Listening to Multi-talker Conversations

Modular and End-to-end Perspectives

PhD Thesis Defense

Desh Raj

January 26, 2024

Motivation

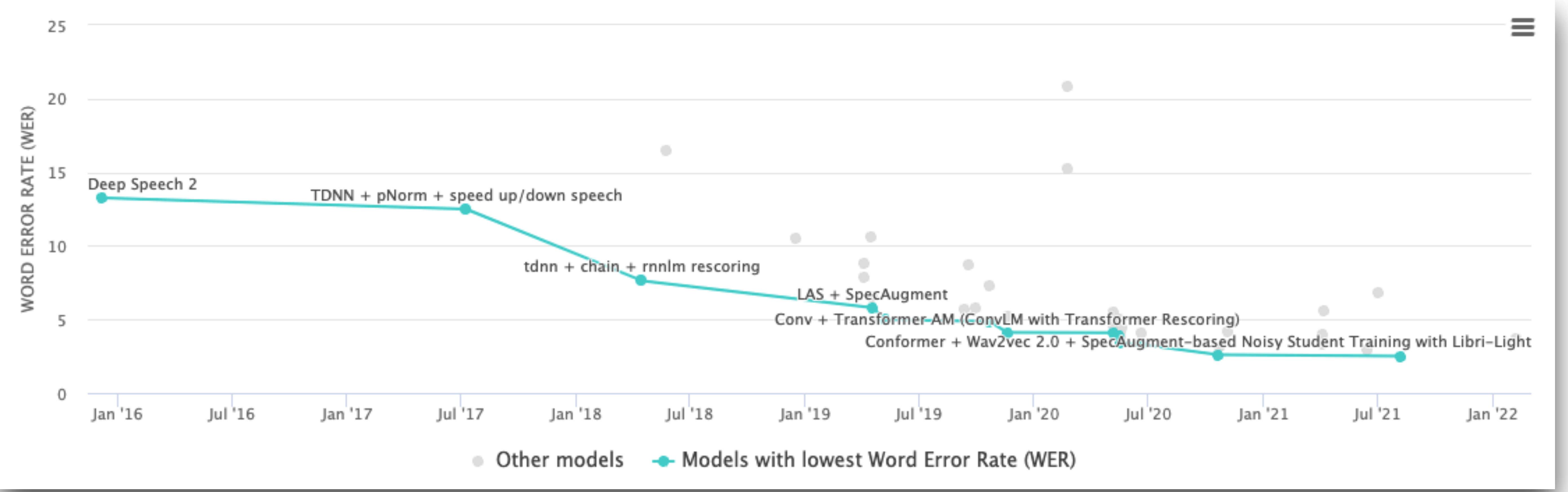
OCTOBER 20, 2020

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

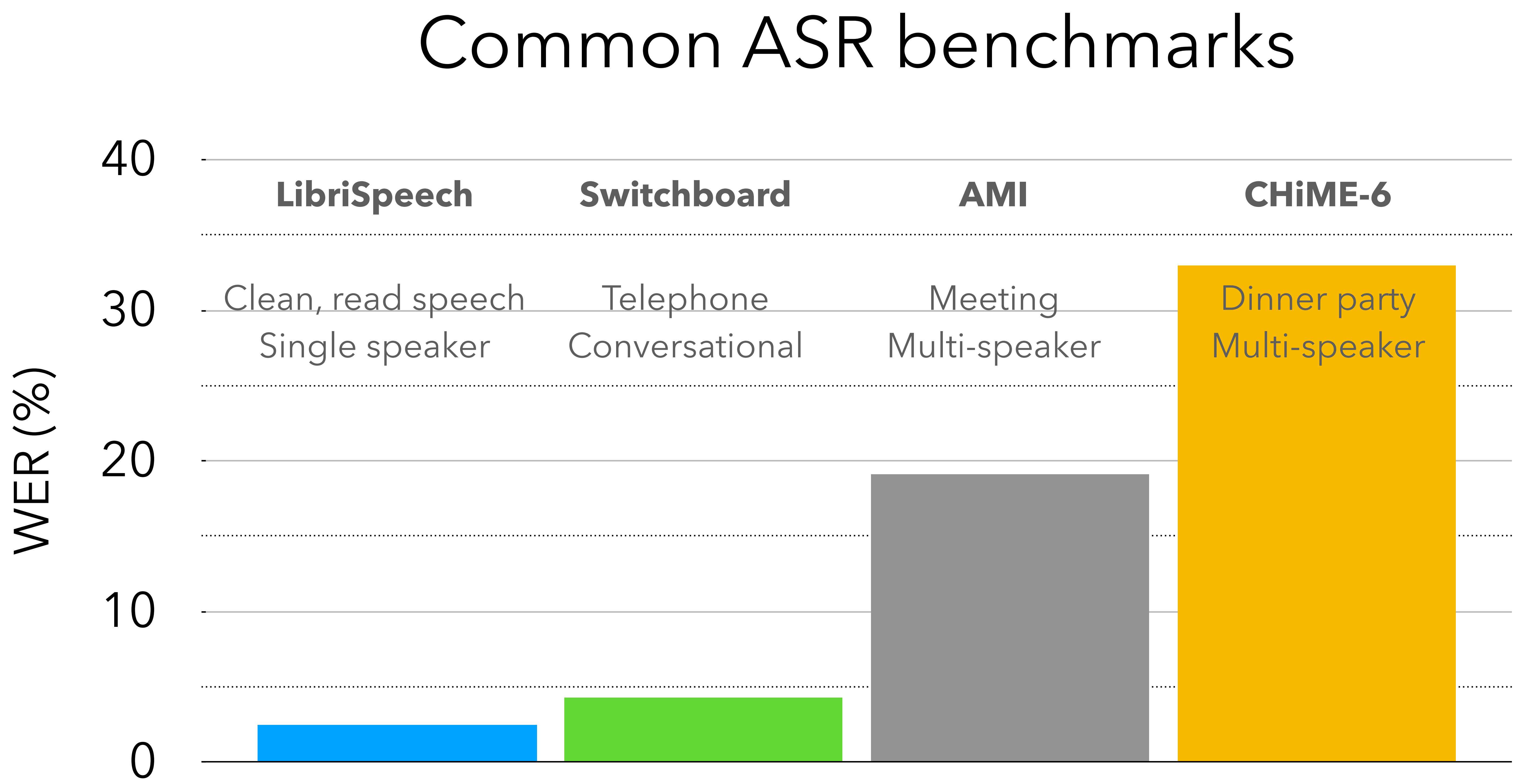
Multi-user applications

Meeting summaries

Collaborative Learning

Child language development

Motivation

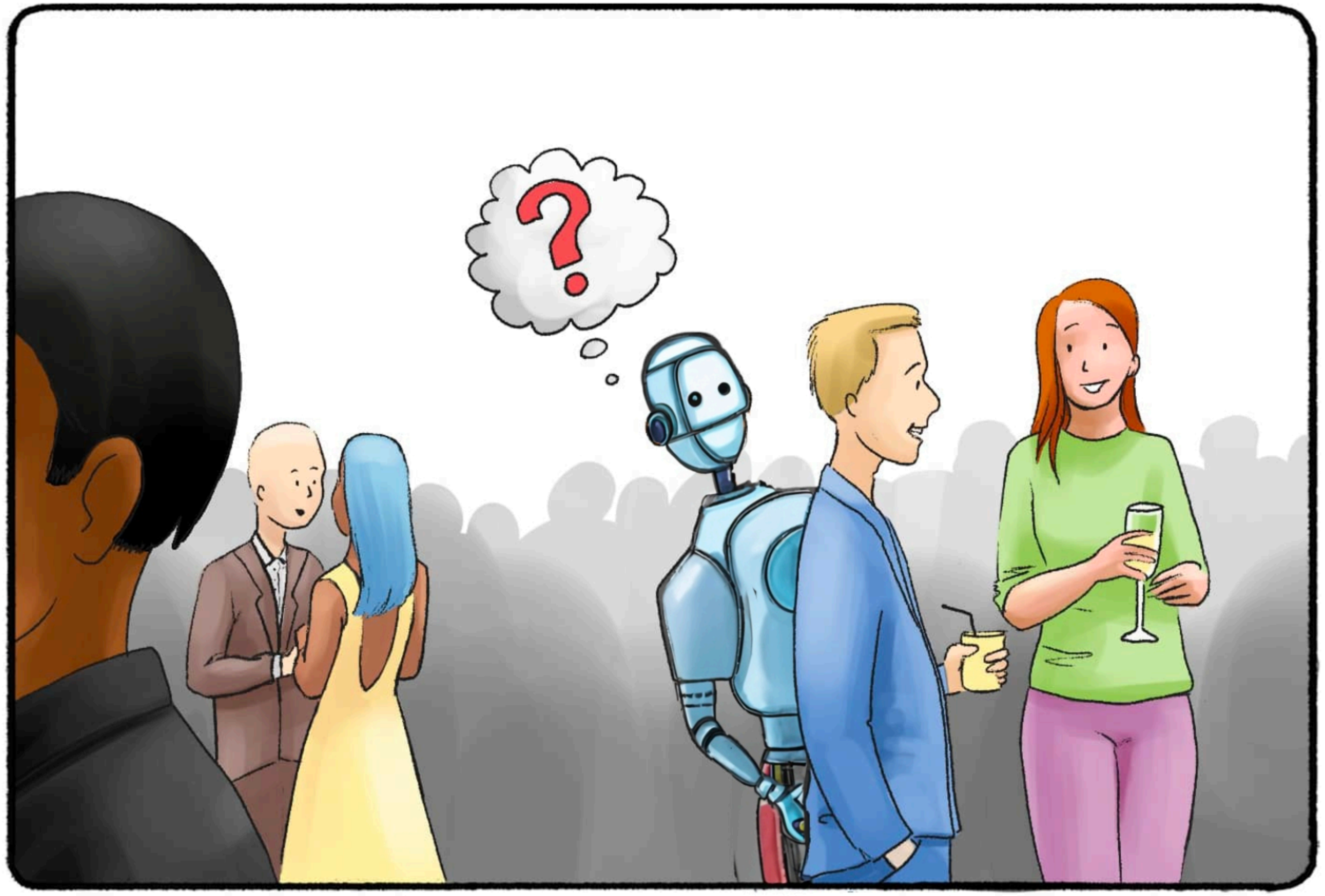


What changed?

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

Motivation

The Cocktail Party Problem



Tasks within the Cocktail Party

Tasks within the Cocktail Party

Speaker Diarization

Raj et al. *Probing the infomation encoded in x-vectors*. **IEEE ASRU 2019**.

Raj et al. *Multi-class spectral clustering with overlaps for speaker diarization*. **IEEE SLT 2021**.

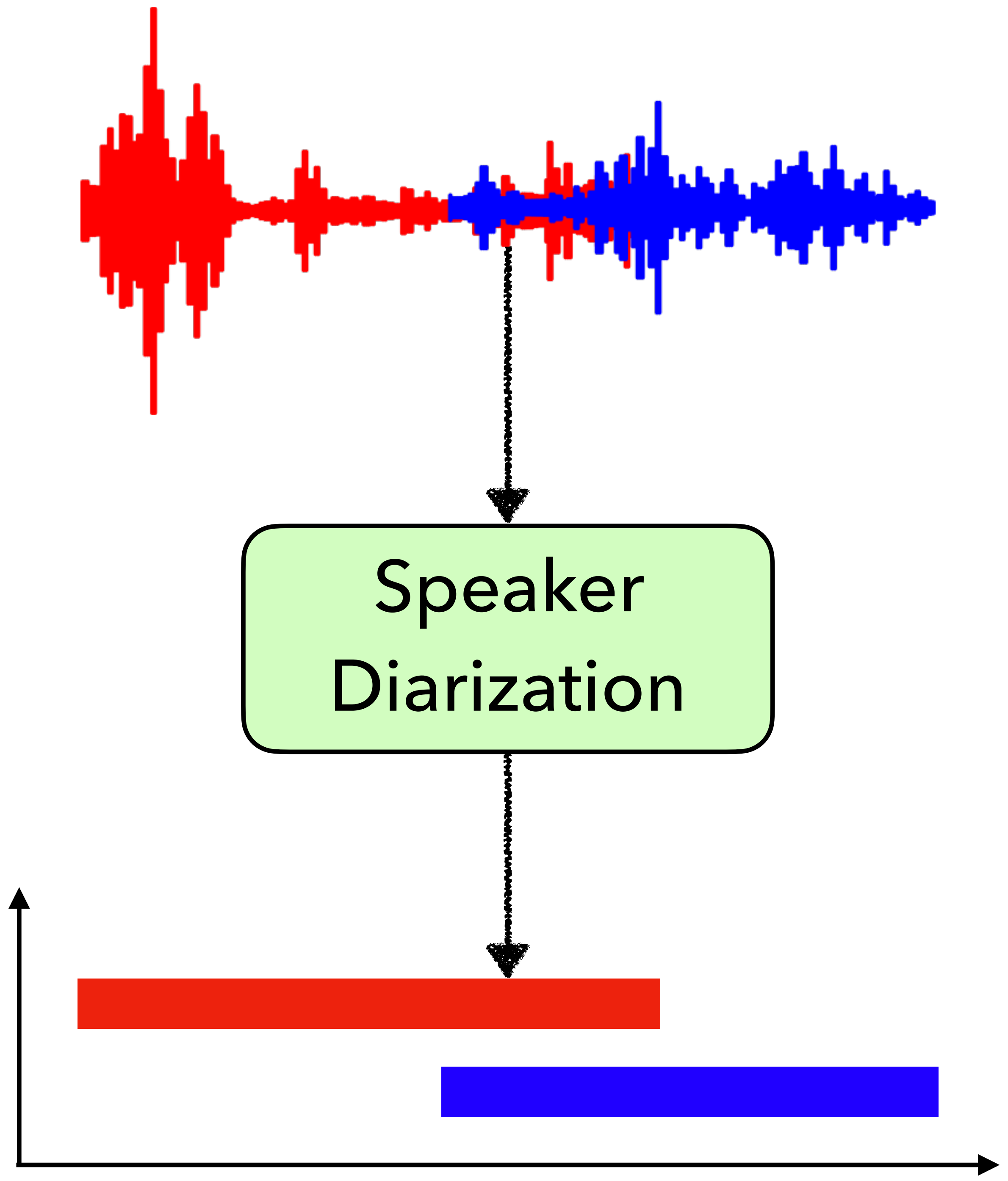
Raj et al. *DOVER-Lap: A method for combining overlap-aware diarization outputs*. **IEEE SLT 2021**.

Horiguchi et al. *The Hitachi-JHU DIHARD III system*. **The Third DIHARD Challenge**.

He et al. *Target-speaker voice activity detection with improved i-vector estimation for unknown number of speaker*. **Interspeech 2021**.

Raj and Khudanpur. *Reformulating DOVER-Lap label mapping as a graph partitioning problem*. **Interspeech 2021**.

Morrone et al. *Low-Latency speech separation guided diarization for telephone conversations*. **IEEE SLT 2022**.



Tasks within the Cocktail Party

Speaker Diarization

Raj et al. *Probing the infomation encoded in x-vectors.* **IEEE ASRU 2019**.

Raj et al. *Multi-class spectral clustering with overlaps for speaker diarization.* **IEEE SLT 2021**.

Raj et al. *DOVER-Lap: A method for combining overlap-aware diarization outputs.* **IEEE SLT 2021**.

Horiguchi et al. *The Hitachi-JHU DIHARD III system.* **The Third DIHARD Challenge**.

He et al. *Target-speaker voice activity detection with improved i-vector estimation for unknown number of speaker.* **Interspeech 2021**.

Raj and Khudanpur. *Reformulating DOVER-Lap label mapping as a graph partitioning problem.* **Interspeech 2021**.

Morrone et al. *Low-Latency speech separation guided diarization for telephone conversations.* **IEEE SLT 2022**.

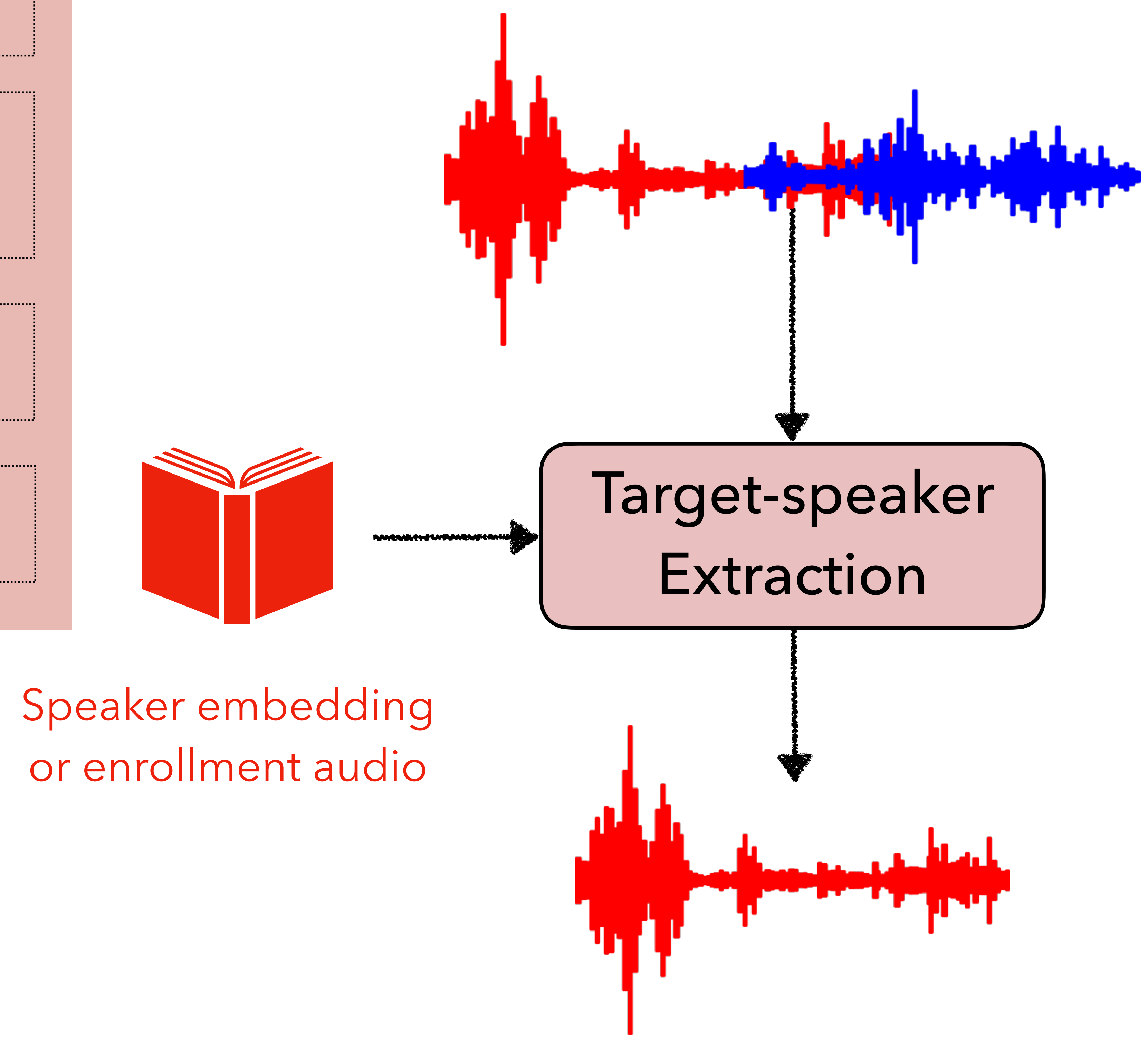
Target-speaker Extraction/Recognition

Zmolikova et al. *Auxiliary loss function for target speech extraction and recognition with weak supervision based on speaker characteristics.* **Interspeech 2021**.

Huang et al. *Adapting self-supervised models to multi-talker speech recognition using speaker embeddings.* **IEEE ICASSP 2023**.

Raj et al. *Anchored speech recognition using neural transducers.* **IEEE ICASSP 2023**.

Raj et al. *GPU-accelerated guided source separation for meeting transcription.* **Interspeech 2023**.



Tasks within the Cocktail Party

Speaker Diarization

Raj et al. *Probing the infomation encoded in x-vectors*. **IEEE ASRU 2019**.

Raj et al. *Multi-class spectral clustering with overlaps for speaker diarization*. **IEEE SLT 2021**.

Raj et al. *DOVER-Lap: A method for combining overlap-aware diarization outputs*. **IEEE SLT 2021**.

Horiguchi et al. *The Hitachi-JHU DIHARD III system*. **The Third DIHARD Challenge**.

He et al. *Target-speaker voice activity detection with improved i-vector estimation for unknown number of speaker*. **Interspeech 2021**.

Raj and Khudanpur. *Reformulating DOVER-Lap label mapping as a graph partitioning problem*. **Interspeech 2021**.

Morrone et al. *Low-Latency speech separation guided diarization for telephone conversations*. **IEEE SLT 2022**.

Target-speaker Extraction/Recognition

Zmolikova et al. *Auxiliary loss function for target speech extraction and recognition with weak supervision based on speaker characteristics*. **Interspeech 2021**.

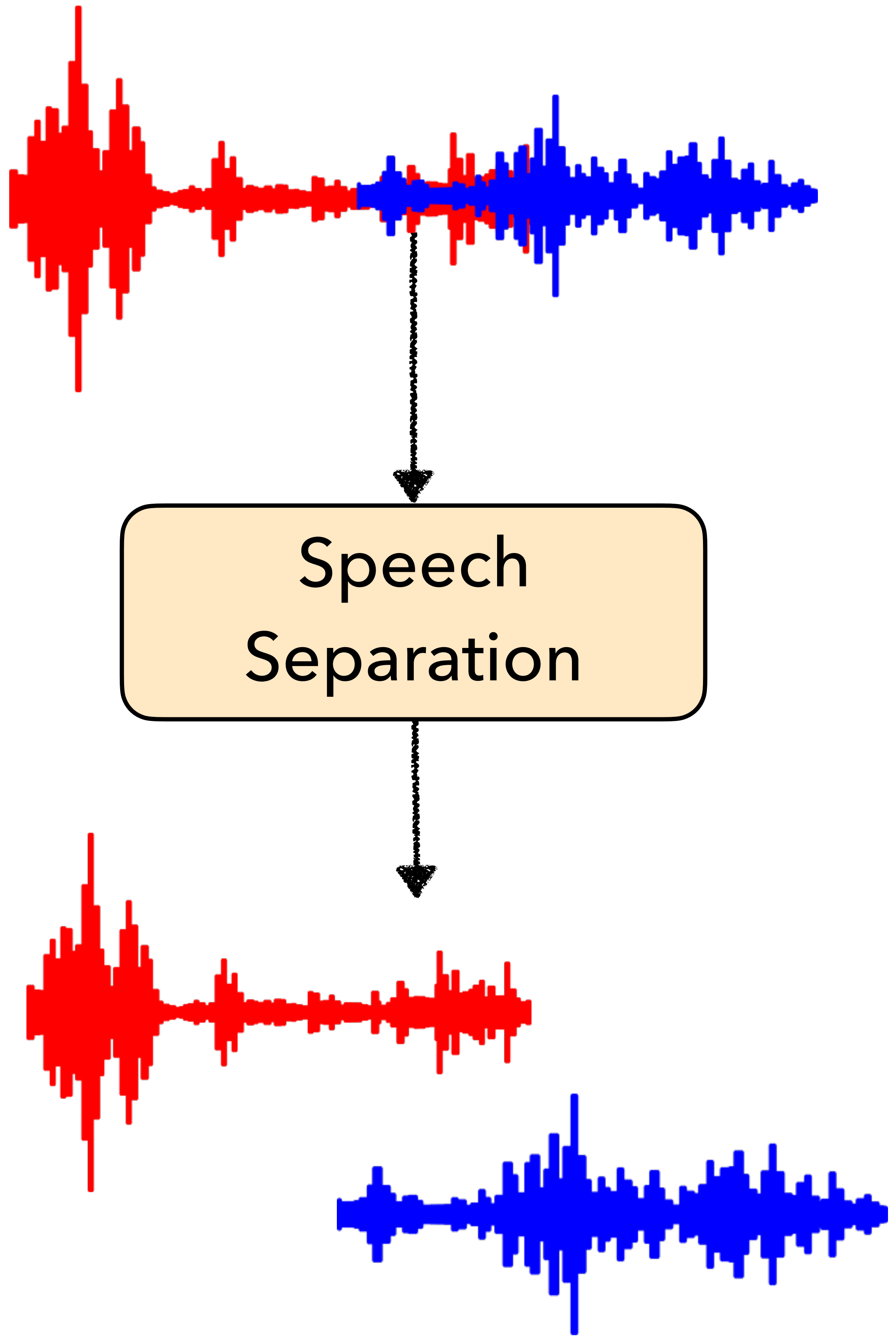
Huang et al. *Adapting self-supervised models to multi-talker speech recognition using speaker embeddings*. **IEEE ICASSP 2023**.

Raj et al. *Anchored speech recognition using neural transducers*. **IEEE ICASSP 2023**.

Raj et al. *GPU-accelerated guided source separation for meeting transcription*. **Interspeech 2023**.

Speech Separation

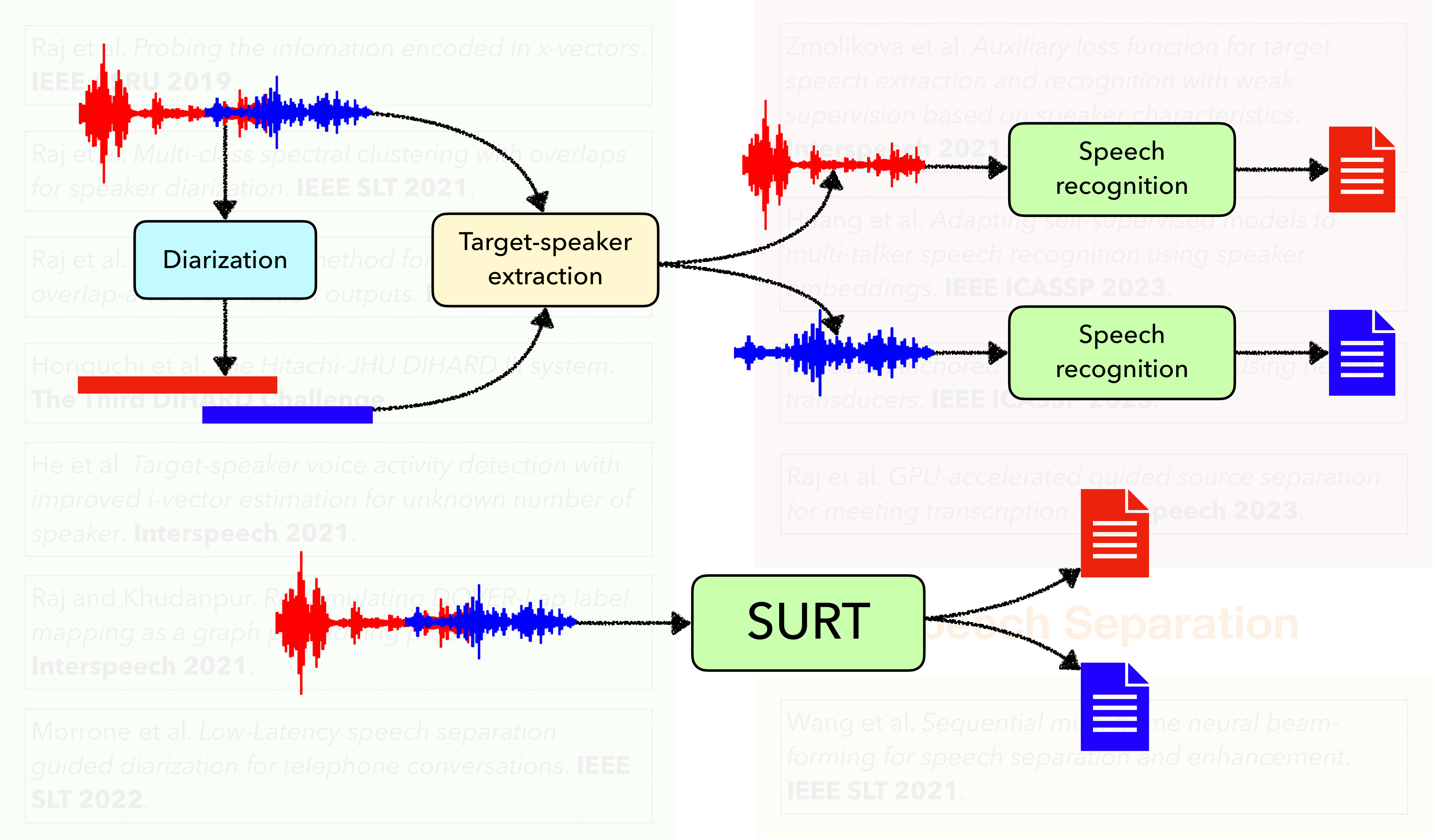
Wang et al. *Sequential multi-frame neural beam-forming for speech separation and enhancement*. **IEEE SLT 2021**.



