM-Mix	SDM	MDM (beamform)
SURT 2.0	36.8	62.5	44.4
+ adapt.	35.1	44.6	41.4

ICSI

	IHM-Mix	SDM
SURT 2.0	27.8	59.7
+ adapt.	24.4	32.2

End-to-end multi-talker ASR

Next steps

1. How to perform word-level speaker attribution?
2. Decoding/rescoring across branches
3. Can we use pre-trained self-supervised models for the encoder?

Multi-talker ASR + diarization

Further reading

1. "The JHU multi-microphone multi-speaker ASR system for the CHiME-6 challenge." A. Arora*, **D. Raj***, A. S. Subramanian*, K. Li*, B. Benyair, M. Maciejewski, P. Zelasko, P. Garcia, S. Watanabe, S. Khudanpur. *The 6th CHiME Workshop, 2020*.

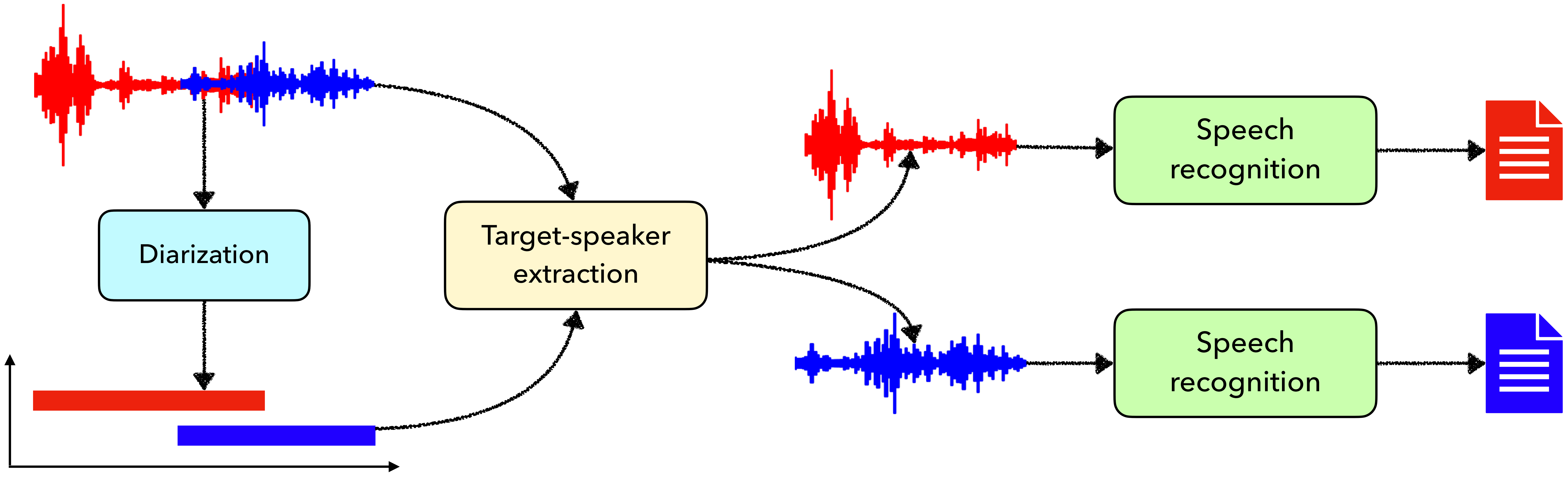
2. "Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis." **D. Raj**, P. Denisov, Z. Chen, H. Erdogan, Z. Huang, M. He, S. Watanabe, J. Du, T. Yoshioka, Y. Luo, N. Kanda, J. Li, S. Wisdom, J. R. Hershey. *IEEE SLT 2021*.

3. "Joint speaker diarization and speech recognition based on region proposal networks." Z. Huang, M. Delcroix, P. Garcia, S. Watanabe, **D. Raj**, S. Khudanpur. *Computer, Speech, and Language, Vol. 72*.

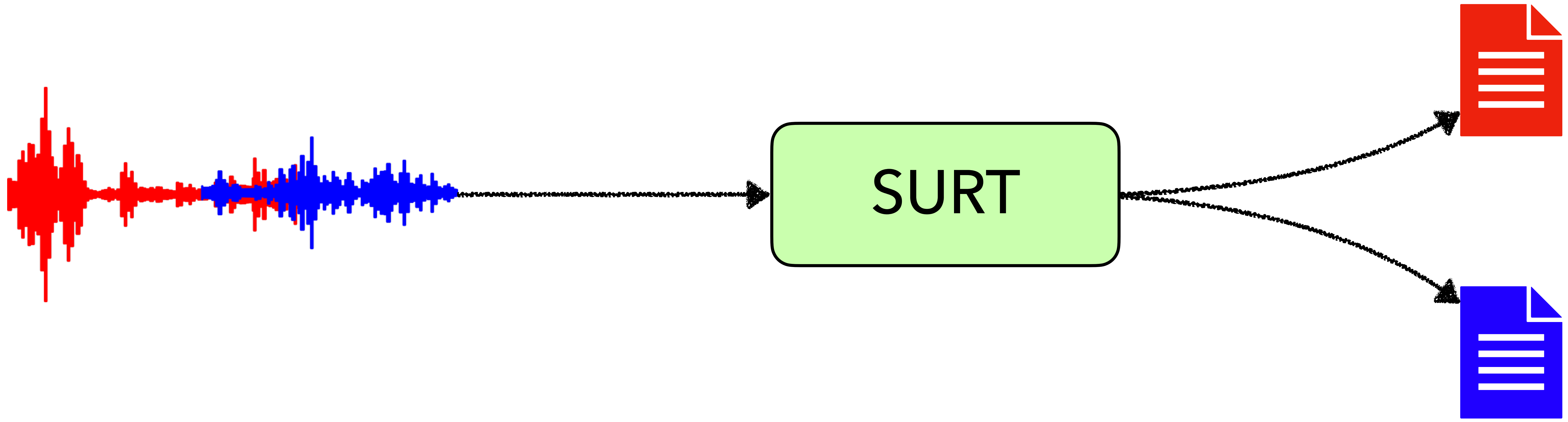
4. "The CHiME-7 DASR Challenge: Distant Meeting Transcription with Multiple Devices in Diverse Scenarios." S. Cornell, M. Wiesner, S. Watanabe, **D. Raj**, X. Chang, P. García, Y. Masuyama, Z. Wang, S. Squartini, and S. Khudanpur. ArXiv, 2023.

- **Overlap-aware** spectral clustering
- Ensemble methods

- GPU-accelerated **GSS** for multi-channel extraction
- Using wake-words for **target-speaker ASR**



Questions?



- **Efficient training** of transducer-based multi-talker ASR
- Improvements in modeling, mixture simulation, and training strategies