Tasks within the Cocktail Party

Speaker Diarization

Target-speaker Extraction/Recognition



Multi-talker ASR

Arora et al. *The JHU multi-microphone multi-speaker ASR system for the CHiME-6 challenge*. **CHiME Workshop at IEEE ICASSP 2020**.

Raj et al. *Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis*. **IEEE SLT 2021**.

Huang et al. *Joint speaker diarization and speech recognition based on region proposal networks*. **Computer, Speech, and Language.**

Raj et al. *Continuous streaming multi-talker ASR with dual-path transducers*. **IEEE ICASSP 2022**.

Cornell et al. *The CHiME-7 DASR challenge: Distant meeting transcription with multiple devices in diverse scenarios*. **CHiME Workshop at Interspeech 2023**.

Raj et al. *SURT 2.0: Advances in transducer-based multi-talker speech recognition*. **IEEE Trans. Audio, Speech, and Lang. Proc.**

Raj et al. *Speaker attribution in the SURT framework*. **Speaker Odyssey 2024 (submitted).**

In this talk...

Speaker Diarization

Raj et al. *Probing the infomation encoded in x-vectors.* **IEEE ASRU 2019**.

Raj et al. *Multi-class spectral clustering with overlaps for speaker diarization.* **IEEE SLT 2021**.

Raj et al. *DOVER-Lap: A method for combining overlap-aware diarization outputs.* **IEEE SLT 2021**.

Horiguchi et al. *The Hitachi-JHU DIHARD III system.* **The Third DIHARD Challenge**.

He et al. *Target-speaker voice activity detection with improved i-vector estimation for unknown number of speaker.* **Interspeech 2021**.

Raj and Khudanpur. *Reformulating DOVER-Lap label mapping as a graph partitioning problem.* **Interspeech 2021**.

Morrone et al. *Low-Latency speech separation guided diarization for telephone conversations.* **IEEE SLT 2022**.

Target-speaker Extraction/Recognition

Zmolikova et al. *Auxiliary loss function for target speech extraction and recognition with weak supervision based on speaker characteristics.* **Interspeech 2021**.

Huang et al. *Adapting self-supervised models to multi-talker speech recognition using speaker embeddings.* **IEEE ICASSP 2023**.

Raj et al. *Anchored speech recognition using neural transducers.* **IEEE ICASSP 2023**.

Raj et al. *GPU-accelerated guided source separation for meeting transcription.* **Interspeech 2023**.

Speech Separation

Wang et al. *Sequential multi-frame neural beam-forming for speech separation and enhancement.* **IEEE SLT 2021**.

Multi-talker ASR

Arora et al. *The JHU multi-microphone multi-speaker ASR system for the CHiME-6 challenge.* **CHiME Workshop at IEEE ICASSP 2020**.

Raj et al. *Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis.* **IEEE SLT 2021**.

Huang et al. *Joint speaker diarization and speech recognition based on region proposal networks.* **Computer, Speech, and Language.**

Raj et al. *Continuous streaming multi-talker ASR with dual-path transducers.* **IEEE ICASSP 2022**.

Cornell et al. *The CHiME-7 DASR challenge: Distant meeting transcription with multiple devices in diverse scenarios.* **CHiME Workshop at Interspeech 2023**.

Raj et al. *SURT 2.0: Advances in transducer-based multi-talker speech recognition.* **IEEE Trans. Audio, Speech, and Lang. Proc.**

Raj et al. *Speaker attribution in the SURT framework.* **Speaker Odyssey 2024 (submitted).**

Outline of the talk

“Modular” and “end-to-end” perspectives

1. Problem statement
2. **Modular system**
    (i) Probabilistic formulation
    (ii) Meeting transcription pipeline
3. **End-to-end system**
    (i) Streaming Unmixing and Recognition Transducer (SURT)
    (ii) Speaker-attributed transcription with SURT
4. Conclusion

“Who spoke what?”

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, ICSI

Problem Statement

Corpora

Corpus Name	LibriCSS	AMI	ICSI
Session length	10 minutes	30-45 minutes	~60 minutes
Total size of corpus	10 hours	100 hours	70 hours
Microphones available	7-channel circular array	2 linear arrays with 8 channels each + headset	6 far-field + headset mics
Number of speakers	8	4	3-10
Overlap ratio	0 to 40%	~20%	~14%
Language	English	English	English
	Simulated (replayed)	Real meetings	Real meetings

Problem Statement

Evaluation metrics

- *Speech Recognition*
    - Word error rate (**WER**) = insertion + deletion + substitution (Levenshtein distance)
- *Speaker Diarization*
    - Diarization error rate (**DER**) = missed speech + false alarm + speaker confusion
- *Multi-talker ASR*
    - **ORC-WER**: WER for overlapping speech **without** speaker attribution
    - **cpWER**: WER for overlapping speech **with** speaker attribution

Part I: Modular System

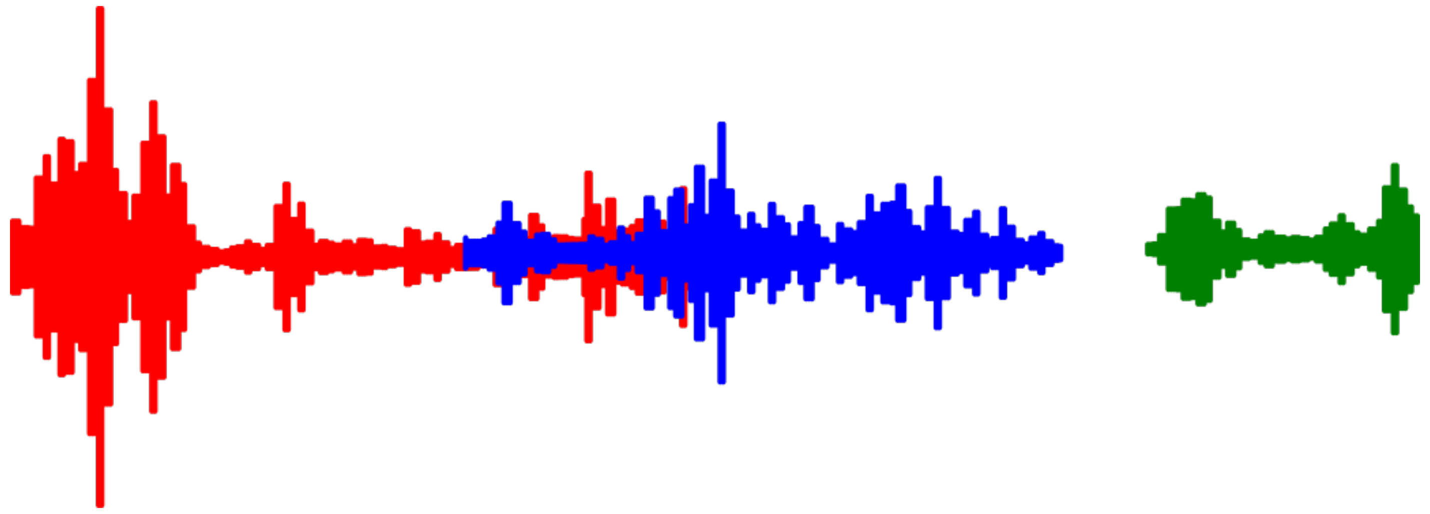
Probabilistic formulation

Input and Output

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

Output: ***speaker-attributed transcripts***

R



Good morning.

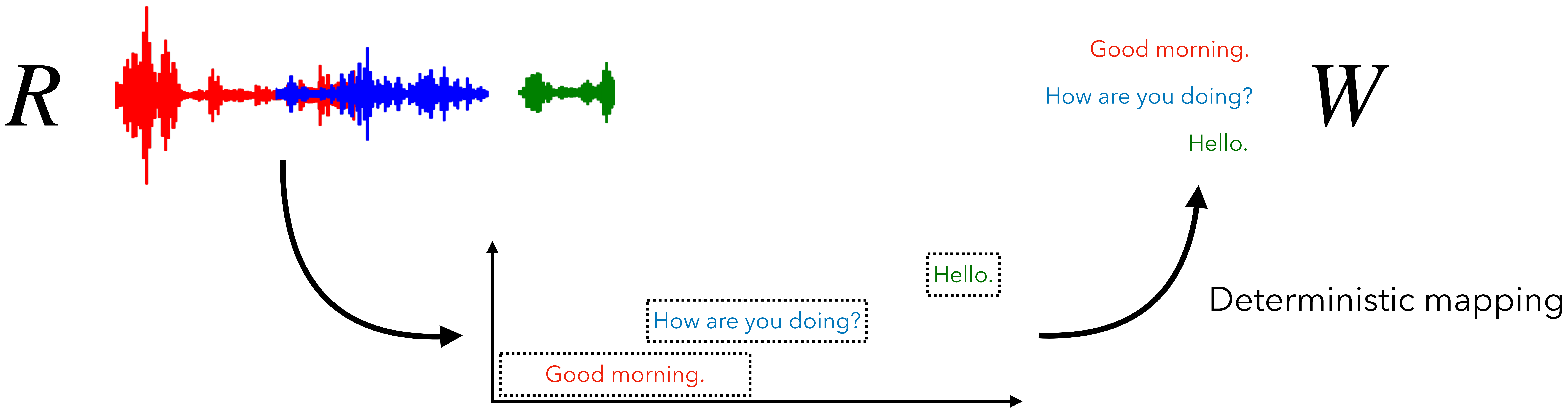
How are you doing?

Hello.

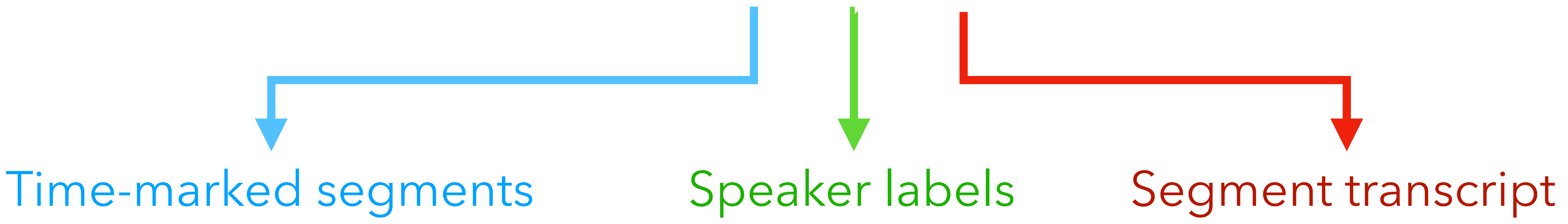
W

Probabilistic formulation

Instead, we model an intermediate solution



$$Y = (\Delta_1^N, u_1^N, \mathbf{y}_1^N)$$



Probabilistic formulation

Maximum *a posteriori*

$$\hat{Y} = \arg\max P(Y \mid R)$$

$$P(Y \mid R) = P\left(\Delta_1^N, u_1^N, \mathbf{y}_1^N \mid R\right)$$

$$= P\left(\Delta_1^N, u_1^N \mid R\right) \ P\left(\mathbf{y}_1^N \mid R, \Delta_1^N, u_1^N\right)$$

Probabilistic formulation

Marginalizing over “target-speaker signals”

$$P\left(\mathbf{y}_1^N \mid R, \Delta_1^N, u_1^N\right) = \int_{\mathbf{X}_1^N} P\left(\mathbf{X}_1^N, \mathbf{y}_1^N \mid R, \Delta_1^N, u_1^N\right)$$

$\mathbf{X}_n$ Target-speaker signal for segment n

Probabilistic formulation

Marginalizing over “target-speaker signals”

$$
\begin{aligned}
P\left(\mathbf{y}_1^N \mid R, \Delta_1^N, u_1^N\right) &= \int_{\mathbf{X}_1^N} P\left(\mathbf{X}_1^N, \mathbf{y}_1^N \mid R, \Delta_1^N, u_1^N\right) \\
&= \int_{\mathbf{X}_1^N} P\left(\mathbf{X}_1^N \mid R, \Delta_1^N, u_1^N\right) P\left(\mathbf{y}_1^N \mid R, \Delta_1^N, u_1^N, \mathbf{X}_1^N\right)
\end{aligned}
$$

Probabilistic formulation

Conditional independence assumptions

$$P\left(\mathbf{y}_1^N \mid R, \Delta_1^N, u_1^N\right) = \int_{\mathbf{X}_1^N} P\left(\mathbf{X}_1^N, \mathbf{y}_1^N \mid R, \Delta_1^N, u_1^N\right)$$

$$= \int_{\mathbf{X}_1^N} P\left(\mathbf{X}_1^N \mid R, \Delta_1^N, u_1^N\right) P\left(\mathbf{y}_1^N \mid R, \Delta_1^N, u_1^N, \mathbf{X}_1^N\right)$$

$$= \int_{\mathbf{X}_1^N} \prod_{j=1}^{N} P\left(\mathbf{X}_j \mid R, \Delta_j, u_j\right) \prod_{j=1}^{N} P\left(\mathbf{y}_j \mid \mathbf{X}_j\right)$$

Target-speaker signal for a segment is independent of the signals for other segments.

Transcript for a segment only depends on the target-speaker signal for that segment.

Probabilistic formulation

Putting it all together

$$\hat{Y} = \arg \max_{\Delta_1^N, u_1^N, \mathbf{y}_1^N} \left[P\left(\Delta_1^N, u_1^N \mid R\right) \int_{\mathbf{X}_1^N} \prod_{j=1}^{N} P\left(\mathbf{X}_j \mid R, \Delta_j, u_j\right) \prod_{j=1}^{N} P\left(\mathbf{y}_j \mid \mathbf{X}_j\right) \right]$$

Computationally intractable!

Probabilistic formulation

The "modular" solution

$$\hat{Y} = \arg\max_{\Delta_1^N, u_1^N, \mathbf{y}_1^N} \left[P\left(\Delta_1^N, u_1^N \mid R\right) \int_{\mathbf{X}_1^N} \prod_{j=1}^{N} P\left(\mathbf{X}_j \mid R, \Delta_j, u_j\right) \prod_{j=1}^{N} P\left(\mathbf{y}_j \mid \mathbf{X}_j\right) \right]$$

STEP 1:

$$\underbrace{\hat{\Delta}_1^N, \hat{u}_1^N = \arg\max_{\Delta_1^N, u_1^N} P\left(\Delta_1^N, u_1^N \mid R\right)}_{\text{speaker diarization}}$$

Probabilistic formulation

The “modular” solution

$$\hat{Y} = \arg \max_{\Delta_1^N, u_1^N, \mathbf{y}_1^N} \left[P\left(\Delta_1^N, u_1^N \mid R\right) \int_{\mathbf{X}_1^N} \prod_{j=1}^{N} P\left(\mathbf{X}_j \mid R, \Delta_j, u_j\right) \prod_{j=1}^{N} P\left(\mathbf{y}_j \mid \mathbf{X}_j\right) \right]$$

STEP 1:

$$\underbrace{\hat{\Delta}_1^N, \hat{u}_1^N = \arg \max_{\Delta_1^N, u_1^N} P\left(\Delta_1^N, u_1^N \mid R\right)}_{\text{speaker diarization}}$$

STEP 2:

$$\underbrace{\hat{\mathbf{X}}_j = g(R, \hat{\Delta}_j, \hat{u}_j)}_{\text{target speaker extraction}}$$

Probabilistic formulation

The “modular” solution

$$\hat{Y} = \arg\max_{\Delta_1^N, u_1^N, \mathbf{y}_1^N} \left[P\left(\Delta_1^N, u_1^N \mid R\right) \int_{\mathbf{X}_1^N} \prod_{j=1}^{N} P\left(\mathbf{X}_j \mid R, \Delta_j, u_j\right) \prod_{j=1}^{N} P\left(\mathbf{y}_j \mid \mathbf{X}_j\right) \right]$$

STEP 1:

$$\underbrace{\hat{\Delta}_1^N, \hat{u}_1^N = \arg\max_{\Delta_1^N, u_1^N} P\left(\Delta_1^N, u_1^N \mid R\right)}_{\text{speaker diarization}}$$

STEP 2:

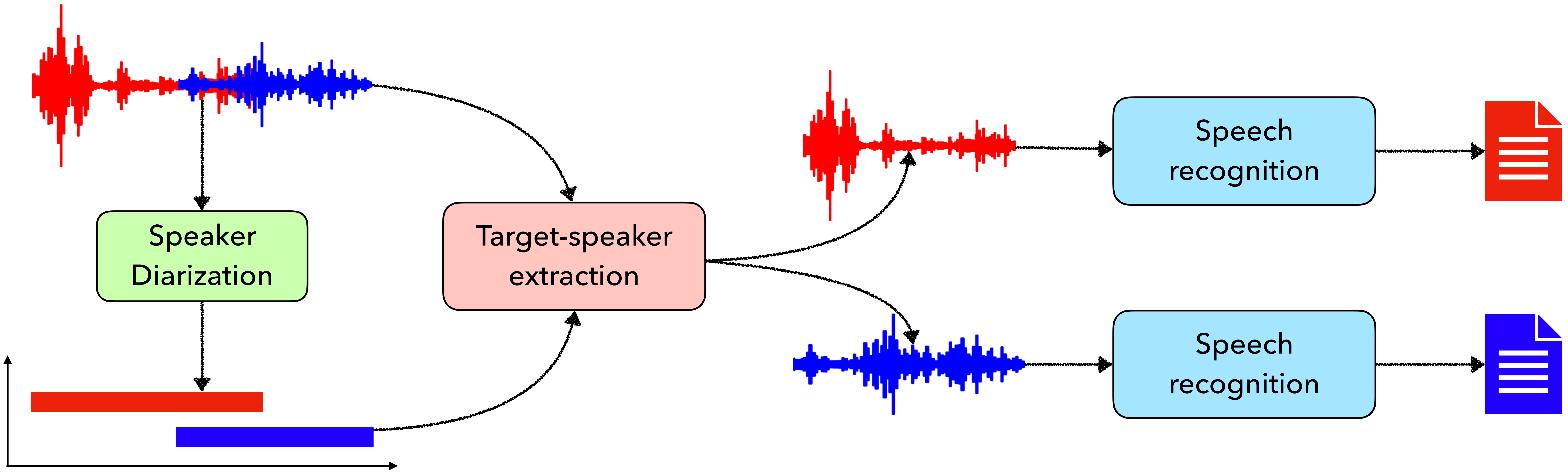
$$\underbrace{\hat{\mathbf{X}}_j = g(R, \hat{\Delta}_j, \hat{u}_j)}_{\text{target speaker extraction}}$$

STEP 3:

$$P\left(\mathbf{y}_1^N \mid R, \Delta_1^N, u_1^N\right) \approx \prod_{j=1}^{N} \underbrace{P\left(\mathbf{y}_j \mid \hat{\mathbf{X}}_j\right)}_{\text{speech recognition}}$$

Meeting transcription pipeline

Based on the modular approach



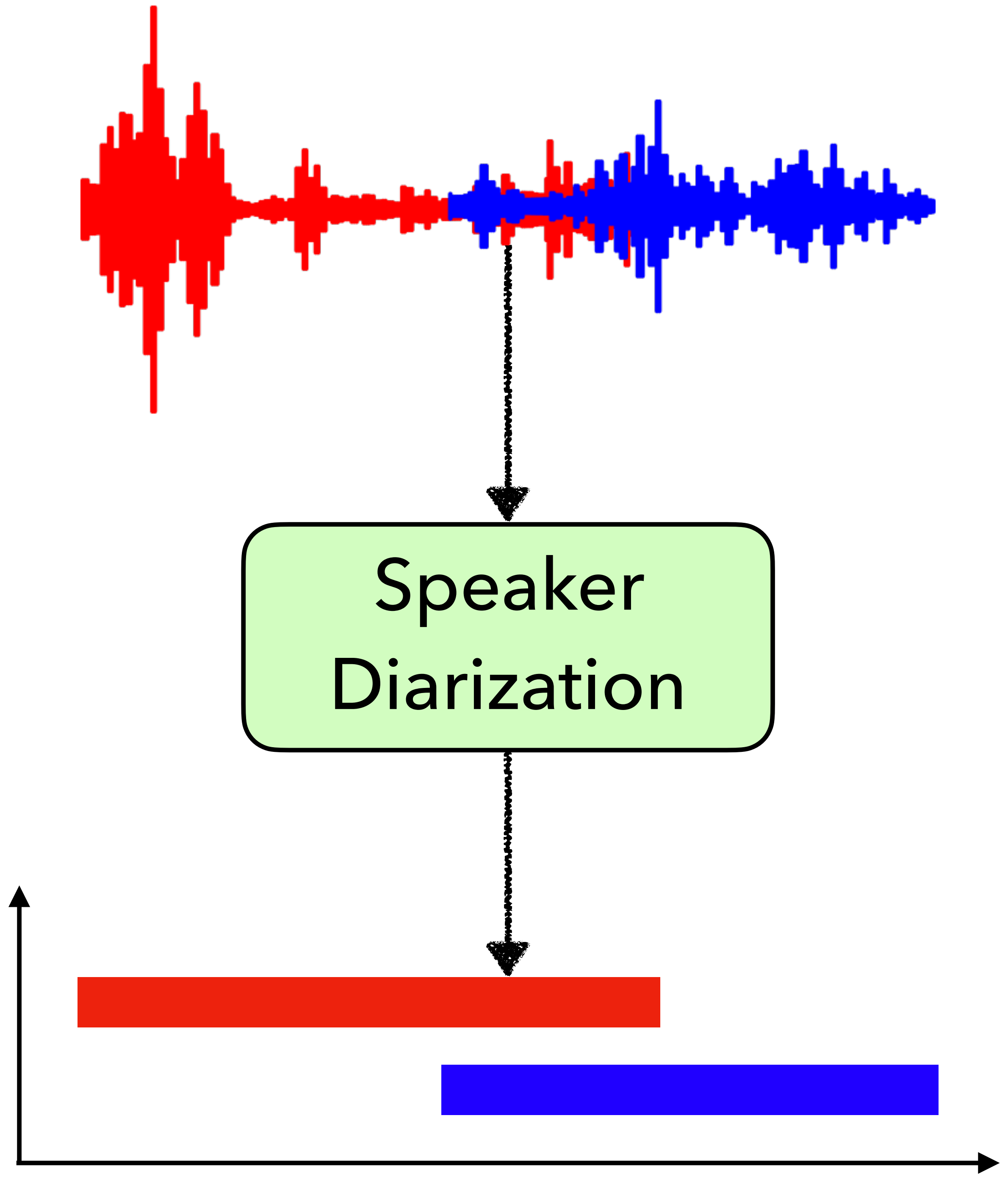
Diarization should correctly identify all speakers (including overlaps).

TSE module should be efficient for extracting signals for all segments.

Meeting transcription pipeline

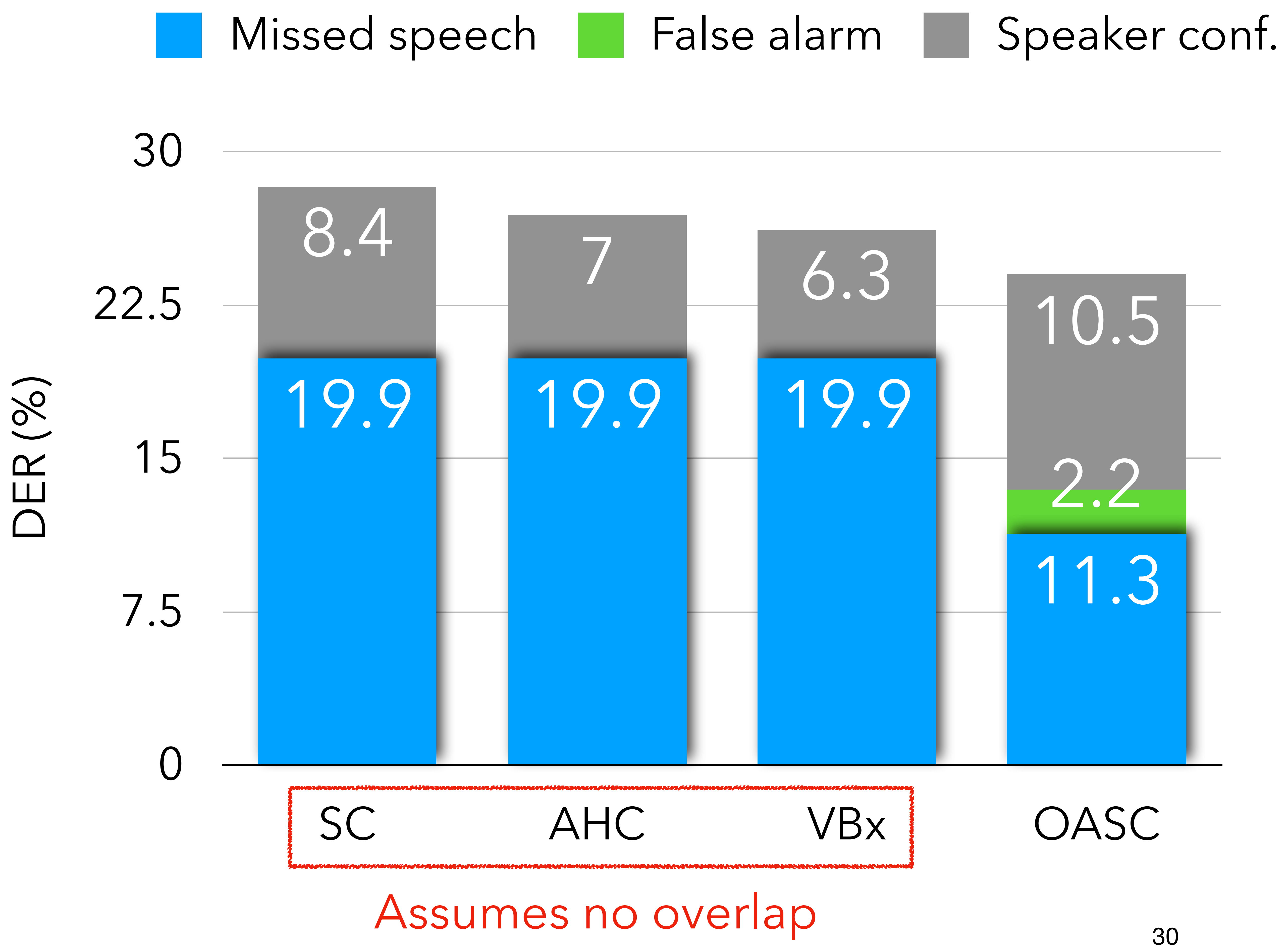
Contribution #1: Overlap-aware spectral clustering

- Clustering-based diarization usually assumed single-speaker segments, which leads to high *missed speech*.
- We propose a new *overlap-aware* diarization method, based on a graphical formulation of spectral clustering.
- This new method can incorporate an *external overlap detector*.



Meeting transcription pipeline

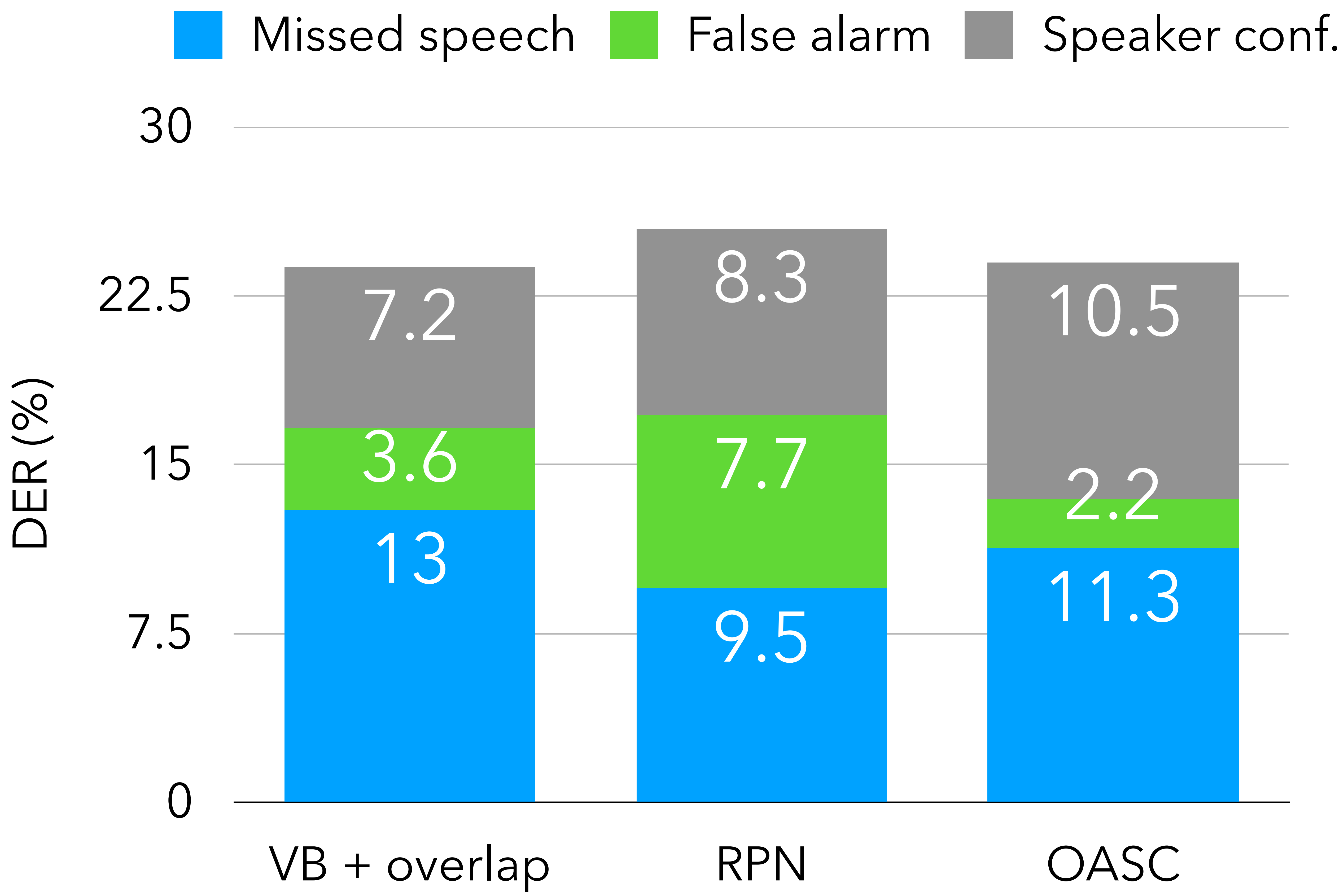
Contribution #1: Overlap-aware spectral clustering



12% relative DER improvement on AMI over spectral clustering baseline.

Meeting transcription pipeline

Contribution #1: Overlap-aware spectral clustering

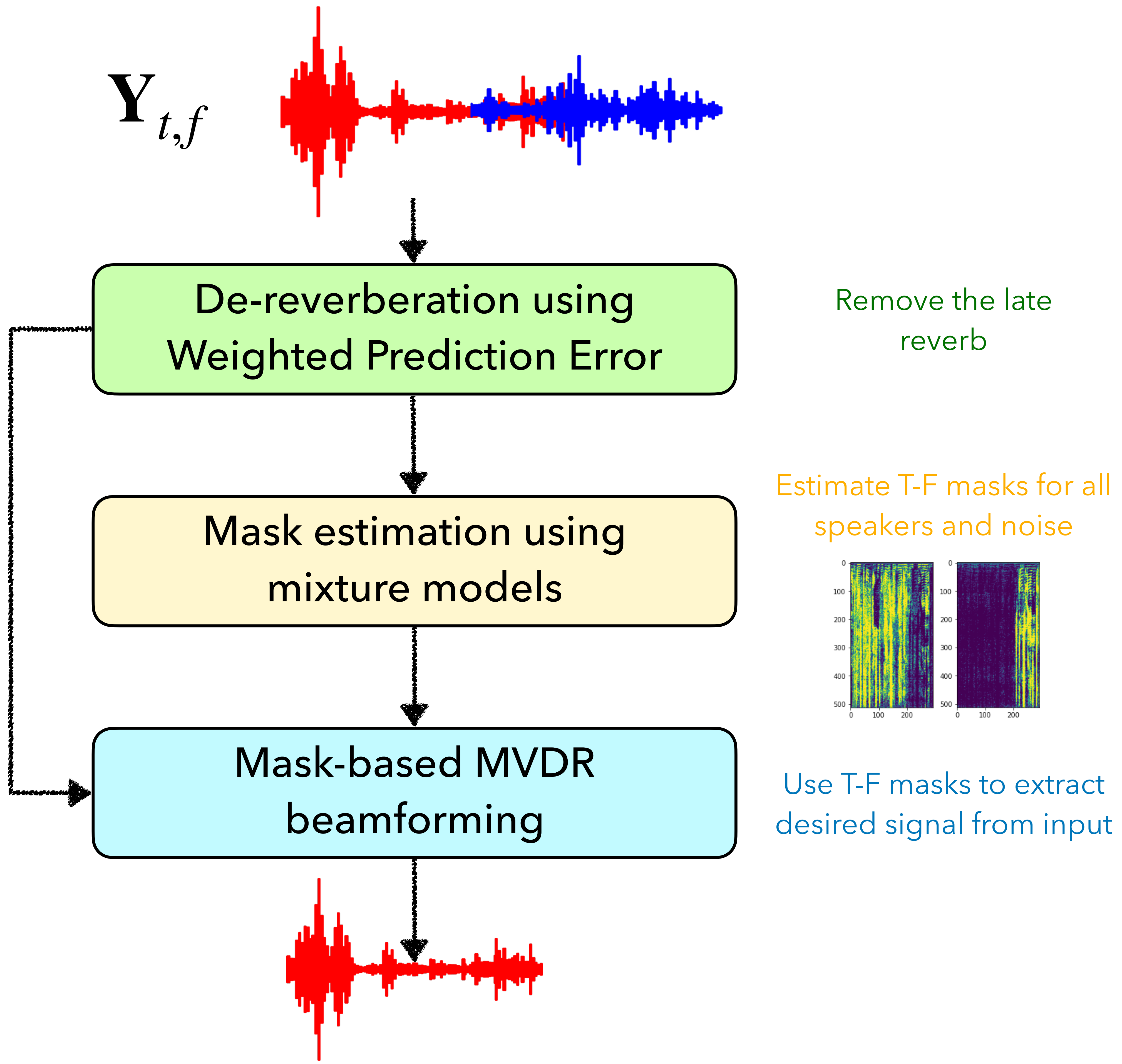


Does not require **matching training data** or **initialization** with other diarization systems.

Meeting transcription pipeline

Contribution #2: GPU-accelerated Guided Source Separation

- GSS is a signal processing method for target-speaker extraction.
- Contains several iterative parts, e.g., mask estimation using complex angular GMMs.
- Implemented **300x faster** GPU-accelerated GSS using smart batching and caching strategies.
- Processing time for CHiME-6 dev set reduced from **19.3h** (using 80 CPUs) to **1.3h** (using 4 GPUs)



Meeting transcription pipeline

Results on LibriCSS

10 minute sessions, 0-40% overlapping speech, mixed LibriSpeech utterances

Diarization	TSE	DER (%)	cpWER (%)
Spectral clustering	None	14.9	18.3

Meeting transcription pipeline

Results on LibriCSS

10 minute sessions, 0-40% overlapping speech, mixed LibriSpeech utterances

Diarization	TSE	DER (%)	cpWER (%)
Spectral clustering	None	14.9	18.3
Overlap-aware SC	None	**11.3** ↓ 24.2%	17.1 ↓ 6.6%

Meeting transcription pipeline

Results on LibriCSS

10 minute sessions, 0-40% overlapping speech, mixed LibriSpeech utterances

<table>
<tr><th>Diarization</th><th>TSE</th><th>DER (%)</th><th>cpWER (%)</th></tr>
<tr><td>Spectral clustering</td><td>None</td><td>14.9</td><td>18.3</td></tr>
<tr><td rowspan="2">Overlap-aware SC</td><td>None</td><td rowspan="2">11.3
24.2%</td><td>17.1</td></tr>
<tr><td>GSS</td><td>12.1
33.9%</td></tr>
</table>

Meeting transcription pipeline
Results on AMI

30 minute sessions, ~20% overlapping speech, real 4-person meetings

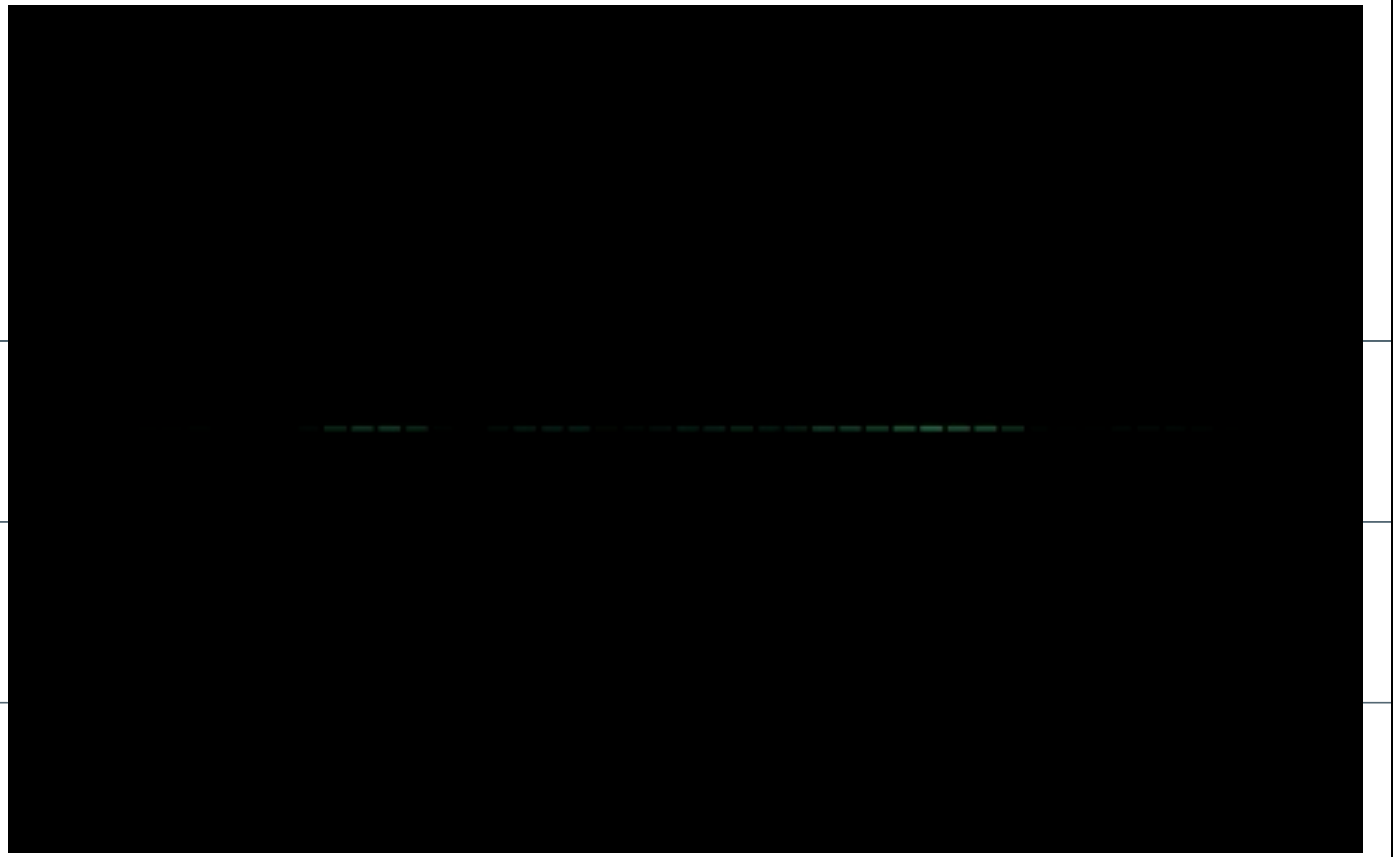
Diarization	TSE	DER (%)	cpWER (%)
Spectral clustering	None	25.5	38.5
Overlap-aware SC	None	23.7	38.5
	GSS		**31.0**

Meeting transcription pipeline

Qualitative analysis

AMI ES2011a (from 817s to 833s)

Speakers

<table>
<tr><th>Reference</th><th></th></tr>
<tr><td>I ALSO THINK THOUGH THAT IT SHOULDN'T HAVE TOO MANY BUTTONS 'CAUSE I HATE THAT WHEN THEY HAVE TOO MANY BUTTONS AND I MEAN I KNOW IT HAS TO HAVE ENOUGH FUNCTIONS BUT LIKE I DON'T KNOW YOU JUST HAVE LIKE EIGHT THOUSAND BUTTONS AND YOU'RE LIKE NO YOU NEVER USE HALF OF THEM | SO</td><td rowspan="4"></td></tr>
<tr><td>YEAH I AGREE | B BUTTON AND THE F BUTTON THEY DON'T DO ANYTHING</td></tr>
<tr><td>UM OH WE JUST | YEAH</td></tr>
<tr><td>YEAH YEAH YEAH</td></tr>
</table>

Meeting transcription pipeline

Qualitative analysis

AMI ES2011a (from 817s to 833s)

cpWER = 40.5%

Speakers

Reference	Spectral clustering + No GSS
I ALSO THINK THOUGH THAT IT SHOULDN'T HAVE TOO MANY BUTTONS 'CAUSE I HATE THAT WHEN THEY HAVE TOO MANY BUTTONS AND I MEAN I KNOW IT HAS TO HAVE ENOUGH FUNCTIONS BUT LIKE I DON'T KNOW YOU JUST HAVE LIKE EIGHT THOUSAND BUTTONS AND YOU'RE LIKE NO YOU NEVER USE HALF OF THEM | SO	I ALSO THINK THOUGH THAT IT SHOULDN'T HAVE TOO MANY BUTTONS 'CAUSE I HATE THAT ONLY HAVE TOO MANY BUTTONS AND I MEAN I KNOW IT HAS TO HAVE MANY FUNCTIONS BUT LIKE | I DUNNO JUST HAVE LIKE EIGHT THOUSAND BUTTONS AND YOU'RE LIKE NO YOU NEVER USE HALF OF THEM
YEAH I AGREE | B BUTTON AND THE F BUTTON THEY DON'T DO ANYTHING	–
UM OH WE JUST | YEAH	–
YEAH YEAH YEAH	–

Meeting transcription pipeline

Qualitative analysis

AMI ES2011a (from 817s to 833s)

cpWER = 72.2%

Speakers	Reference	Overlap-aware Spectral clustering + No GSS
	I ALSO THINK THOUGH THAT IT SHOULDN'T HAVE TOO MANY BUTTONS 'CAUSE I HATE THAT WHEN THEY HAVE TOO MANY BUTTONS AND I MEAN I KNOW IT HAS TO HAVE ENOUGH FUNCTIONS BUT LIKE I DON'T KNOW YOU JUST HAVE LIKE EIGHT THOUSAND BUTTONS AND YOU'RE LIKE NO YOU NEVER USE HALF OF THEM | SO	I ALSO THINK THAT IT SHOULDN'T HAVE TOO MANY BUTTONS 'CAUSE I HATED NOT ONLY HAVE TOO MANY BUTTONS AND THINGS BUT I MEAN I KNOW IT HAS TO HAVE NO MANY FUNCTIONS BUT LIKE | I DUNNO JUST HAVE LIKE EIGHT THOUSAND BUTTONS AND YOU'RE LIKE YOU KNOW YOU NEVER USE HALF THE TIME
	YEAH I AGREE | B BUTTON AND THE F BUTTON THEY DON'T DO ANYTHING	IT SHOULDN'T HAVE TOO MANY BUTTONS 'CAUSE I HATE THAT ONLY HAVE TOO MANY BUTTONS AND THINGS BUT I MEAN I KNOW IT HAS TO HAVE NO MANY FUNCTIONS BUT
	UM OH WE JUST | YEAH	–
	YEAH YEAH YEAH	–

Meeting transcription pipeline
Qualitative analysis

AMI ES2011a (from 817s to 833s)

cpWER = 29.1%

Speakers

Reference	Overlap-aware Spectral clustering + GSS
I ALSO THINK THOUGH THAT IT SHOULDN'T HAVE TOO MANY BUTTONS 'CAUSE I HATE THAT WHEN THEY HAVE TOO MANY BUTTONS AND I MEAN I KNOW IT HAS TO HAVE ENOUGH FUNCTIONS BUT LIKE I DON'T KNOW YOU JUST HAVE LIKE EIGHT THOUSAND BUTTONS AND YOU'RE LIKE NO YOU NEVER USE HALF OF THEM | SO	I ALSO THINK THOUGH THAT IT SHOULDN'T HAVE TOO MANY BUTTONS 'CAUSE I HAD THAT ONLY HAVE TOO MANY BUTTONS AND I MEAN I KNOW IT HAS TO HAVE ENOUGH FUNCTIONS BUT LIKE | I DUNNO JUST HAVE LIKE EIGHT THOUSAND BUTTONS AND YOU'RE LIKE NO YOU NEVER USE HALF OF THEM
YEAH I AGREE | B BUTTON AND THE F BUTTON THEY DON'T DO ANYTHING	S YEAH I AGREE M THE BUTTON ON F BUTTON THEY DON'T DO ANYTHING
UM OH WE JUST | YEAH	–
YEAH YEAH YEAH	–

Modular system

Limitations

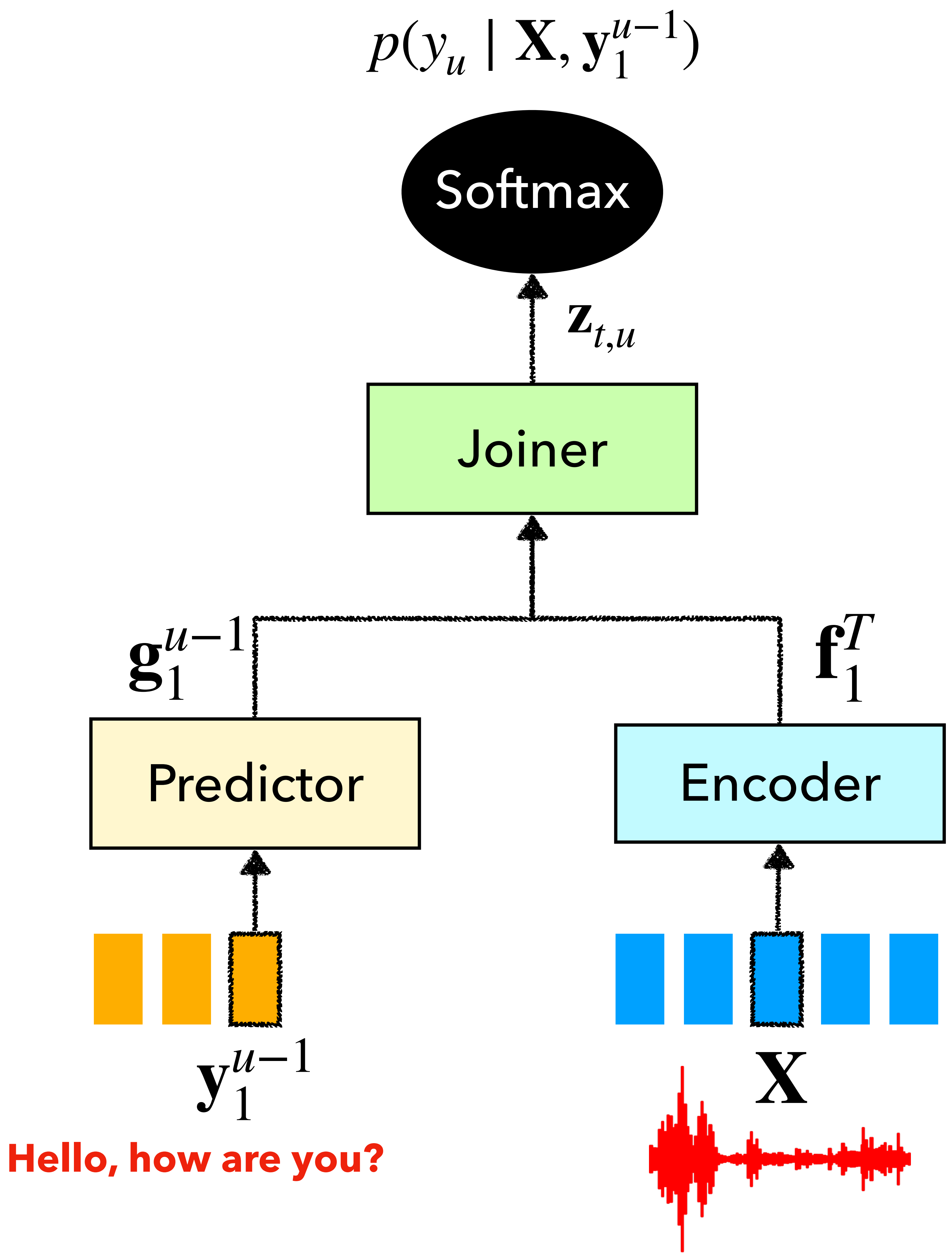
- Modules are independently optimized for different objectives
- Higher accumulated latency
- Error propagation through modules
- Requires more engineering efforts to maintain

- Cannot be used for streaming or single-channel inputs

Part II: End-to-end System

Preliminary

Neural transducers for ASR



- **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

$$P(\mathbf{y} \mid \mathbf{X}) = \sum_{\mathbf{a} \in \mathscr{B}^{-1}(\mathbf{y})} P(\mathbf{a} \mid \mathbf{X}) = \sum_{\mathbf{a} \in \mathscr{B}^{-1}(\mathbf{y})} \prod_{t=1}^{T} P(a_t \mid \mathbf{X}, \mathbf{a}_{1:t-1})$$

Continuous, streaming, multi-talker ASR

Using neural transducers

- **Continuous:** does not rely on external segmentation
- **Streaming:** does not use right context; overlapping speech is transcribed simultaneously

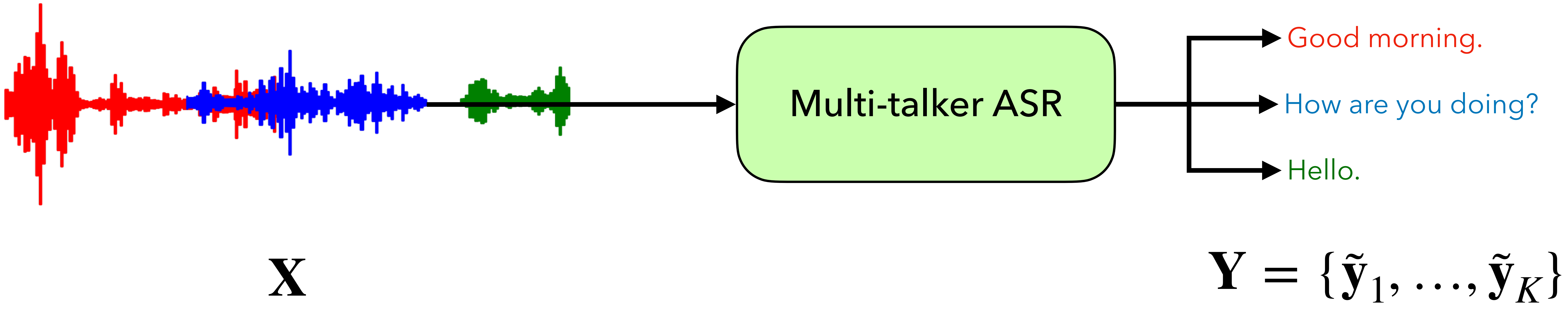
- Assume we have K speakers in the input audio

Continuous, streaming, multi-talker ASR

Option 1: Single output stream per speaker

- Assume each speaker's transcript is conditionally independent of others given the audio

$$P(\mathbf{Y} \mid \mathbf{X}) = P(\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_K \mid \mathbf{X}) \approx \prod_{k=1}^{K} P(\tilde{\mathbf{y}}_k \mid \mathbf{X})$$



Continuous, streaming, multi-talker ASR

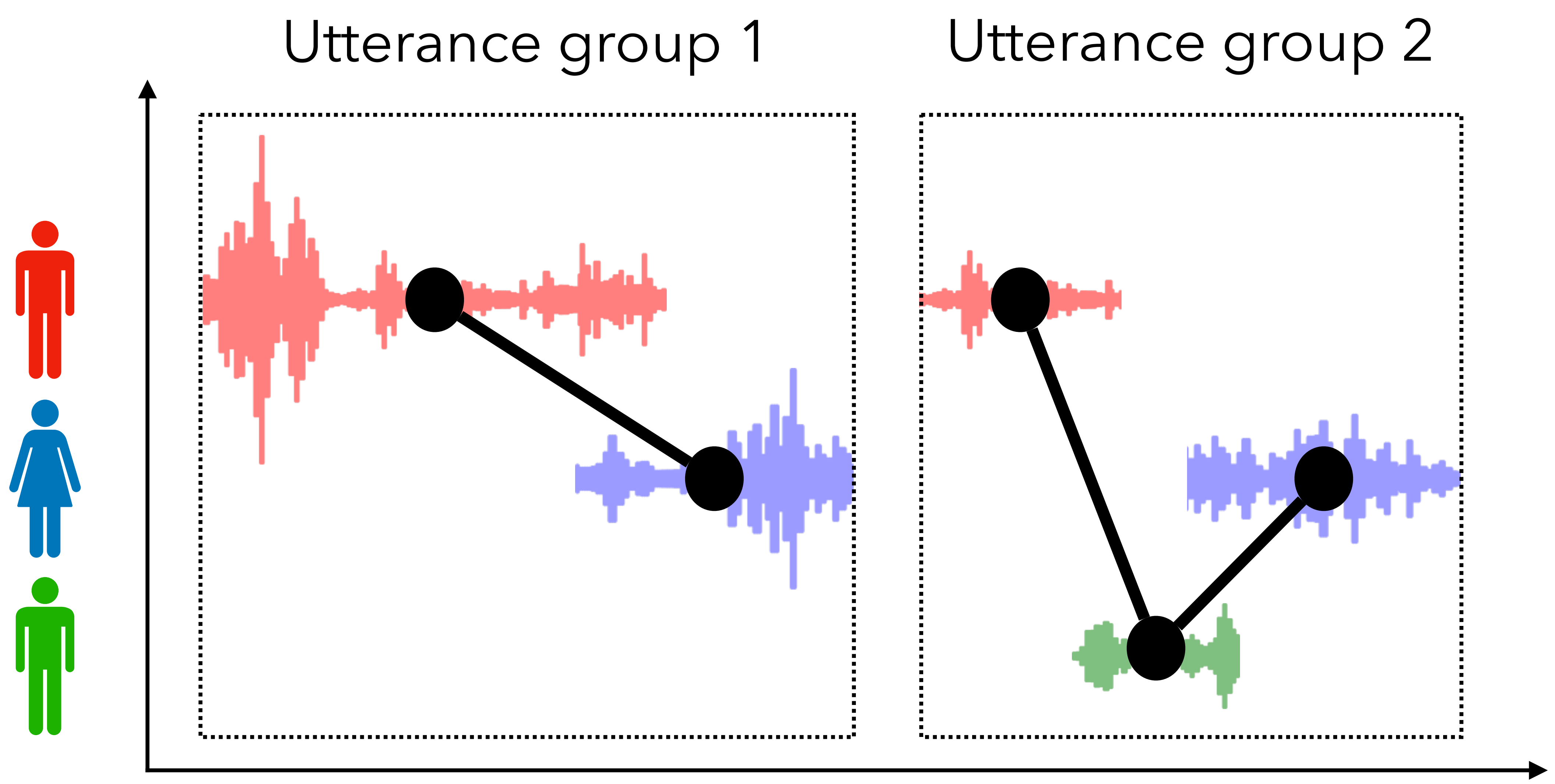
Option 1: Single output stream per speaker

- **Limitations:**
    1. Number of output **channels** is K, i.e., model depends on input
    2. Requires $\mathcal{O}(K^2)$ number of transducer loss computations to solve speaker permutation problem at the output

- Need to find a solution with fixed number of output channels

Continuous, streaming, multi-talker ASR

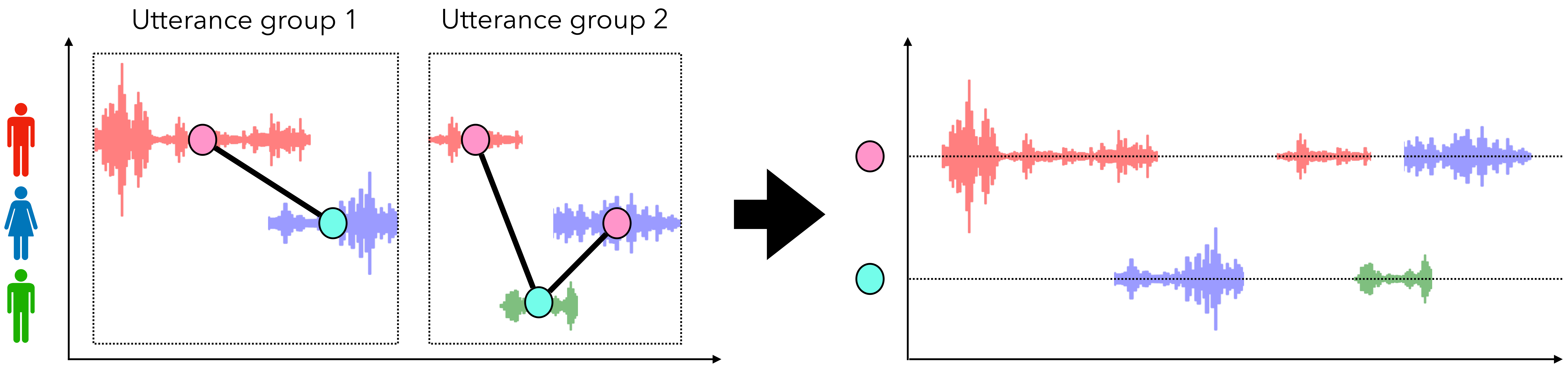
Option 2: Graph coloring approach



- Graph with each utterance as a node
- If two utterances overlap, connect them with an edge

Continuous, streaming, multi-talker ASR

Option 2: Graph coloring approach



- If the graph is colorable with C colors, then the utterances can be mapped to C channels without overlaps.
- Overlaps of 3 or more speakers are extremely rare, so we can assume 2 output channels henceforth.

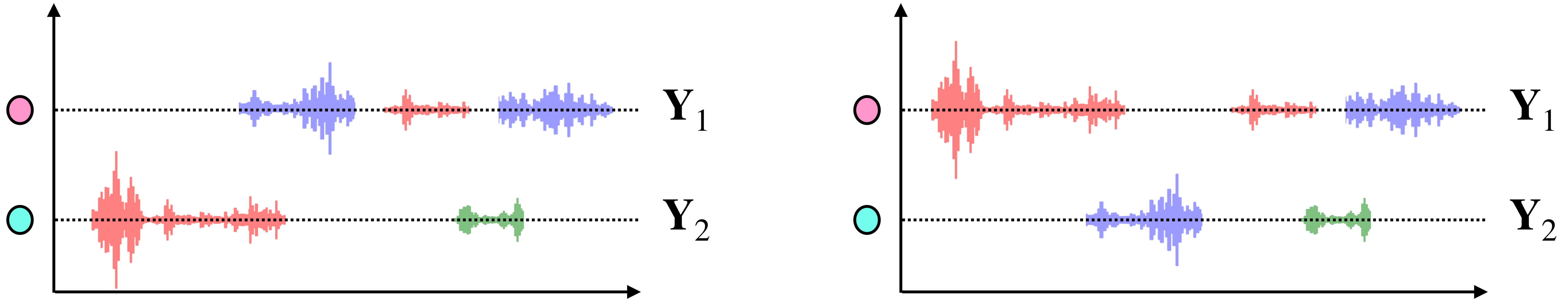
Continuous, streaming, multi-talker ASR

Permutation invariant training (PIT)

$$P(\mathbf{y}_1, \ldots, \mathbf{y}_N \mid \mathbf{X}) = \max_{\zeta} P(\mathbf{Y}_1, \mathbf{Y}_2 \mid \mathbf{X})$$
$$\approx \max_{\zeta} P(\mathbf{Y}_1 \mid \mathbf{X}) P(\mathbf{Y}_2 \mid \mathbf{X}),$$

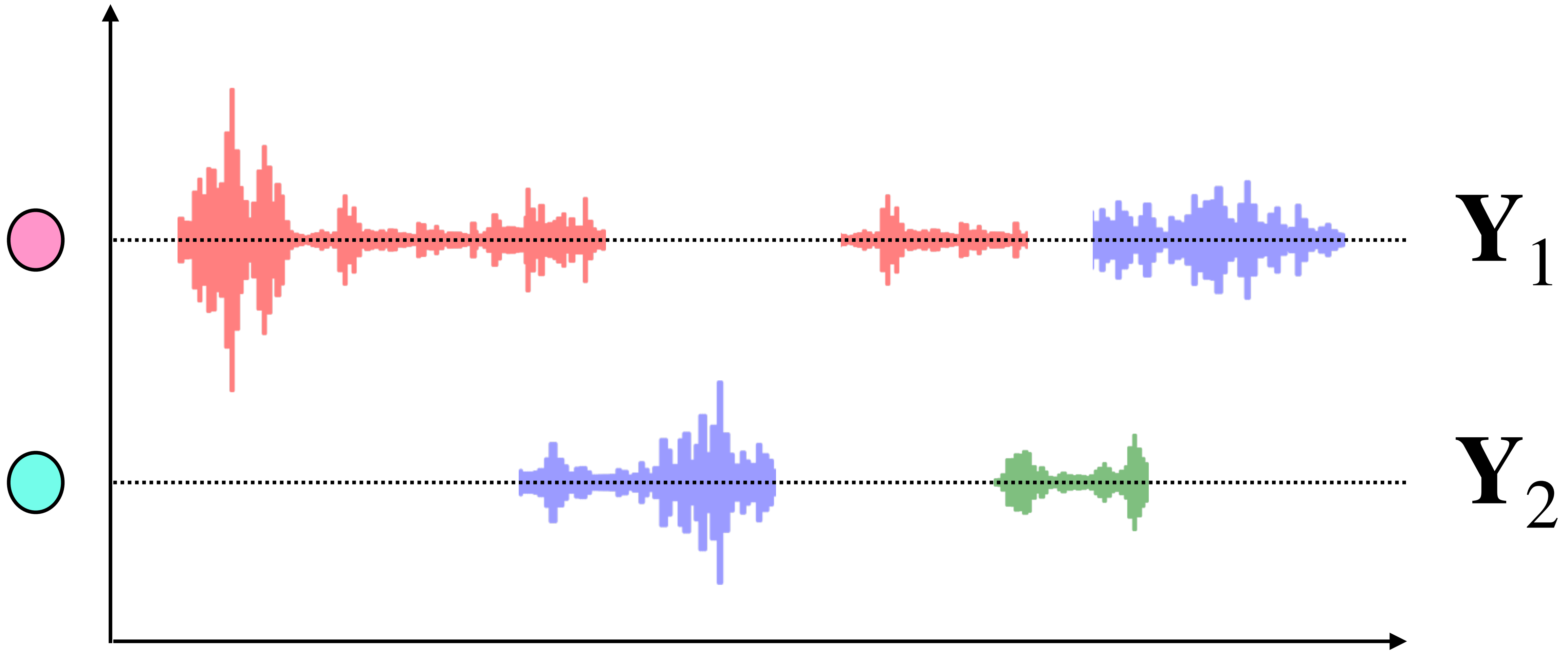
$$\mathscr{L}_{\text{pit}}(\mathbf{y}_{1:N}, \mathbf{X}; \Theta) = \min_{\zeta} \left[-\log P_{\Theta}(\mathbf{Y}_1 \mid \mathbf{X}) - \log P_{\Theta}(\mathbf{Y}_2 \mid \mathbf{X}) \right]$$

- ζ: all possible assignment of $\mathbf{y}_1, \ldots, \mathbf{y}_N$ on to two output channels
- Number of assignments is **exponential** in the number of utterance groups!



Continuous, streaming, multi-talker ASR

Heuristic error assignment training (HEAT)



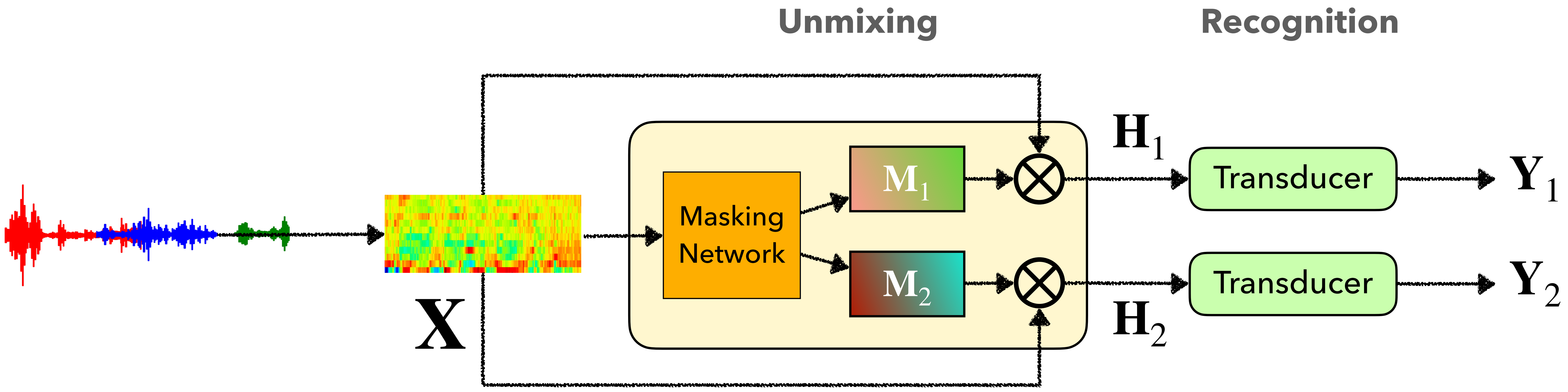
$$P(\mathbf{y}_1, \ldots, \mathbf{y}_N \mid \mathbf{X}) = P(\mathbf{Y}_1 \mid \mathbf{X})P(\mathbf{Y}_2 \mid \mathbf{X})$$

- Assign utterances to first available channel in order of start time

$$\mathscr{L}_{\text{heat}}(\mathbf{y}_{1:N}, \mathbf{X}; \Theta) = -\log P_{\Theta}(\mathbf{Y}_1 \mid \mathbf{X}) - \log P_{\Theta}(\mathbf{Y}_2 \mid \mathbf{X})$$

Streaming Unmixing and Recognition Transducer (SURT)

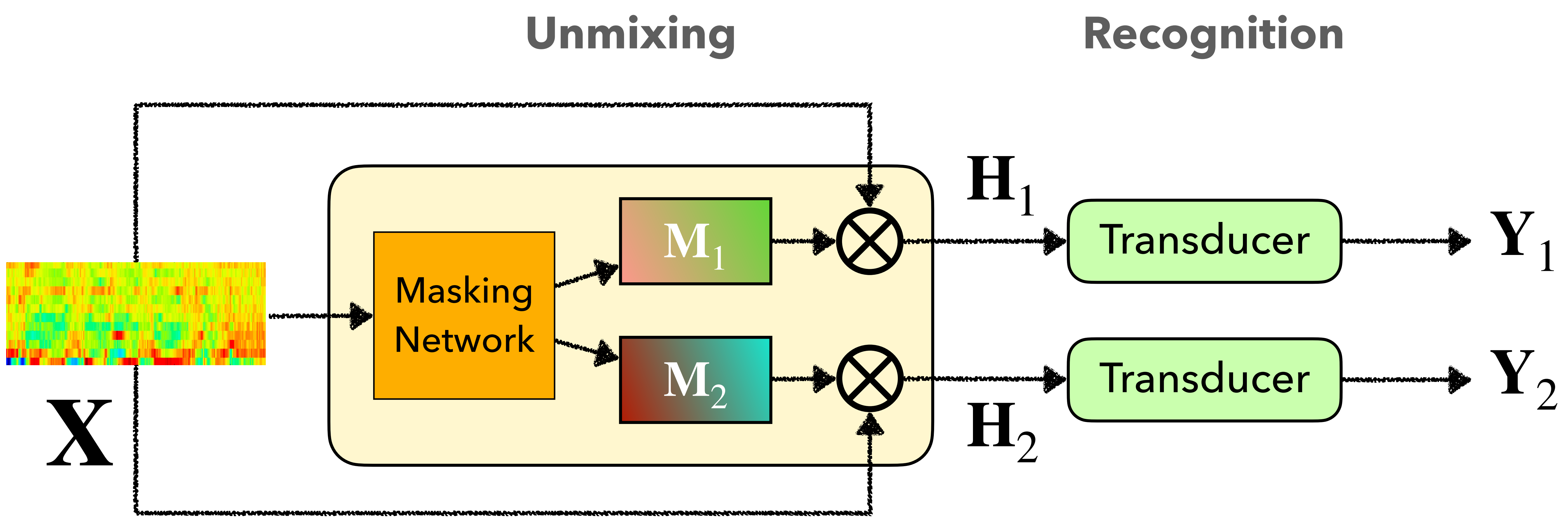
Model



Streaming Unmixing and Recognition Transducer (SURT)

Some challenges

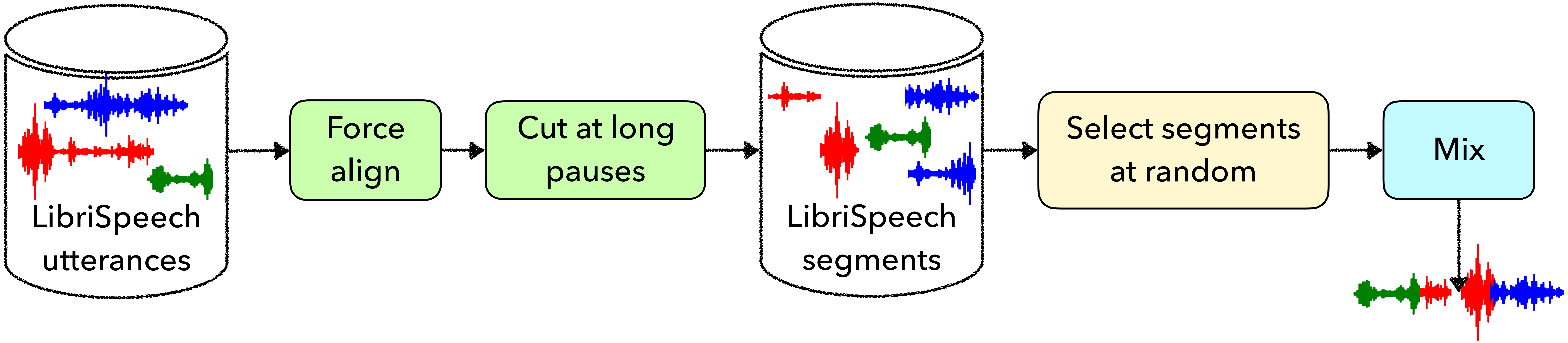
1. How to train the model efficiently?
2. What kind of errors can happen with such models?
3. Can the model work well on real meetings?



Problem 1

Making training efficient

#1: Shorter training mixtures



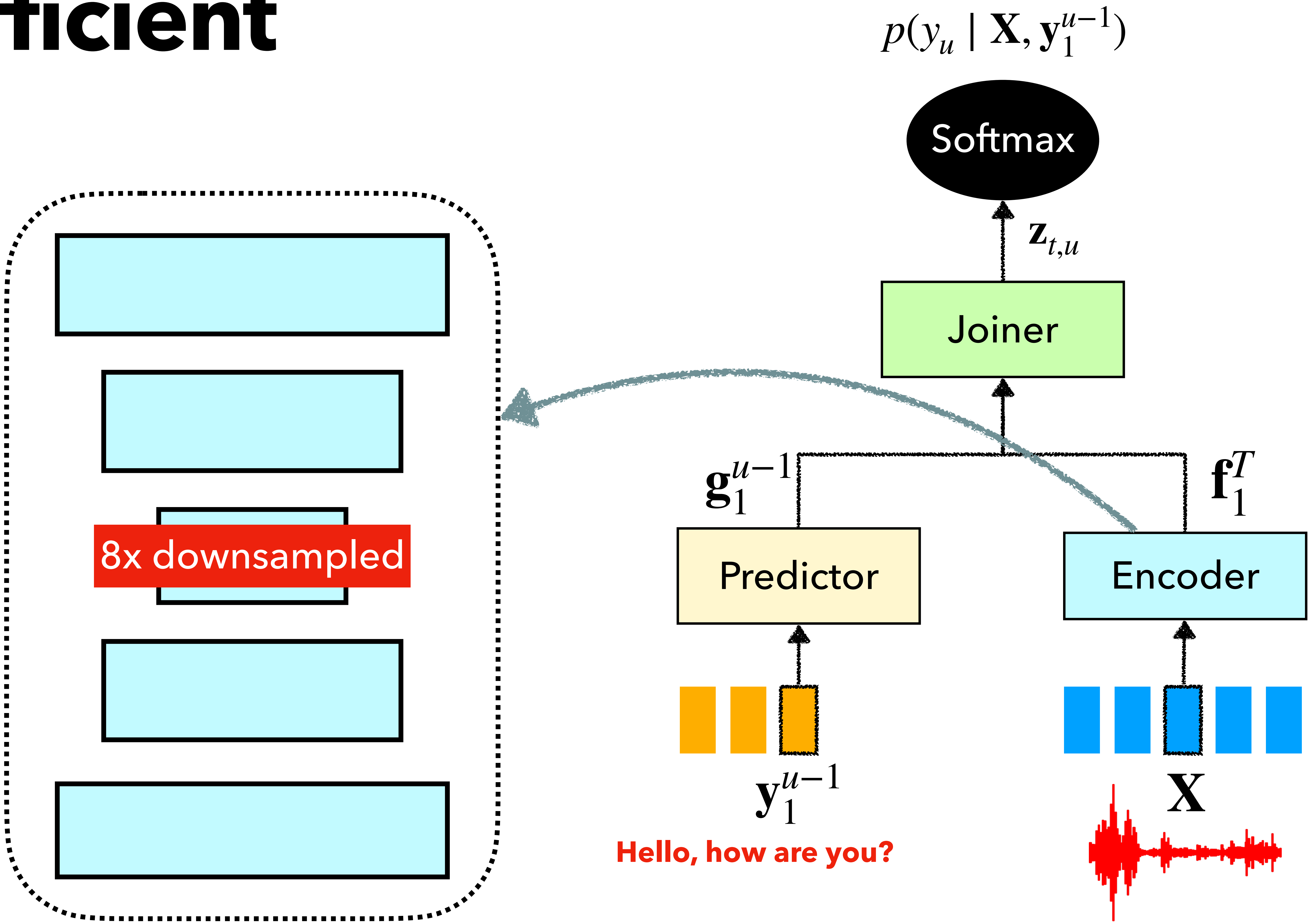
- Create synthetic mixtures from sub-segments instead of full-utterances
- Multiple turns of conversation more important than long single-speaker regions

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)



Yao, Zengwei et al. “Zipformer: A faster and better encoder for automatic speech recognition.” ICLR, 2024.

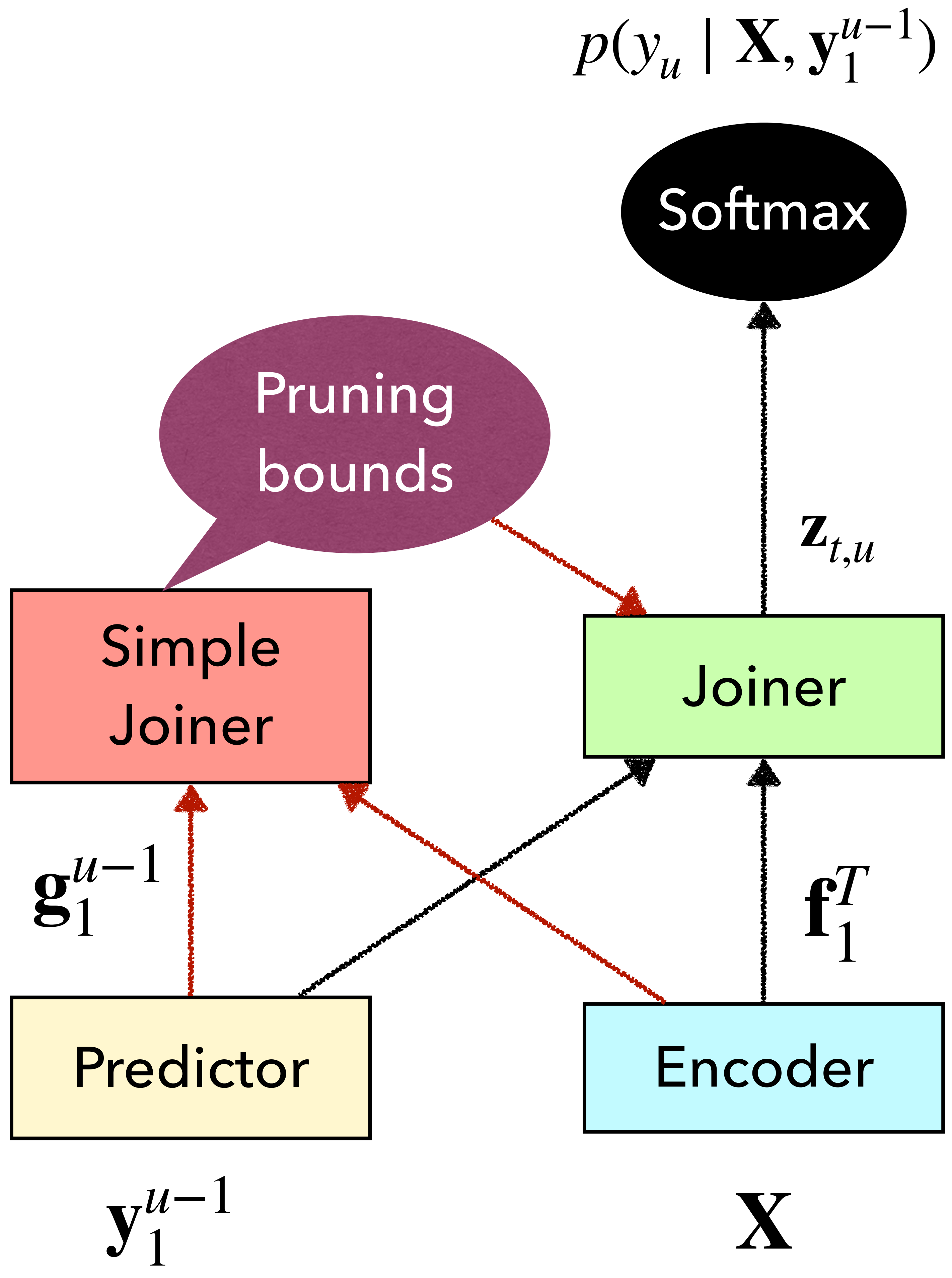
Problem 1

Making training efficient

#3: Pruned transducer loss

- Original transducer loss computes sum over all possible alignments
- Instead, pruned loss sums over a subset of alignments:

$$P(\mathbf{y} \mid \mathbf{X}) = \sum_{\mathbf{a} \in \mathcal{B}^{-1}_{\text{pruned}}(\mathbf{y})} P(\mathbf{a} \mid \mathbf{X})$$

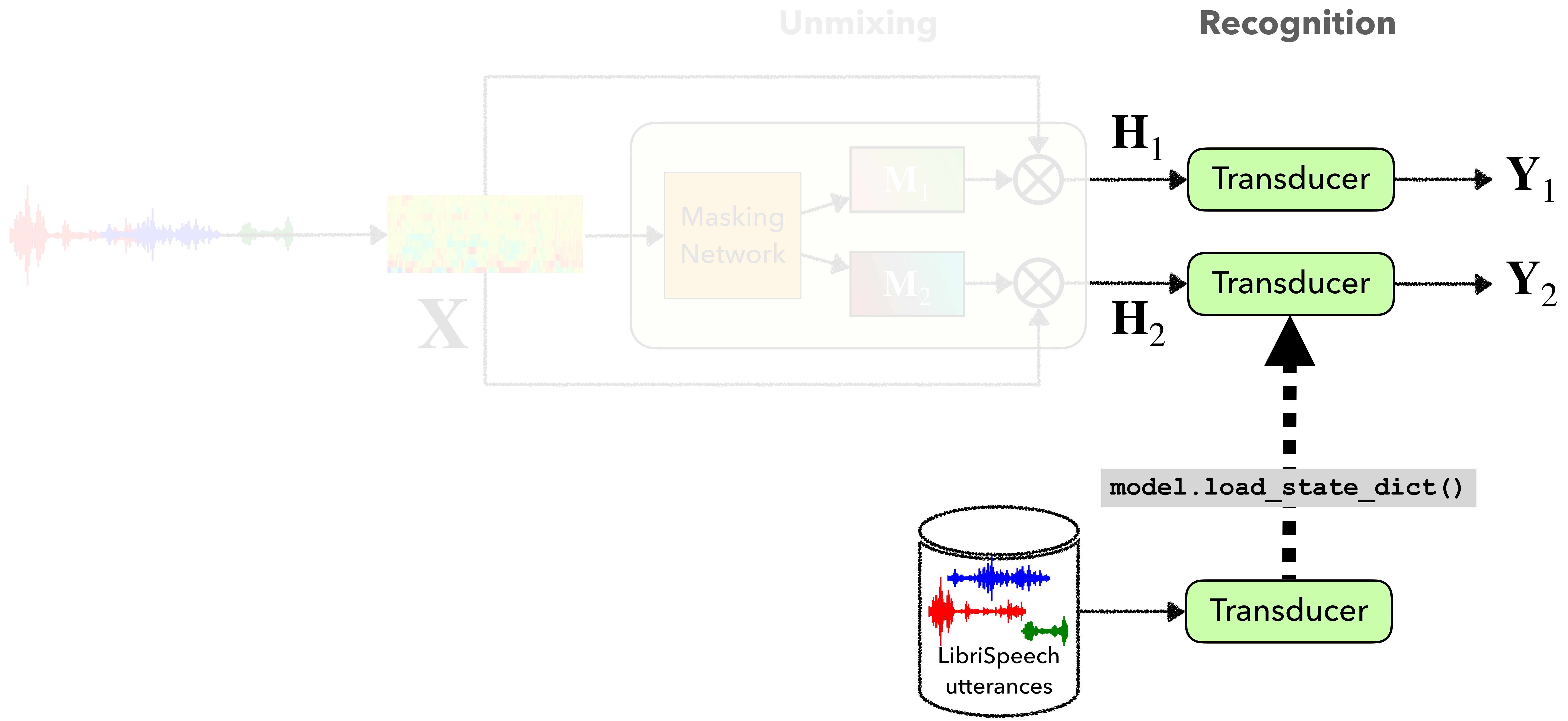


Kuang, F., Guo, L., Kang, W., Lin, L., Luo, M., Yao, Z., & Povey, D. Pruned RNN-T for fast, memory-efficient ASR training. *Interspeech 2022*.

Problem 1

Making training efficient

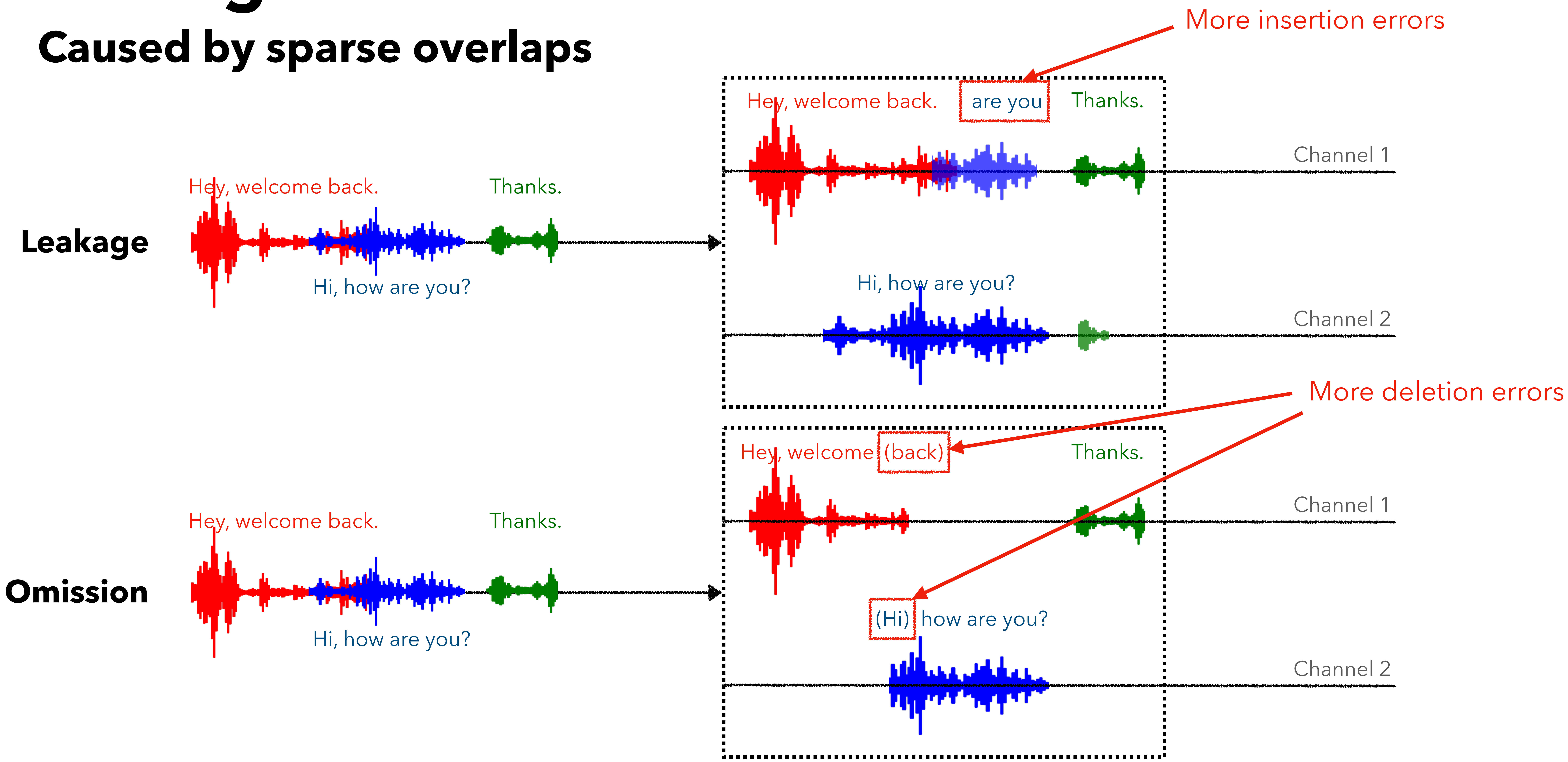
#4: Single-speaker pre-training



Problem 2

Leakage and omission errors

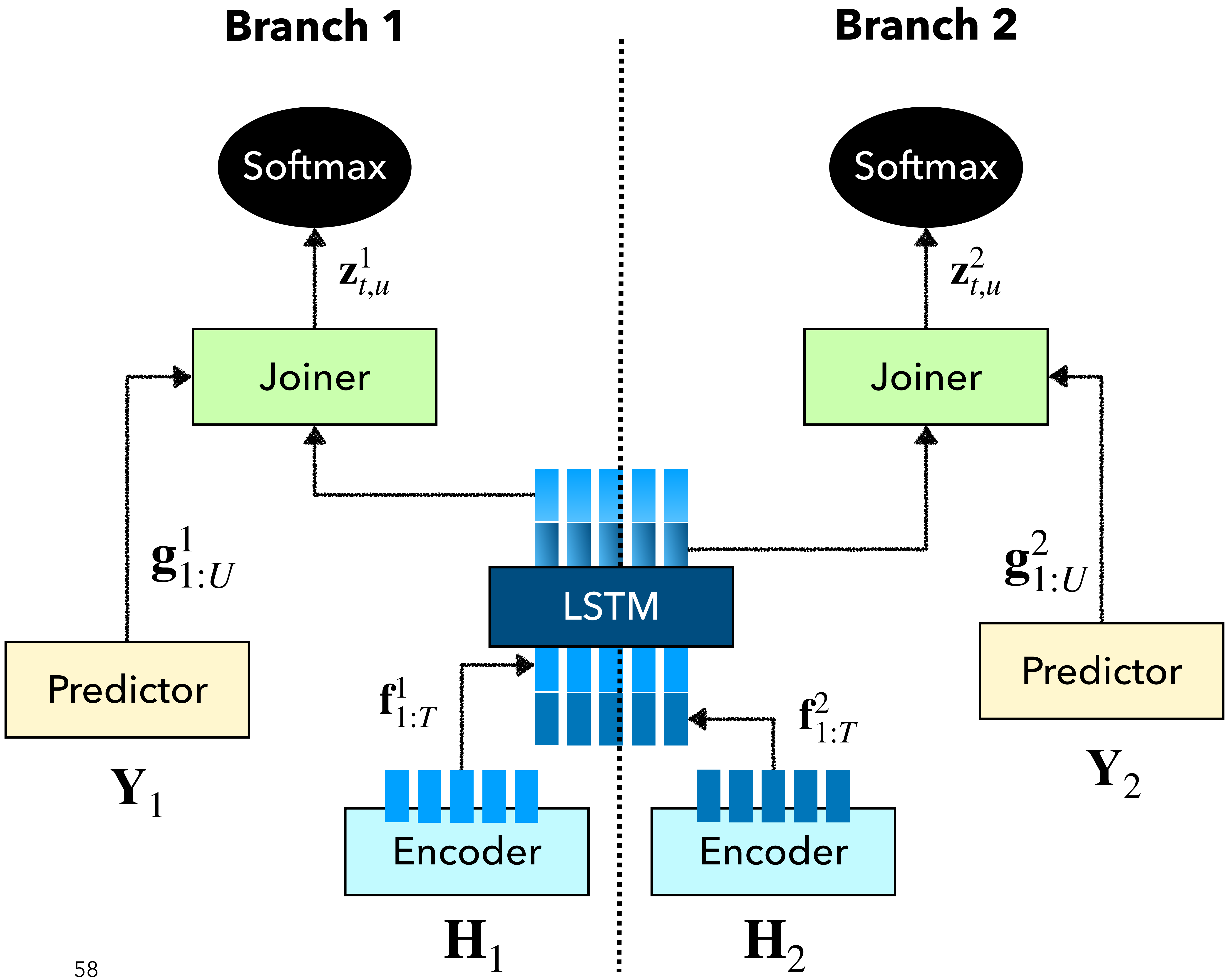
Caused by sparse overlaps



Leakage and omission errors

#1: Architectural changes

1. Masking network: use dual-path LSTM, which is better for separation
2. Encoder: use "branch tying"
3. Decoder: use "stateless" prediction network



Problem 2

Leakage and omission errors

#2: Masking loss and encoder CTC loss

We use 2 auxiliary loss functions:

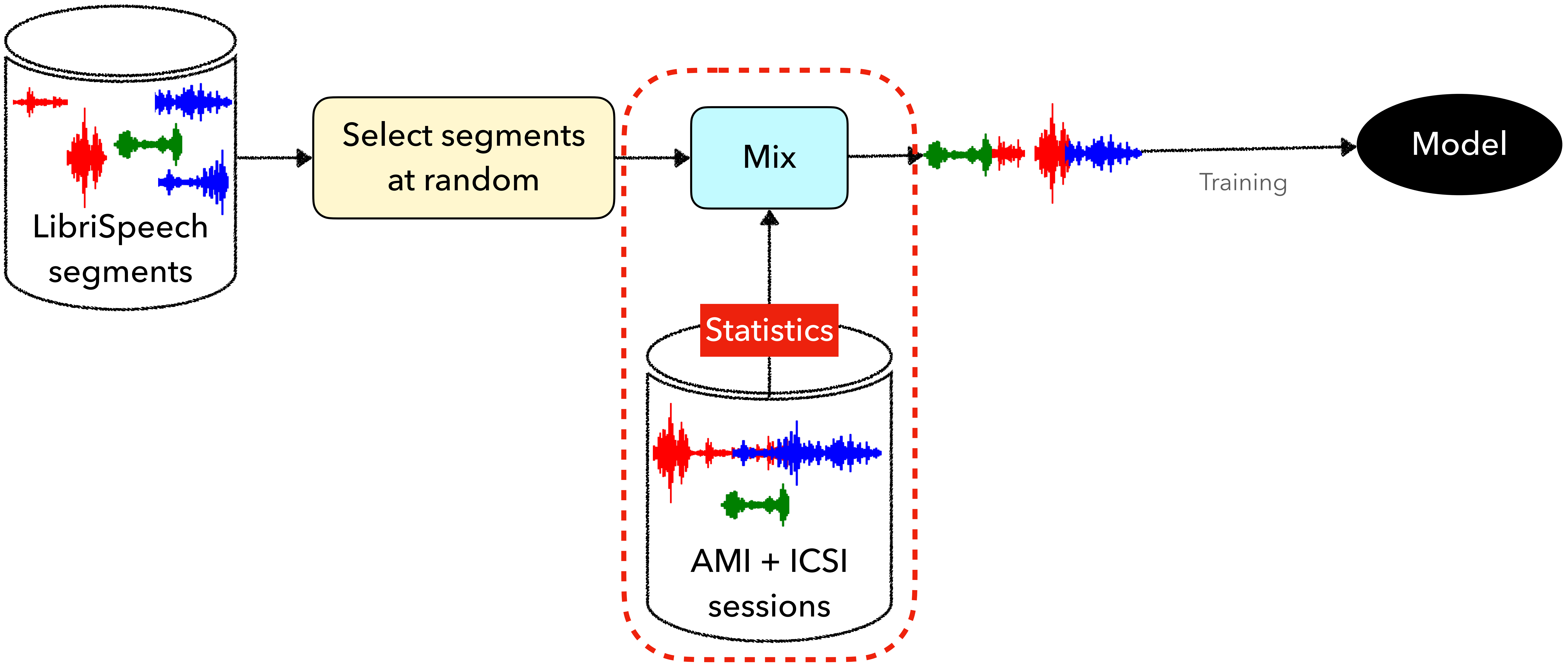
1. **CTC loss** at the output of the encoder (for better alignment)
2. **MSE loss** on the masked filterbanks (for better separation)

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

Problem 3

Performance on real meetings

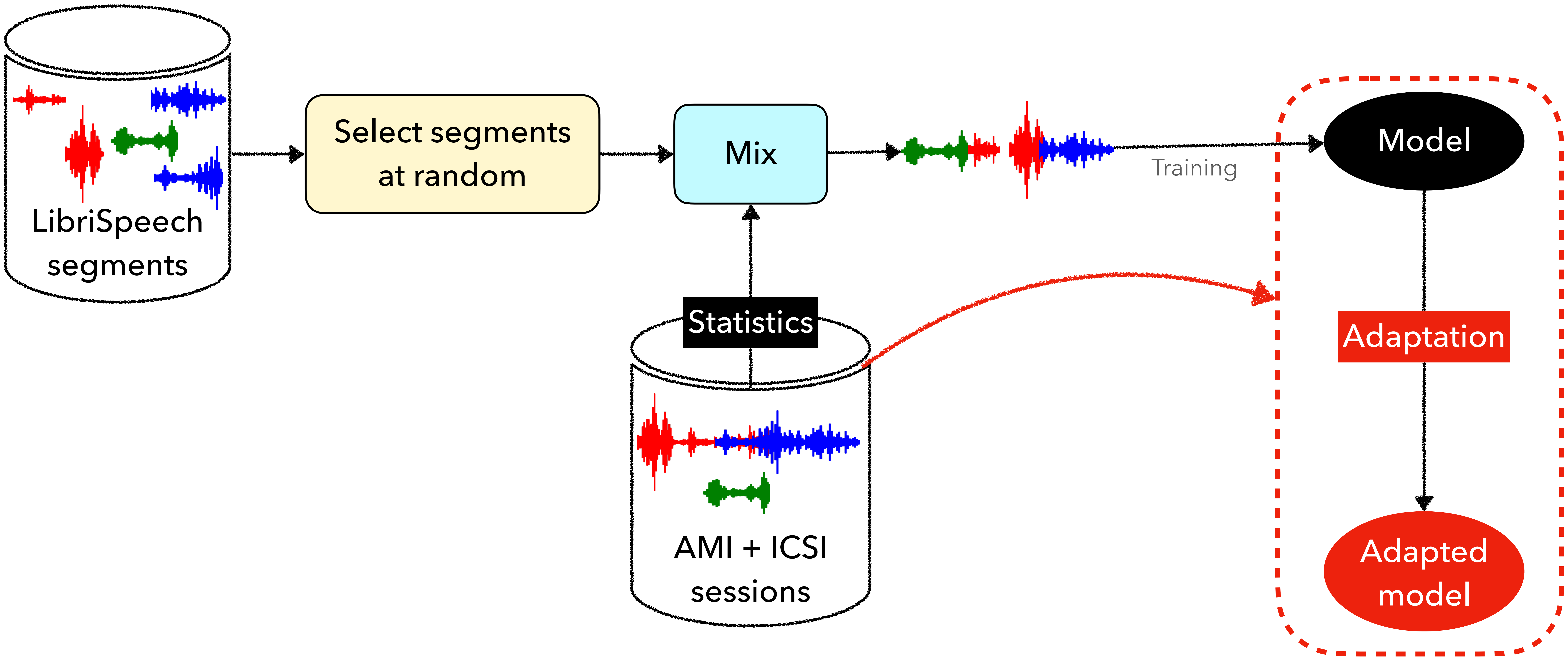
#1: Simulation using real meeting statistics



Problem 3

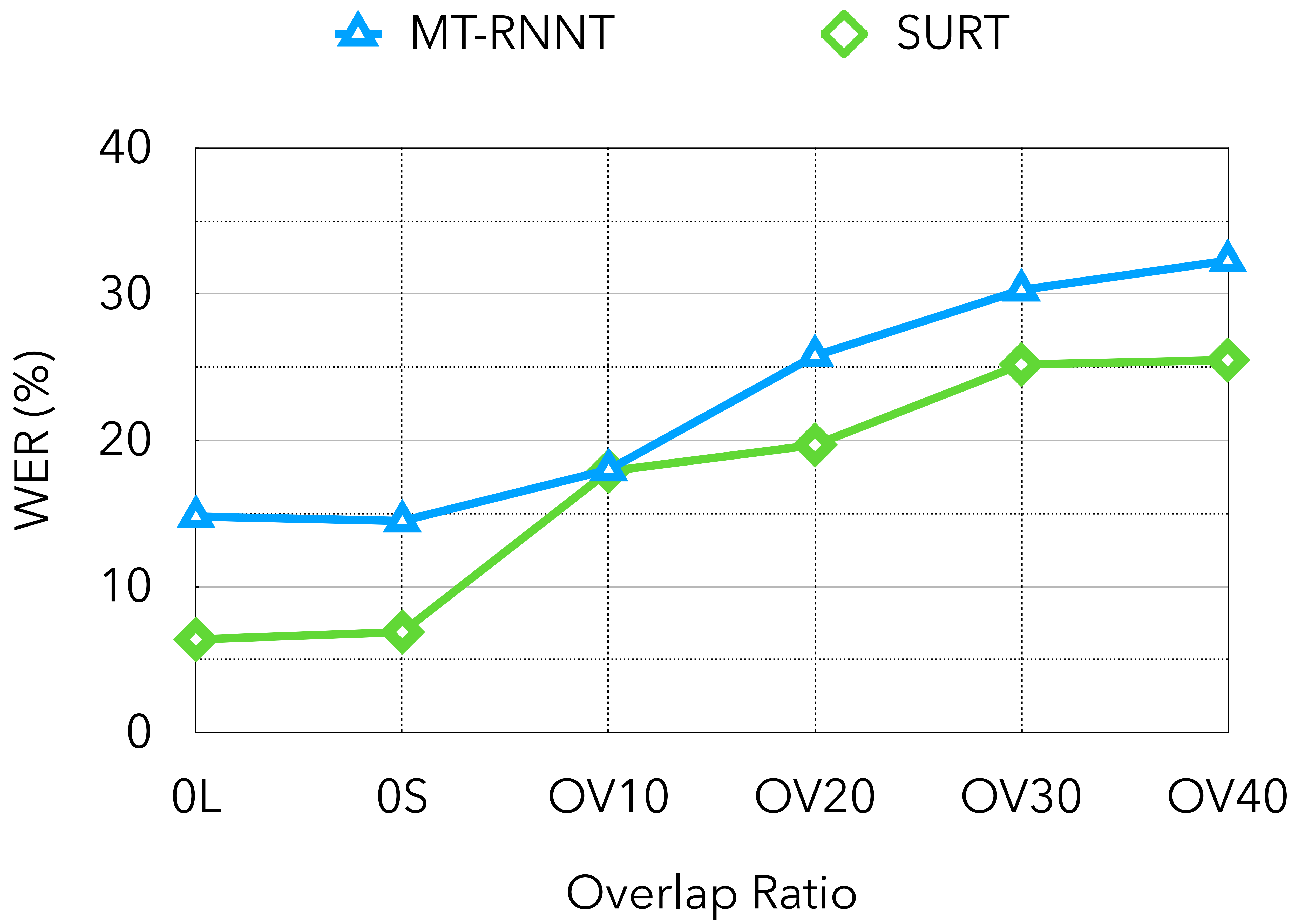
Performance on real meetings

#2: Domain adaptation



Results on LibriCSS

#1: SURT outperforms larger multi-turn RNN-T model



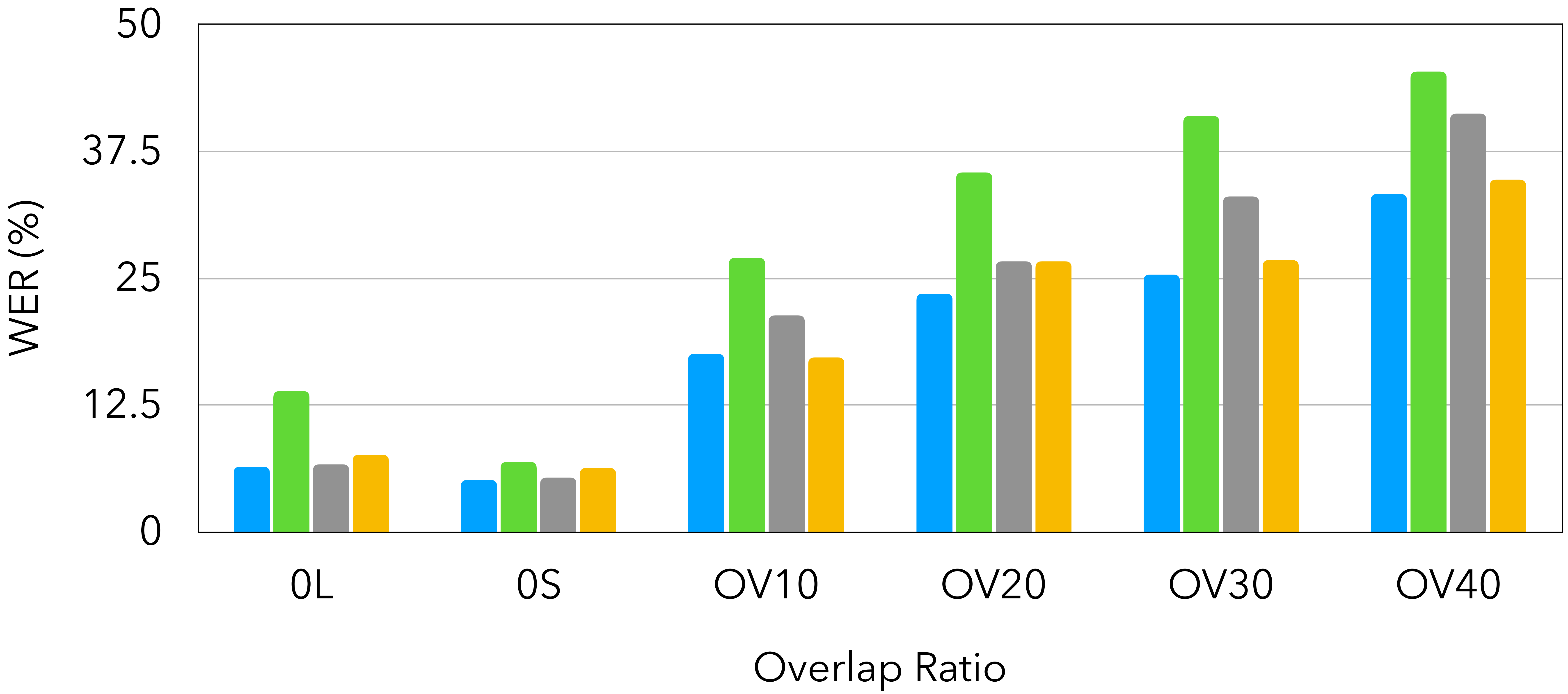
Model	# params (M)	WER (%)
MT-RNNT	81.0	22.6
SURT	**37.9**	**16.9**

Sklyar, Ilya et al. "Multi-Turn RNN-T for Streaming Recognition of Multi-Party Speech." IEEE ICASSP 2022: 8402-8406.

Results on LibriCSS

#2: Effect of architectural changes

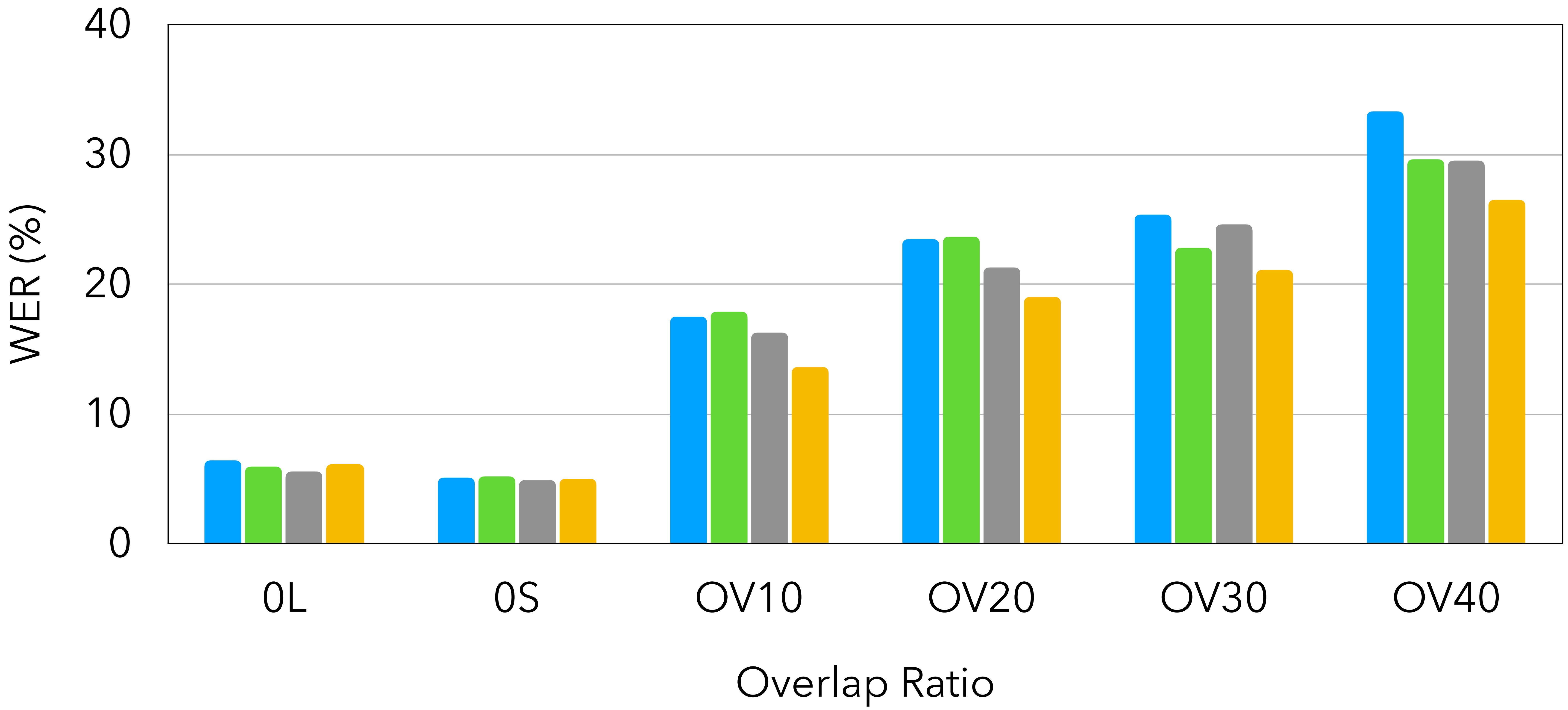
- Most improvement comes from using DP-LSTM in masking network.



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

Results on LibriCSS

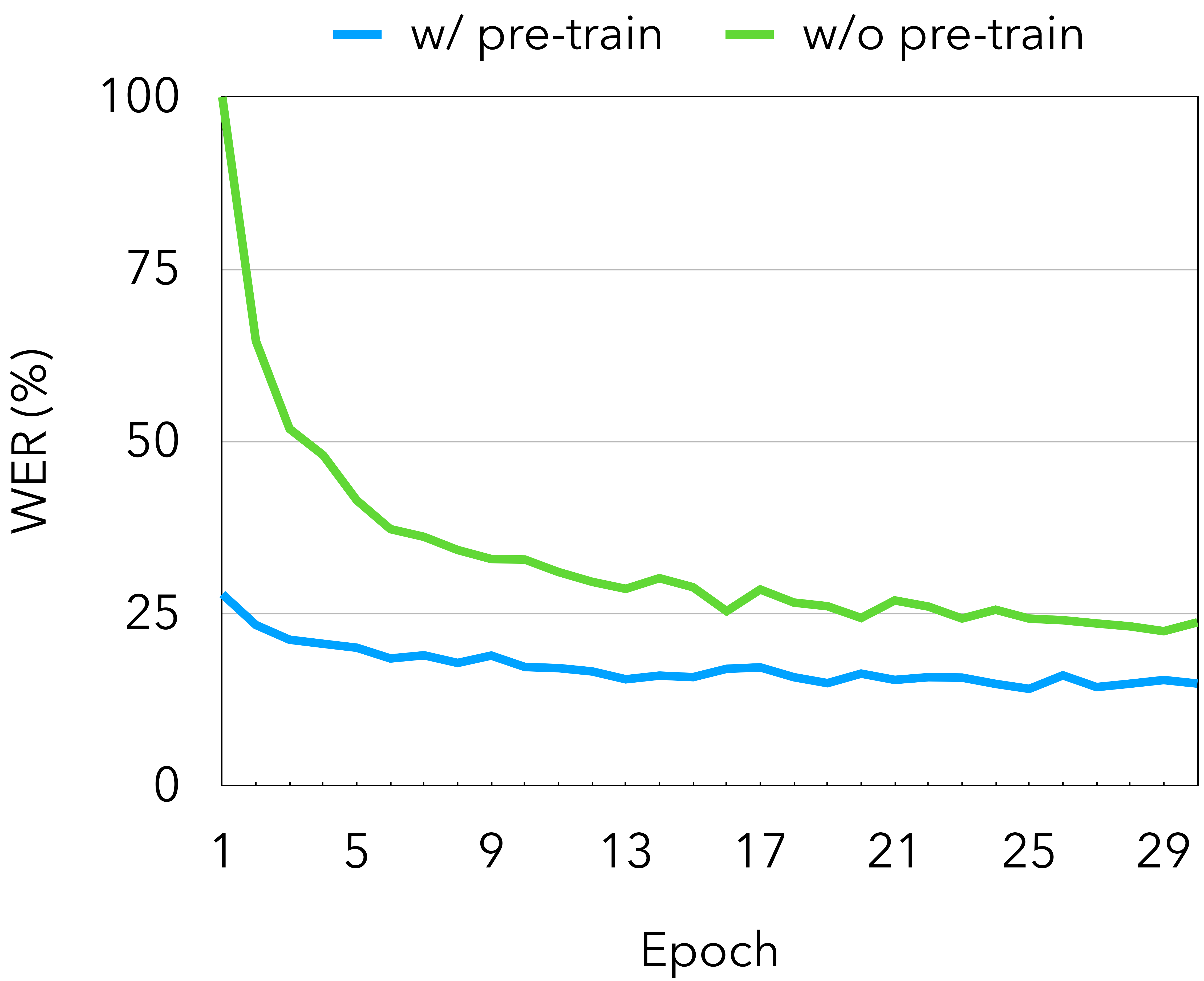
#3: Effect of auxiliary objectives



	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

IHM-Mix = close talk, SDM = far-field (single-channel)

AMI

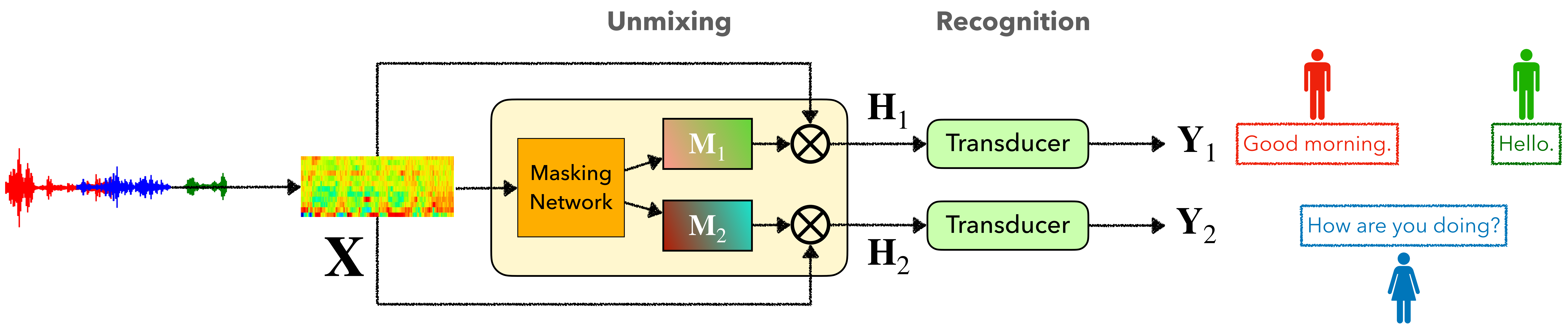
	IHM-Mix	SDM	MDM (beamform)
SURT	36.8	62.5	44.4
+ adapt.	35.1	44.6	41.4

ICSI

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

Speaker attribution with SURT

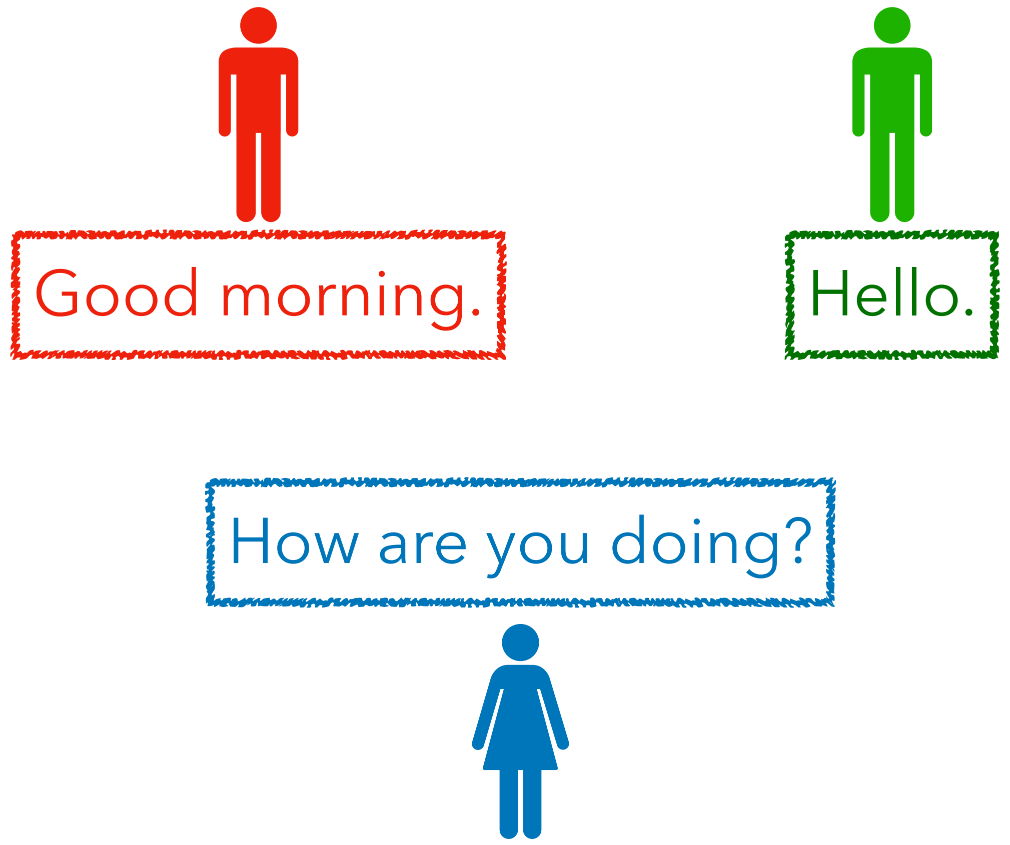
How to predict speaker labels with ASR tokens?



Speaker attribution with SURT

Heuristic error assignment training for speakers

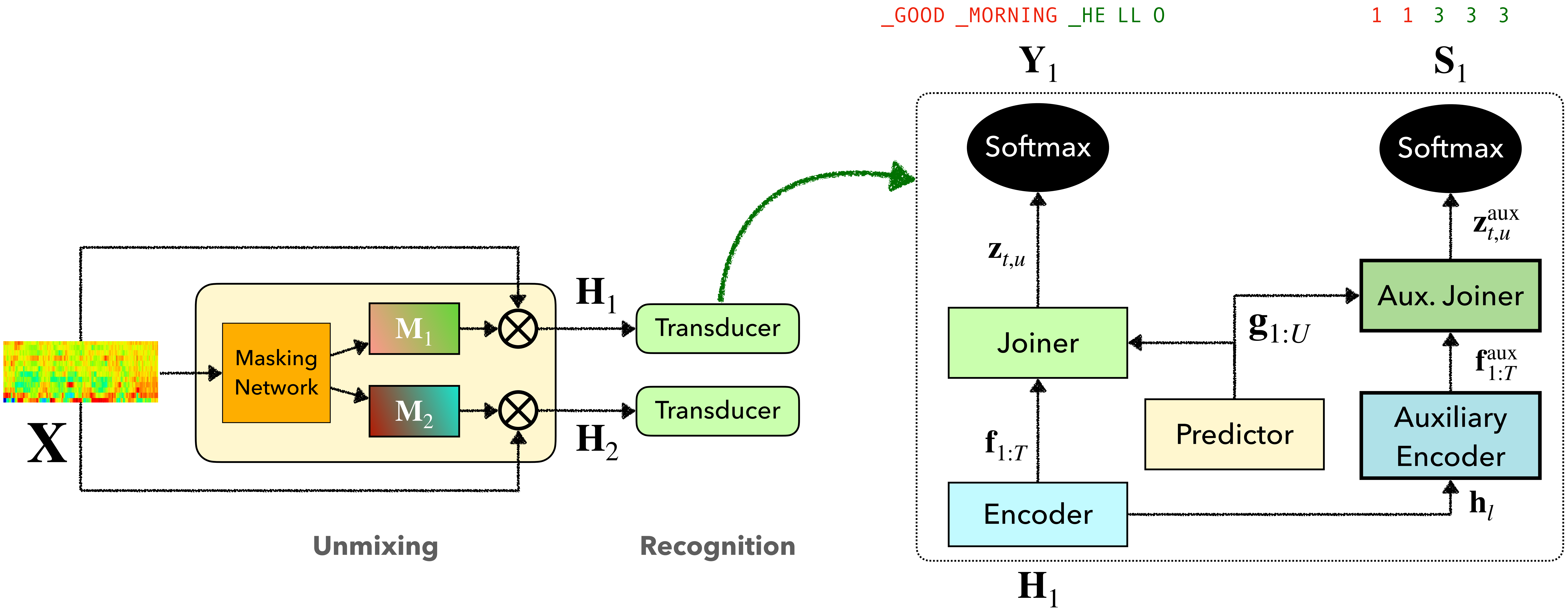
- Use the same 2-branch strategy, but predict speaker labels instead of ASR tokens
- How to do both tasks jointly?



$\mathbf{Y}_1$	_GOOD	_MORNING	_HE	LL	O
$\mathbf{S}_1$	1	1	3	3	3
$\mathbf{Y}_2$	_HOW	_ARE	_YOU	_DO	ING
$\mathbf{S}_2$	2	2	2	2	2

Speaker attribution with SURT

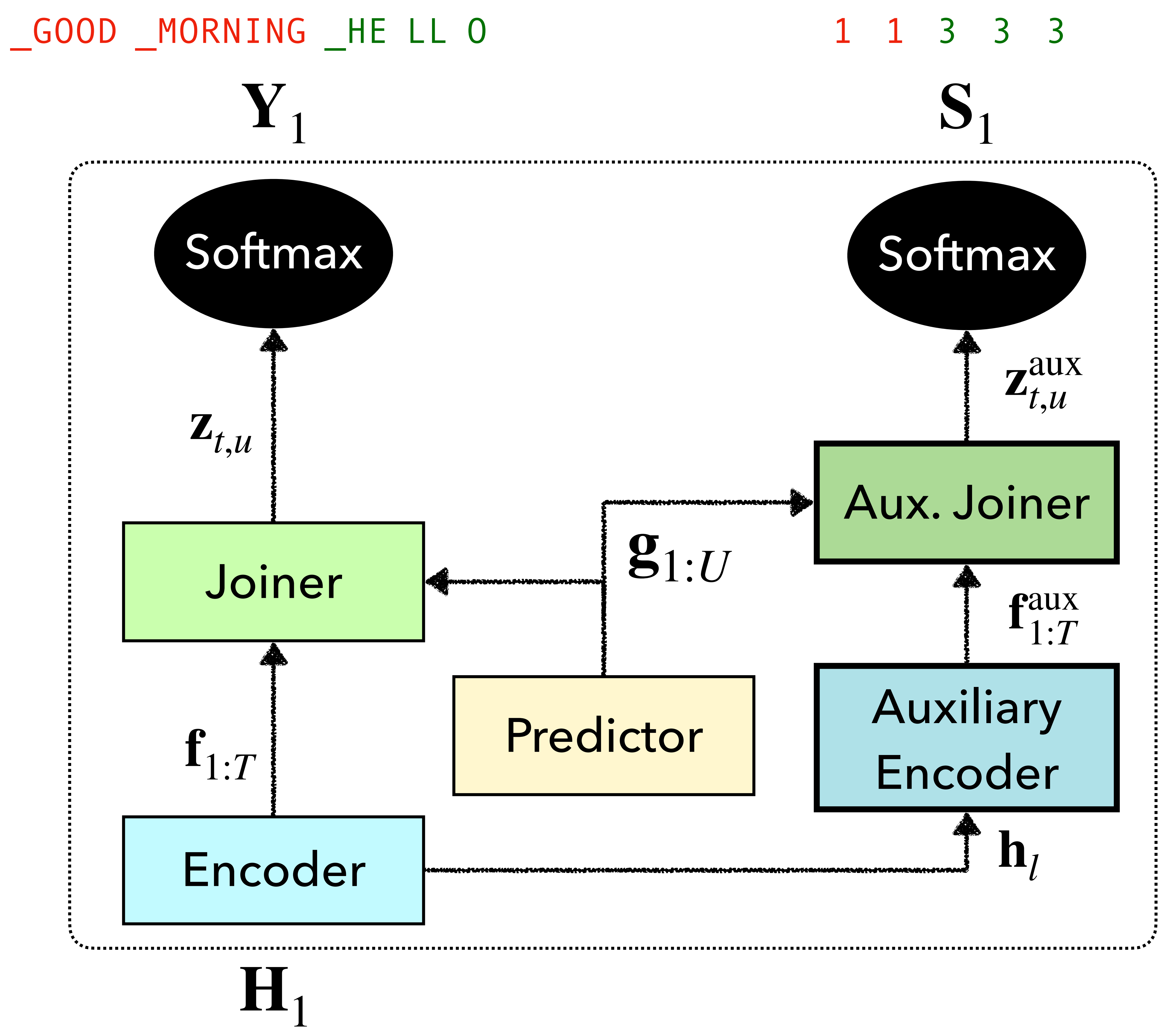
Auxiliary speaker encoder



Speaker attribution with SURT

Synchronizing speaker labels with ASR tokens

- At inference time, it is not necessary that both output streams emit same number of tokens.
- Even if they do, they may not be frame synchronous.



$\mathbf{Y}_1$	<blk>	_GOOD	_MORNING	<blk>	_HE	<blk>	LL	O
$\mathbf{S}_1$	<blk>	1	<blk>	1	<blk>	3	<blk>	3

Speaker attribution with SURT

Hybrid autoregressive transducer (HAT)

RNN-Transducer

$$P(\mathbf{a}_t \mid \mathbf{f}_1^t, \mathbf{g}_1^{u(t)-1}) = \text{Softmax}(\mathbf{z}_{\text{t,u}})$$

- Multinomial distribution over blank and non-blank tokens
- Cannot model blank probability separately

HAT

$$P(\mathbf{a}_t \mid \mathbf{f}_1^t, \mathbf{g}_1^{u(t)-1}) = \begin{cases} b_{t,u}, & \text{if } \mathbf{a}_t = \phi, \\ (1 - b_{t,u}) \ \text{Softmax}(\mathbf{z}_{\text{t,u}}[1:]), & \text{otherwise} \end{cases} \qquad b_{t,u} = \sigma(\mathbf{z}_{t,u}[0])$$

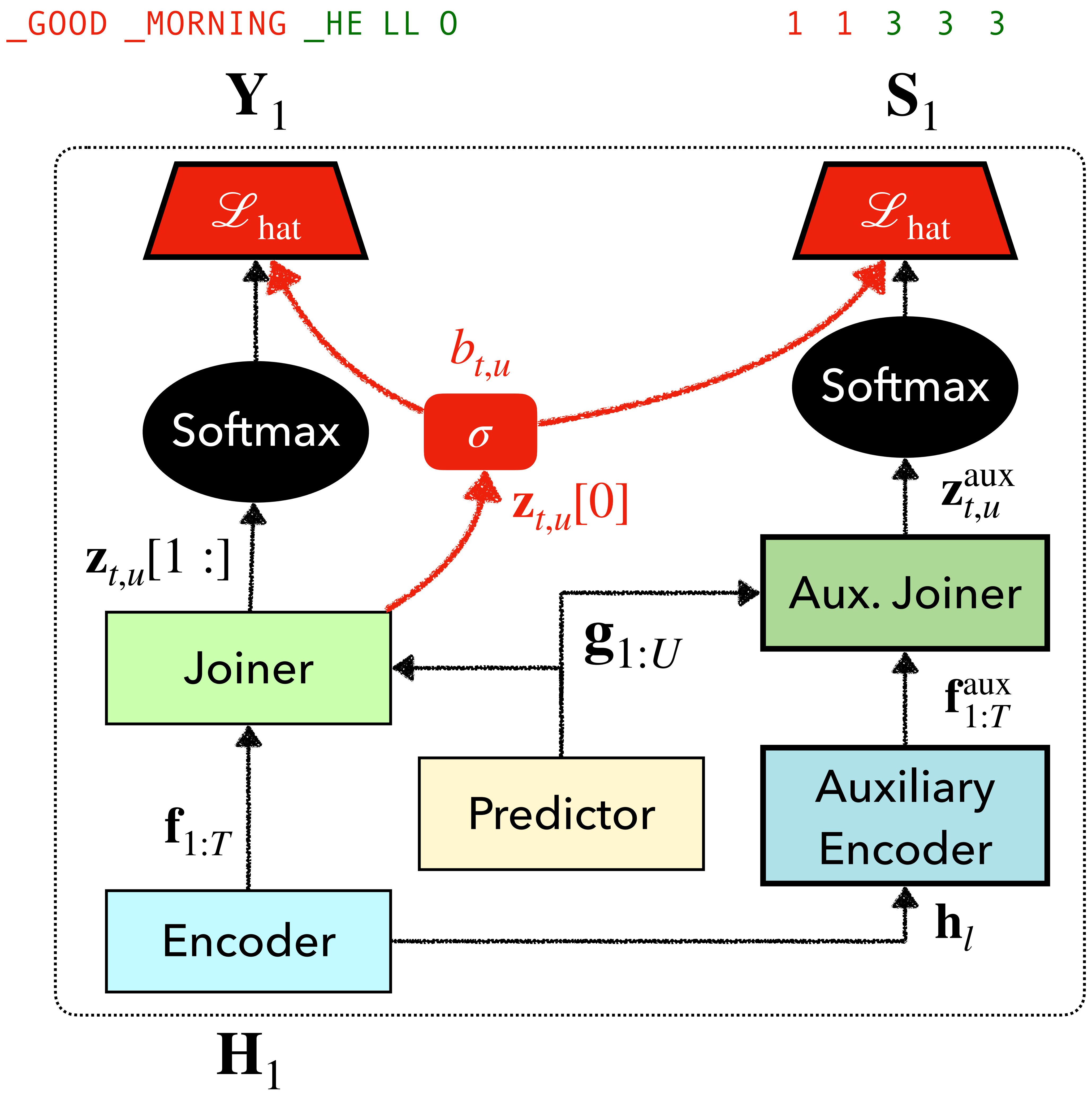
- Bernoulli distribution for blank; multinomial over non-blank tokens
- Probability of blank given directly by $b_{t,u}$

Variani, Ehsan et al. "Hybrid Autoregressive Transducer (HAT)." *IEEE ICASSP 2020*.

Speaker attribution with SURT

Synchronization by sharing `<blk>`

- If ASR branch emits `<blk>` do the same for speaker branch
- This is achieved by using HAT-style blank factorization, and sharing blank logit between ASR and speaker branch



Speaker attribution with SURT

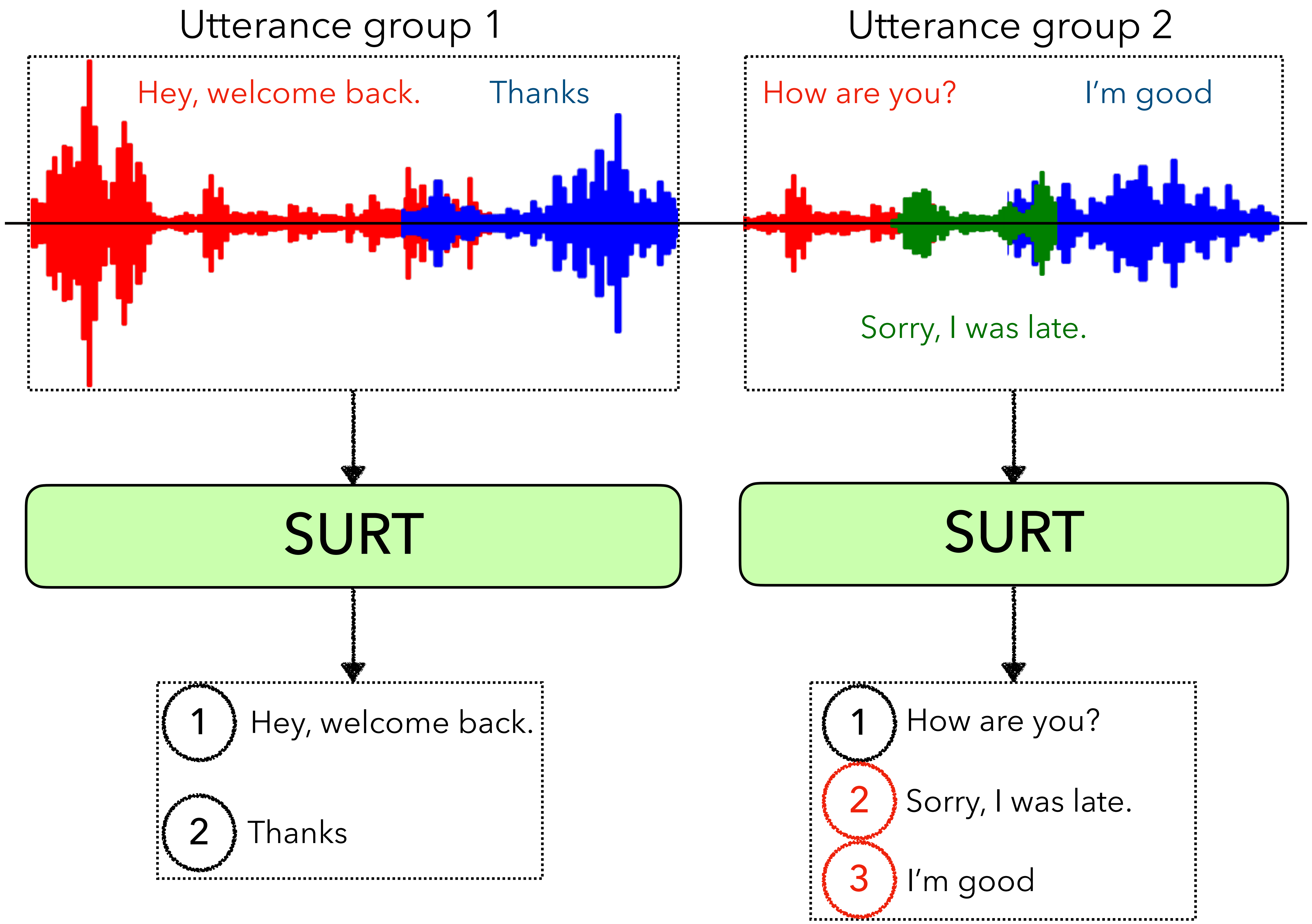
Results on AMI (evaluation on utterance groups)

Utterance group = set of utterances connected by overlaps or short pauses

			Streaming		Offline
Mic Setting	**ORC-WER**	**WDER**	**cpWER**		**Modular System cpWER**
IHM-Mix	34.9	9.3	42.3		–
SDM	43.2	10.9	50.3		38.5
MDM (beamformed)	40.5	9.9	47.3		31.0

Speaker attribution with SURT

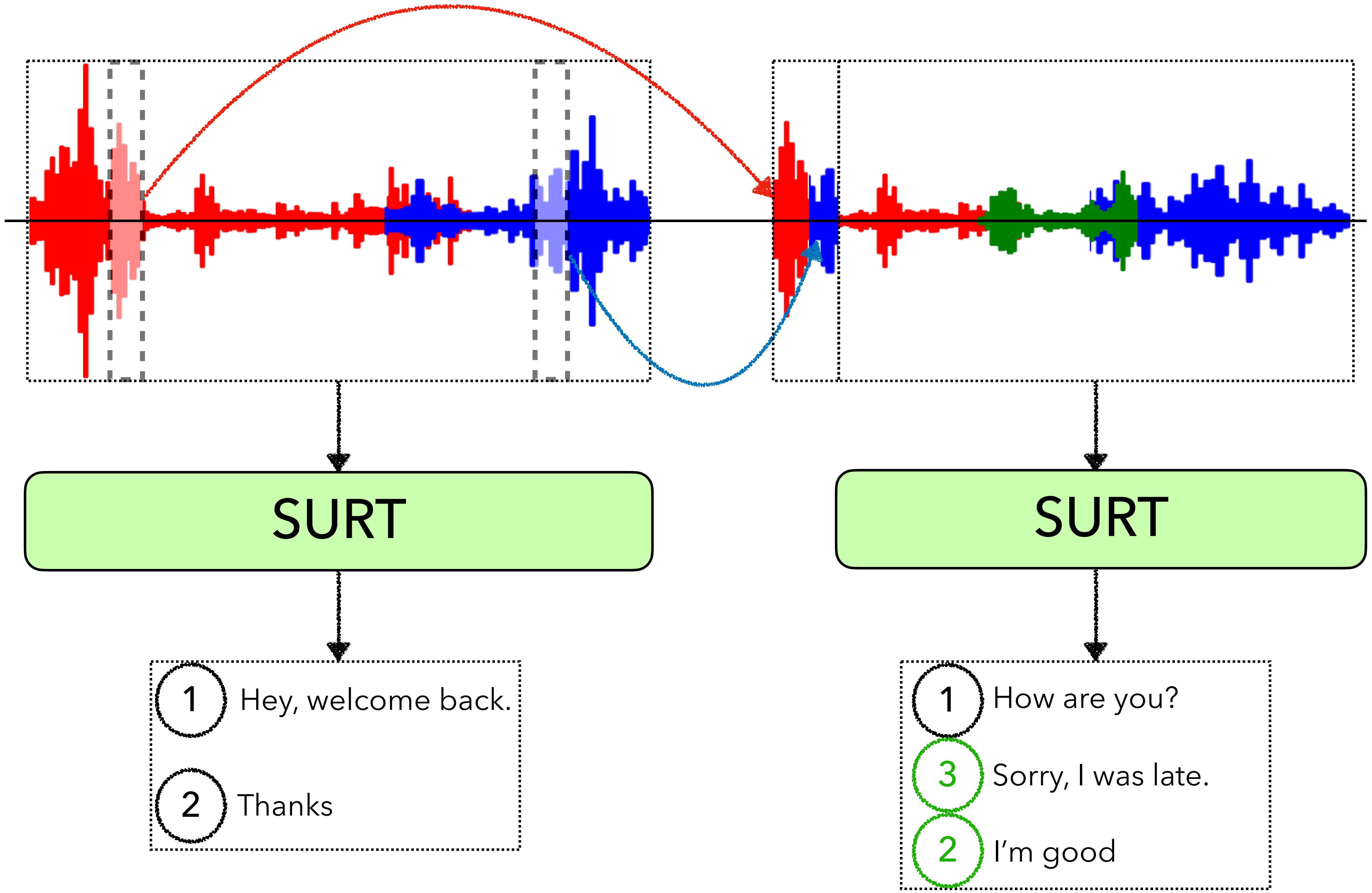
From utterance groups to full sessions



- How to maintain relative speaker labels when processing different utterance groups within the same session?

Speaker attribution with SURT

Speaker prefixing approach



- Extract high-confidence frames of predicted speakers and prefix them in front of current input.
- Remove prefixed part from encoder representation.

Speaker attribution with SURT

Evaluation on AMI IHM-Mix setting

"Enrollment" = using small chunk from speaker's enrollment speech for prefixing

Evaluation	Method	cpWER
Utterance group	SURT w/o speaker prefix	42.3
Full session	SURT w/o speaker prefix	100.1
	SURT w/ speaker prefix (128 frames = 1.28s per speaker)	82.8
	+ enrollment	53.8

Conclusions and Future Work

Conclusions

- Modular system is an **approximate solution** for the probabilistic formulation of multi-talker ASR problem.
- Provides **flexibility** of components, but **errors propagate**.
- For end-to-end modeling, we extended neural transducers for multi-talker ASR, resulting in the **SURT** model.
- We demonstrated how to train SURT efficiently, and how to **jointly predict** ASR tokens and speaker labels with the model.
- **Single model** to perform speaker-attributed transcription!

Future Work

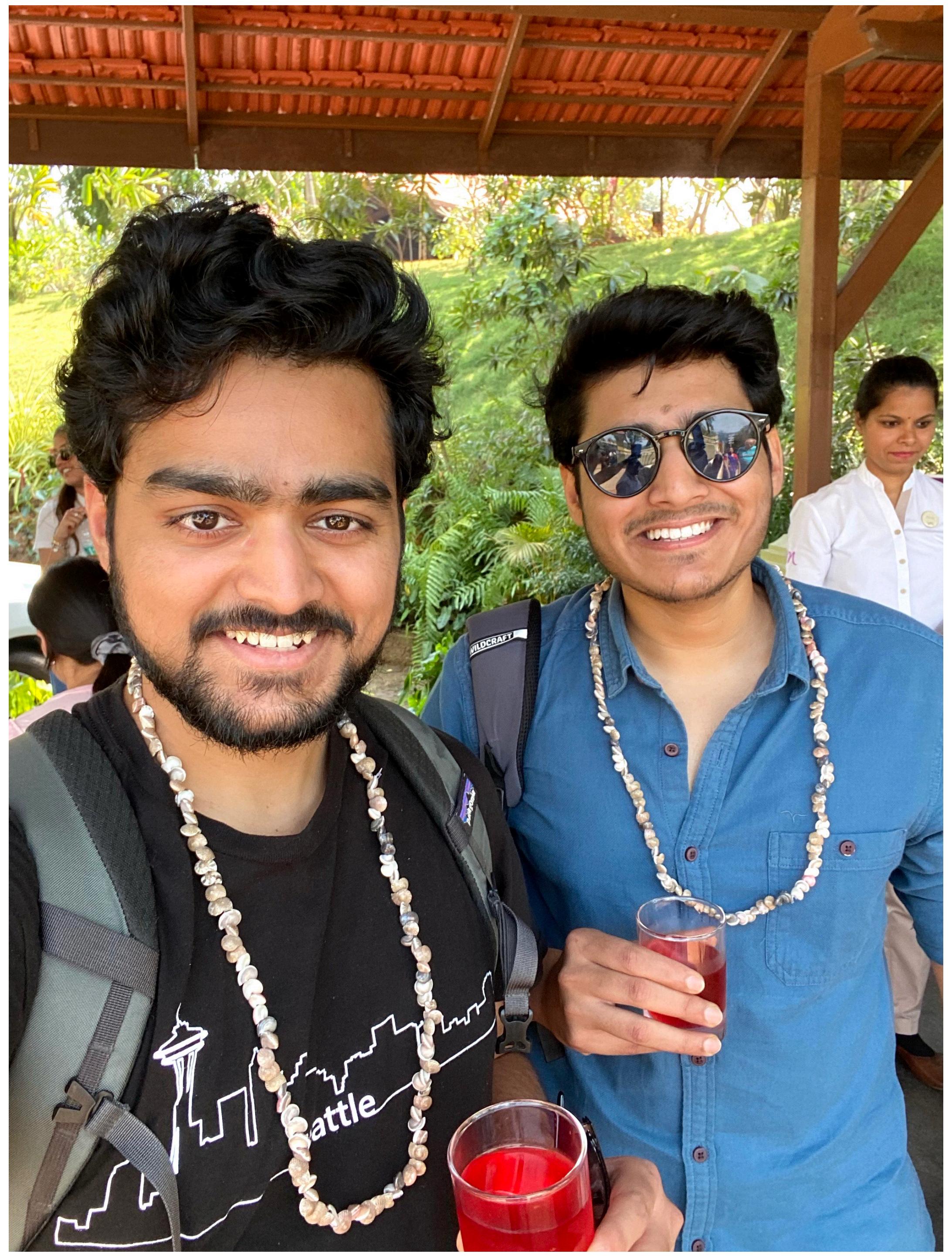
Improving the accuracy

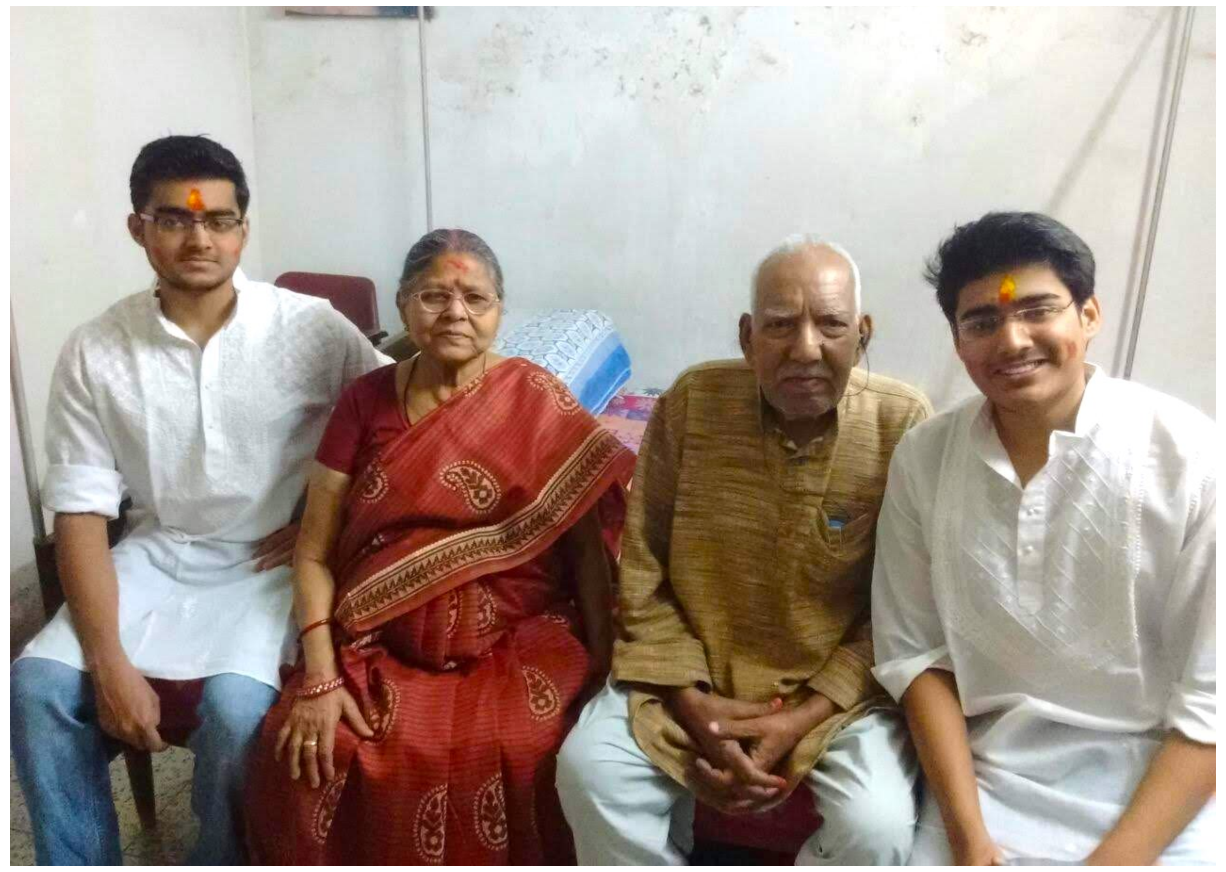
- `MODELING` Full session evaluation has high error rates → *speaker tracking with latent embeddings?*
- `TRAINING` Using larger models → *teacher-student training for the encoder?*
- `DECODING` Search errors in ASR/speaker modeling → *speaker-guided beam search?*
- `DECODING` Rescoring the whole conversation → *possible application of LLMs?*

Improving the efficiency

- `MODELING` Two branch strategy is wasteful → *multi-blank modeling?*
- `TRAINING` Deeper integration of ASR and speaker encoders → *revisit joint training?*

Thanks!



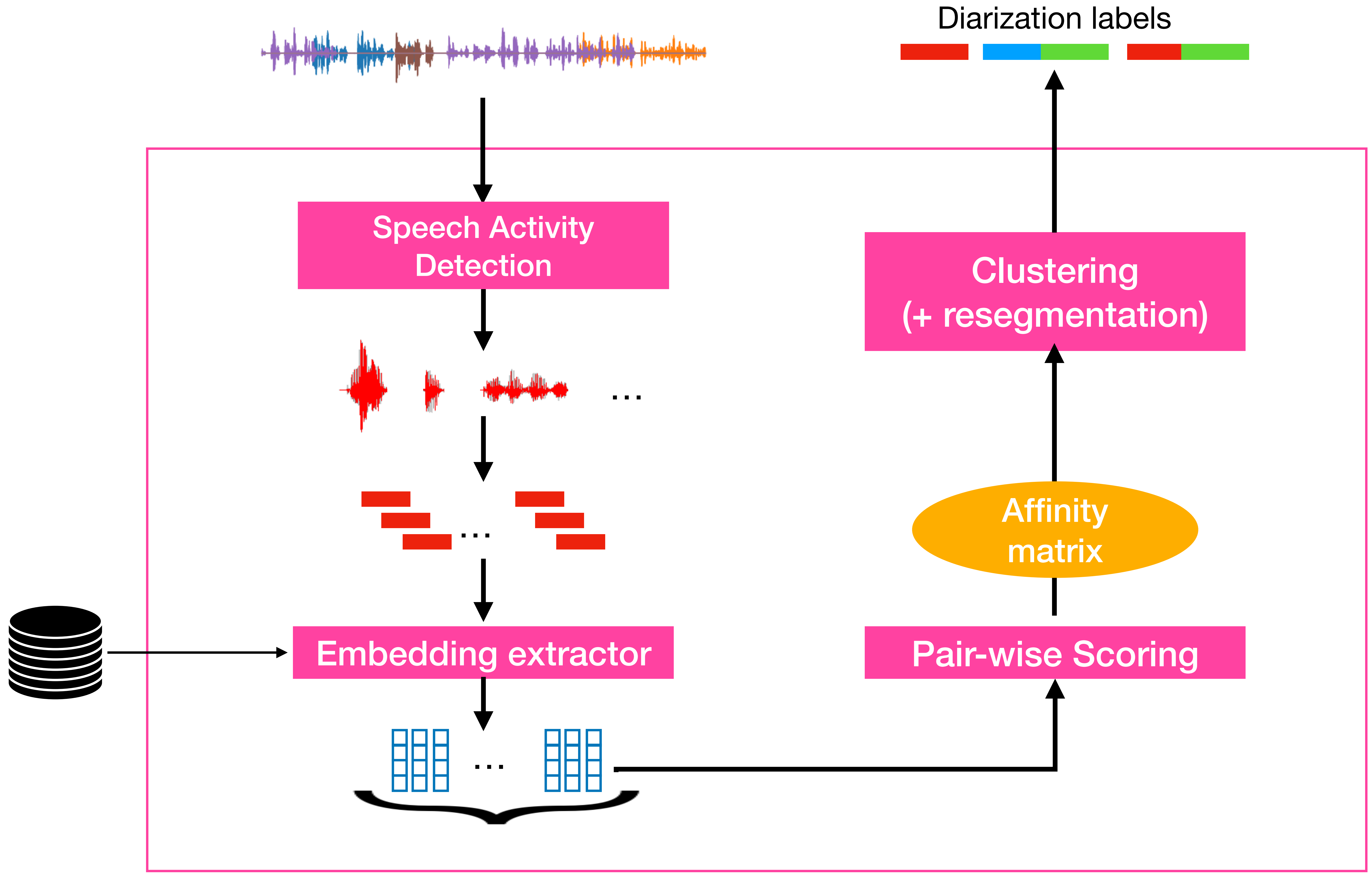


Extra Slides

Overlap-aware Spectral Clustering

Clustering-based diarization

Overview of the process



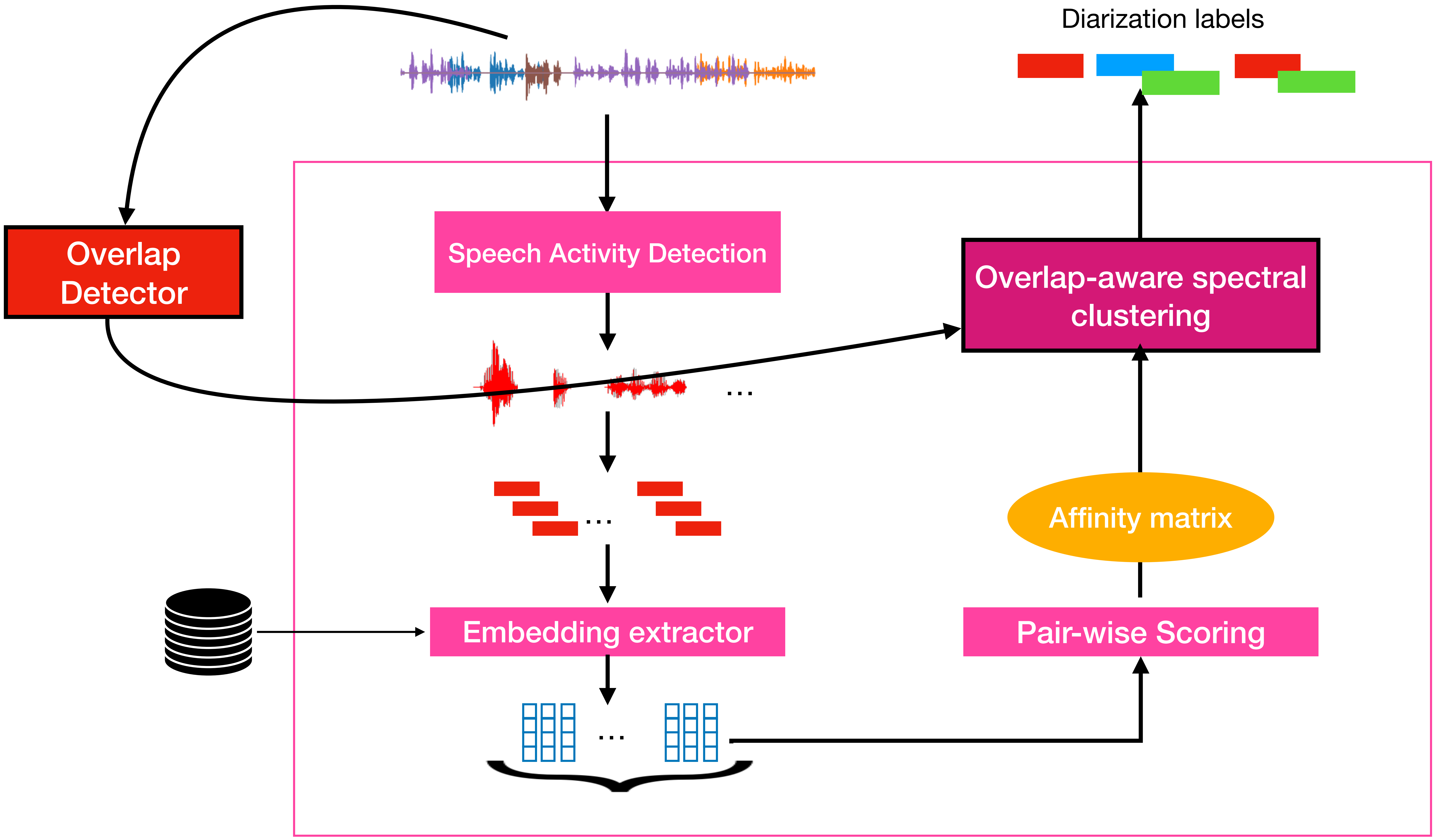
Clustering paradigm assumes single-speaker segments

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)

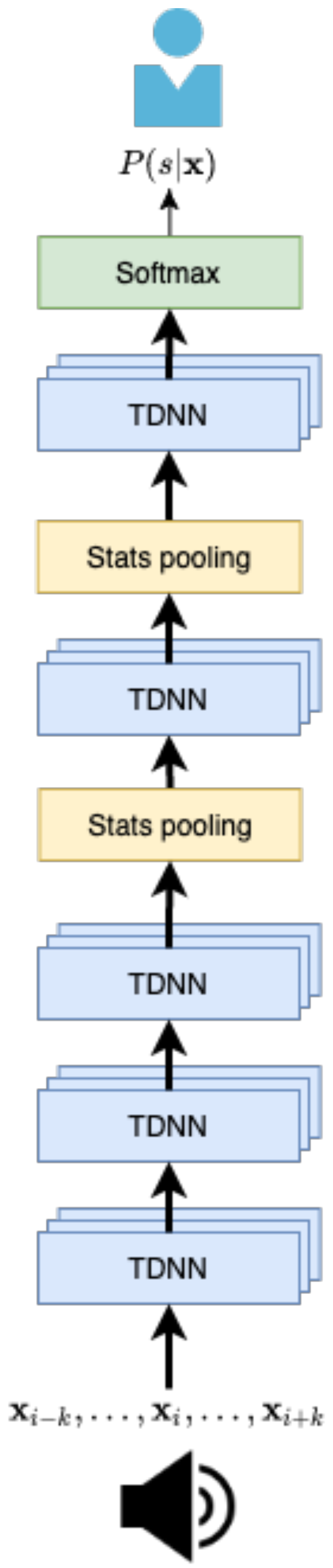
*"Given the current performance of the systems, the **overlapped speech gains more relevance** … **more than 50% of the DER** in our best systems … has to be addressed in the future …"*
- BUT team (2019)

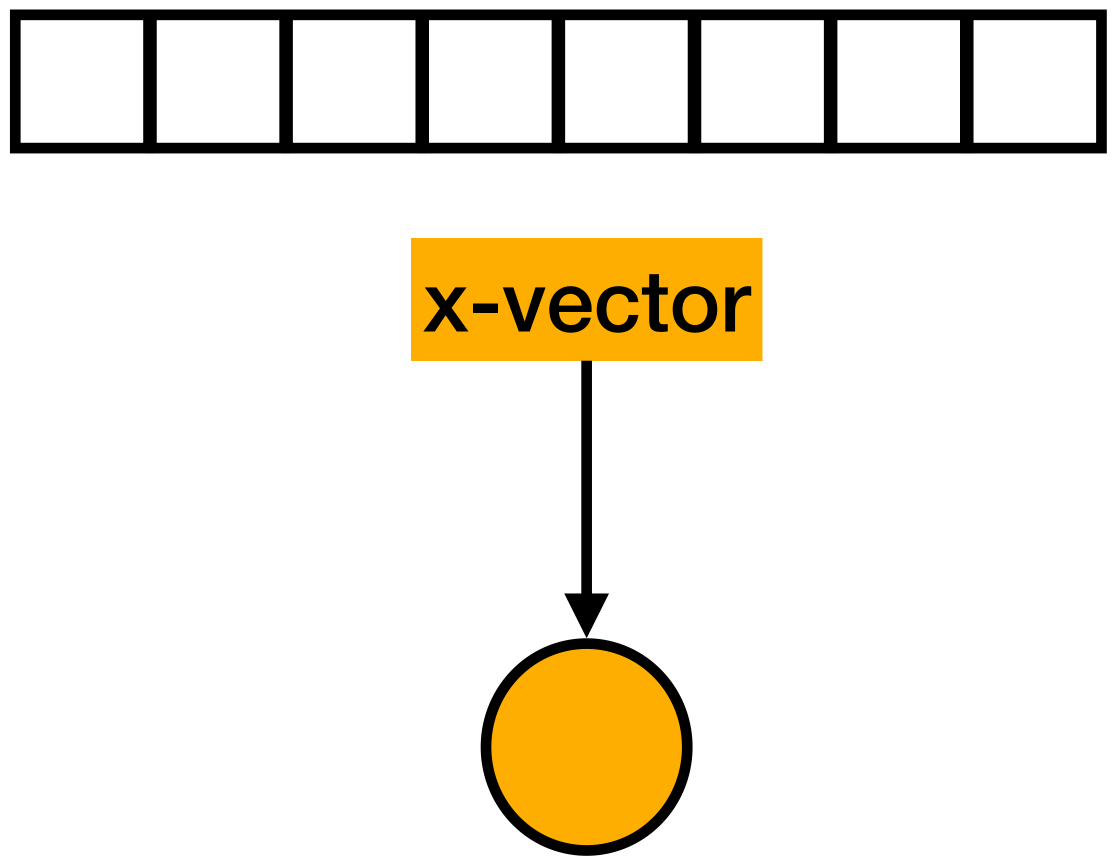
Overlap-aware spectral clustering



New formulation for spectral clustering

The basic clustering problem: a graph view

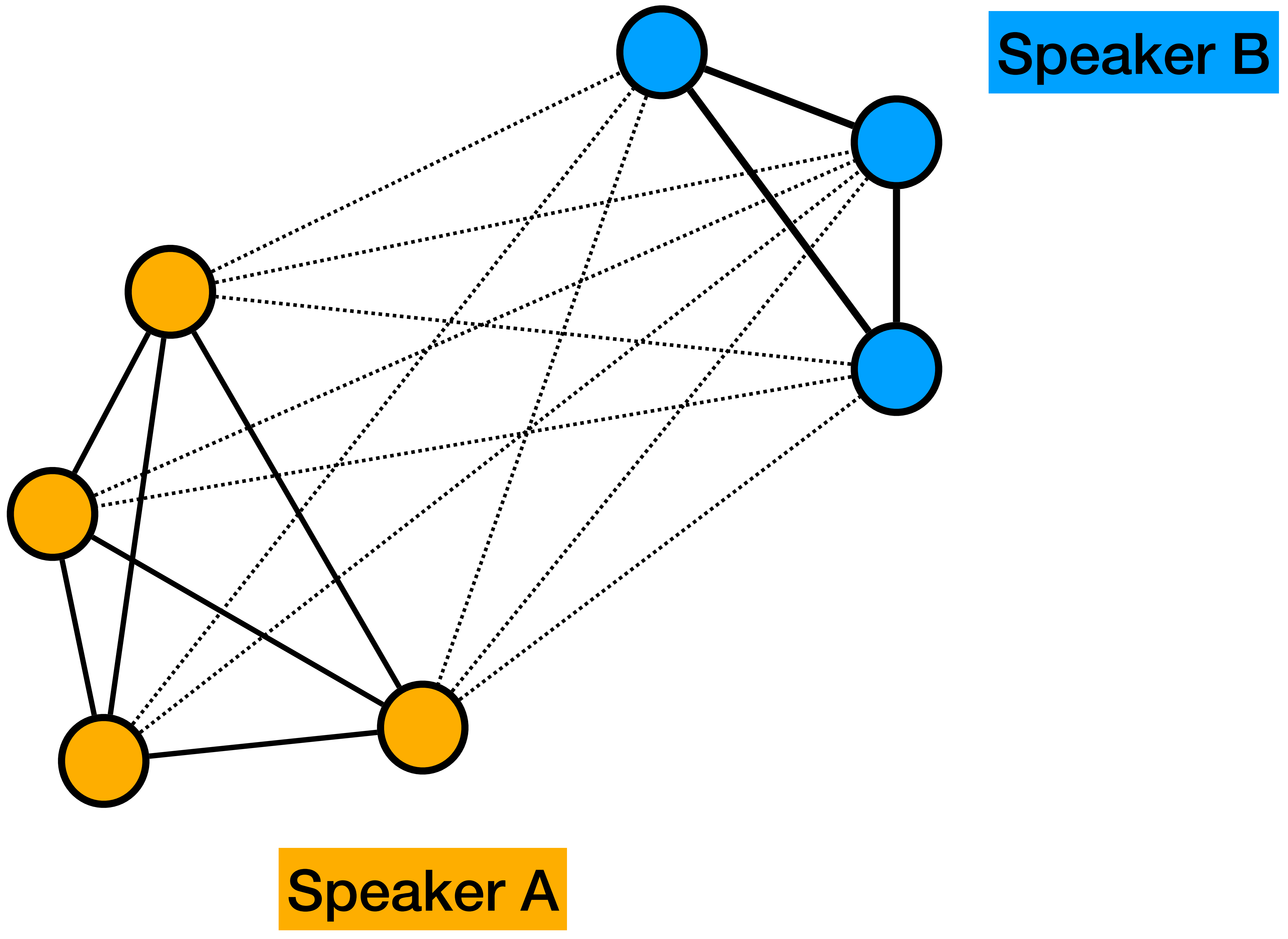




Cosine similarity

New formulation for spectral clustering

The basic clustering problem: a graph view

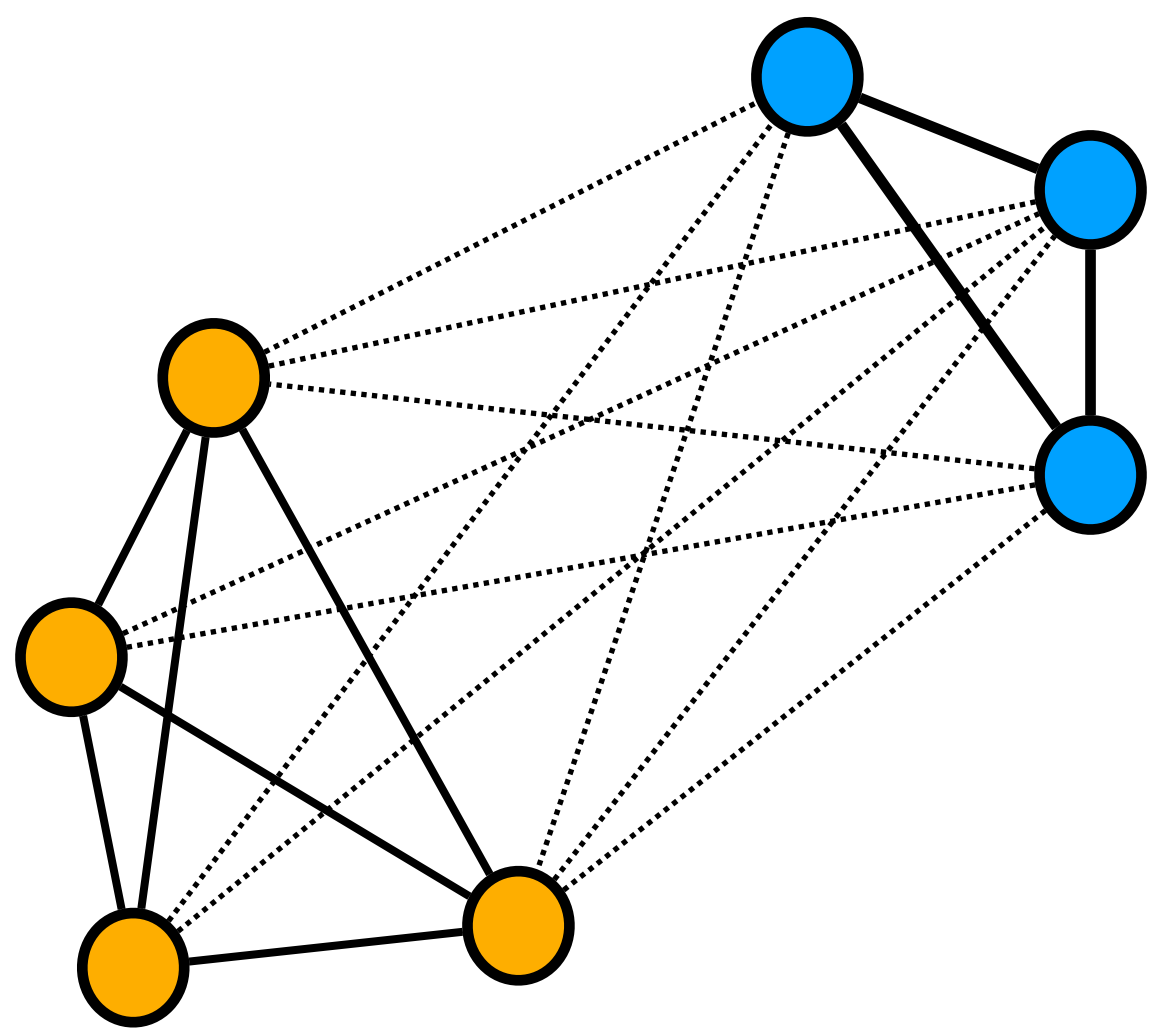


Edge weights within a group ↑

Edge weights across groups ↓

New formulation for spectral clustering

The basic clustering problem: a graph view



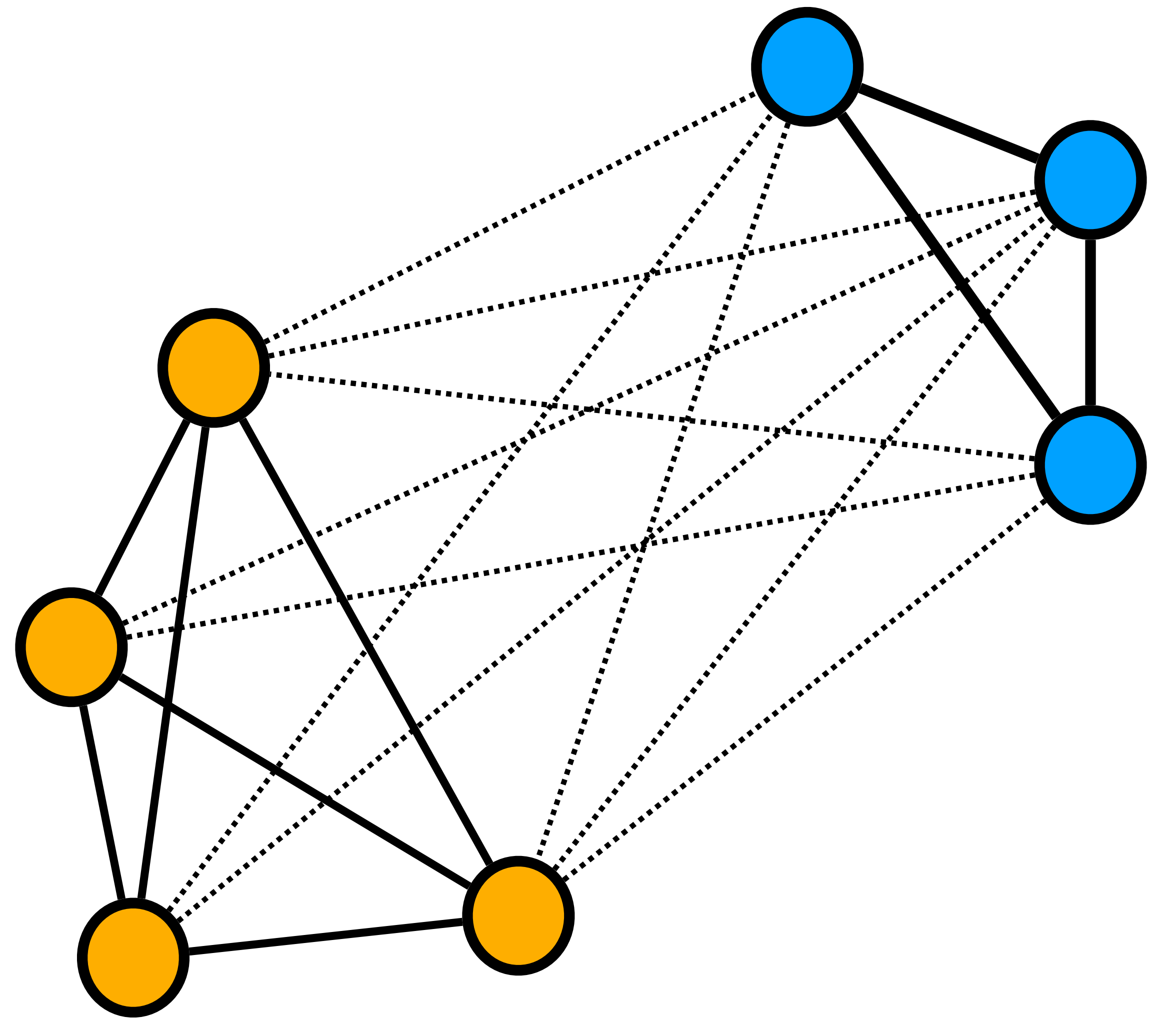
maximize $\dfrac{\text{Edge weights within a group}}{\text{Edge weights across groups}}$

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

K speakers, **N** segments

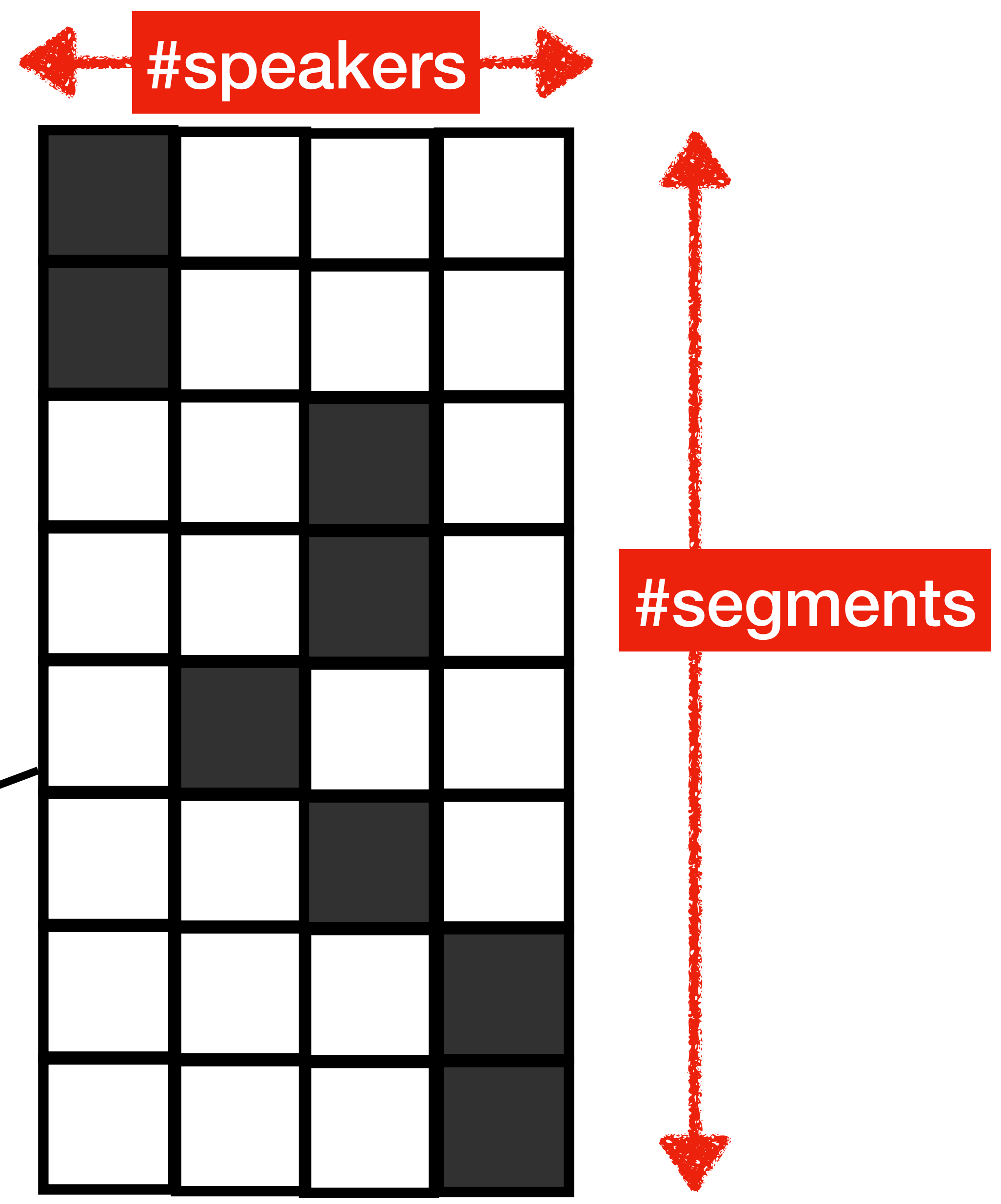
New formulation for spectral clustering

The basic clustering problem: a graph view



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

Final cluster assignment matrix



New formulation for spectral clustering

This problem is NP-hard!

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

Remove the discrete constraints to make the problem solvable

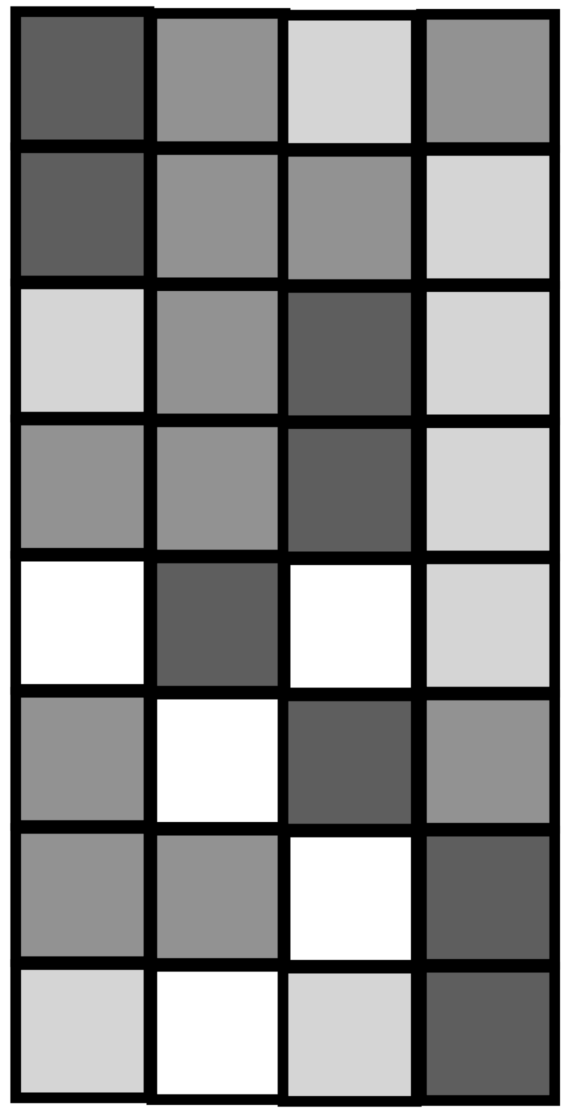
New formulation for spectral clustering

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

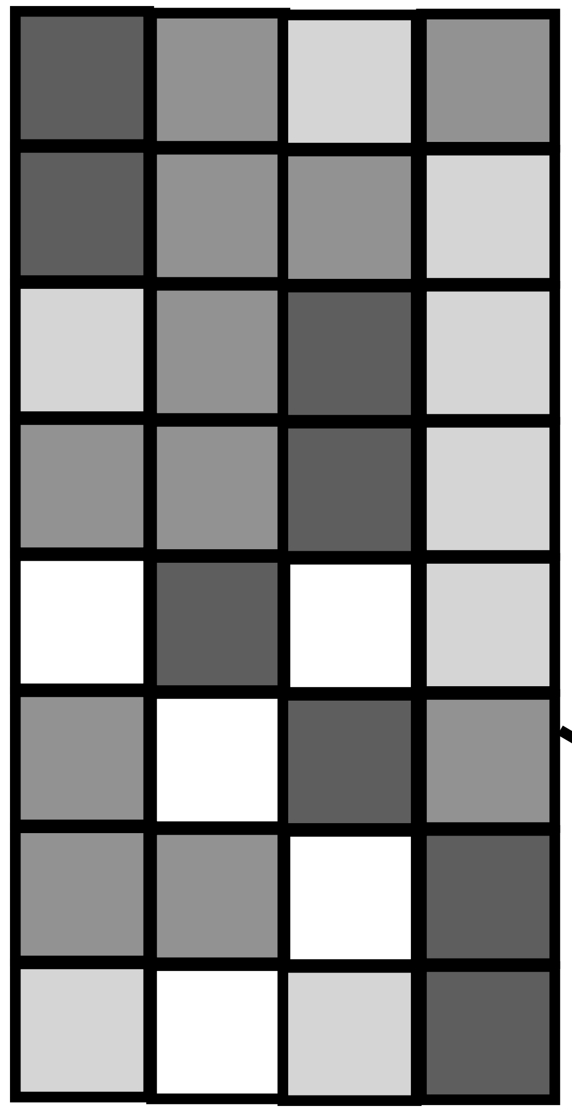


and its orthonormal transforms

Set of solutions to the **relaxed** problem

New formulation for spectral clustering

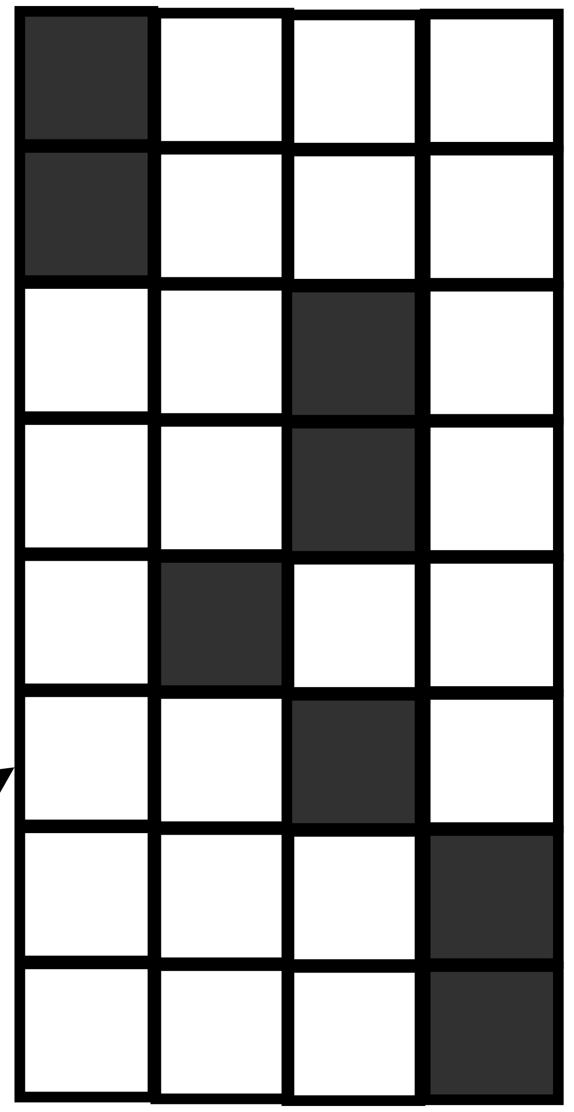
Now we need to discretize this solution!



and its orthonormal transforms

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

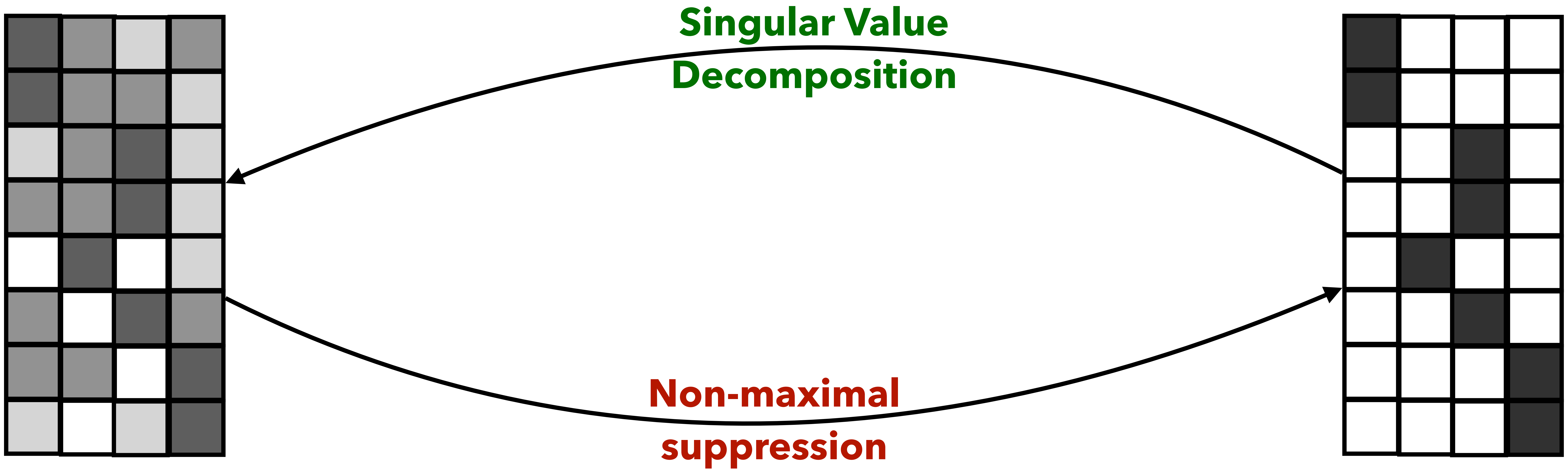
$$X\mathbf{1}_K = \mathbf{1}_N\,.$$



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!

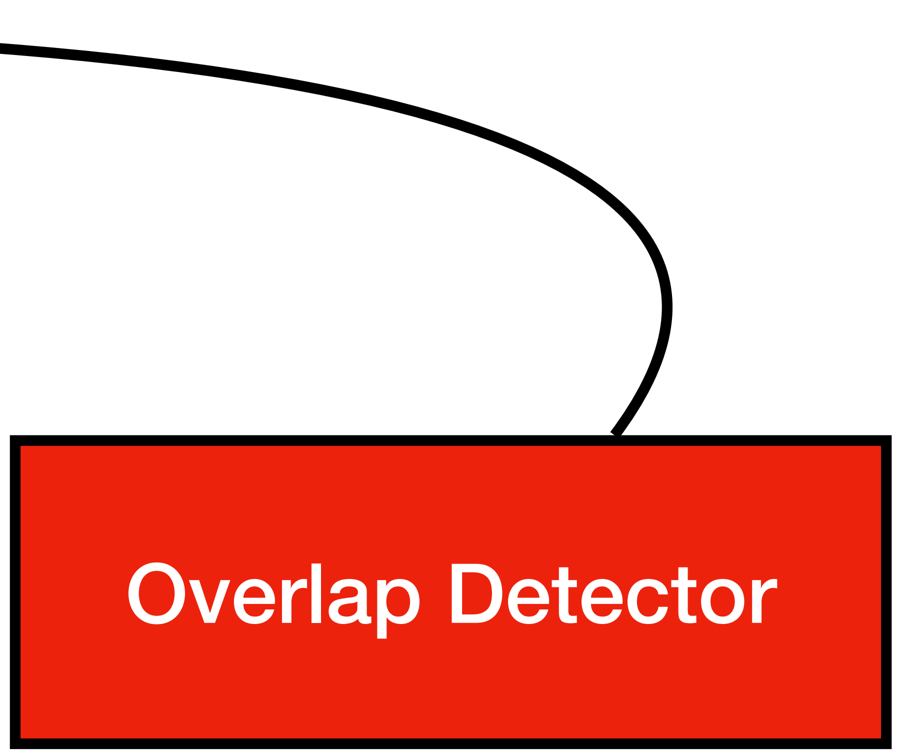


and its orthonormal transforms

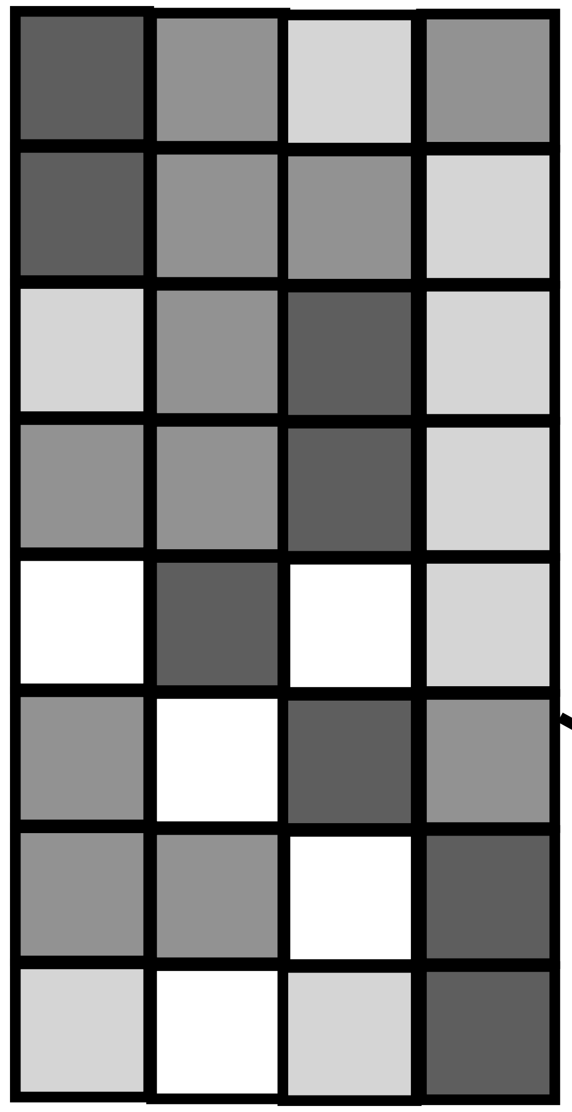
Iterate until convergence

Let us now make it overlap-aware

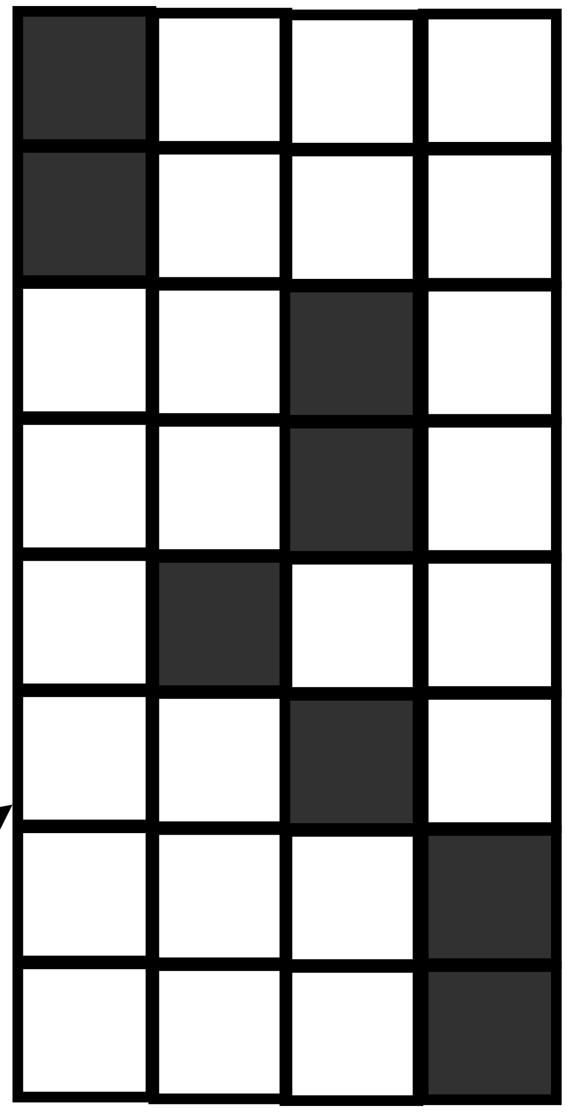
Suppose we have $\mathbf{v}_{OL}$



and its orthonormal transforms



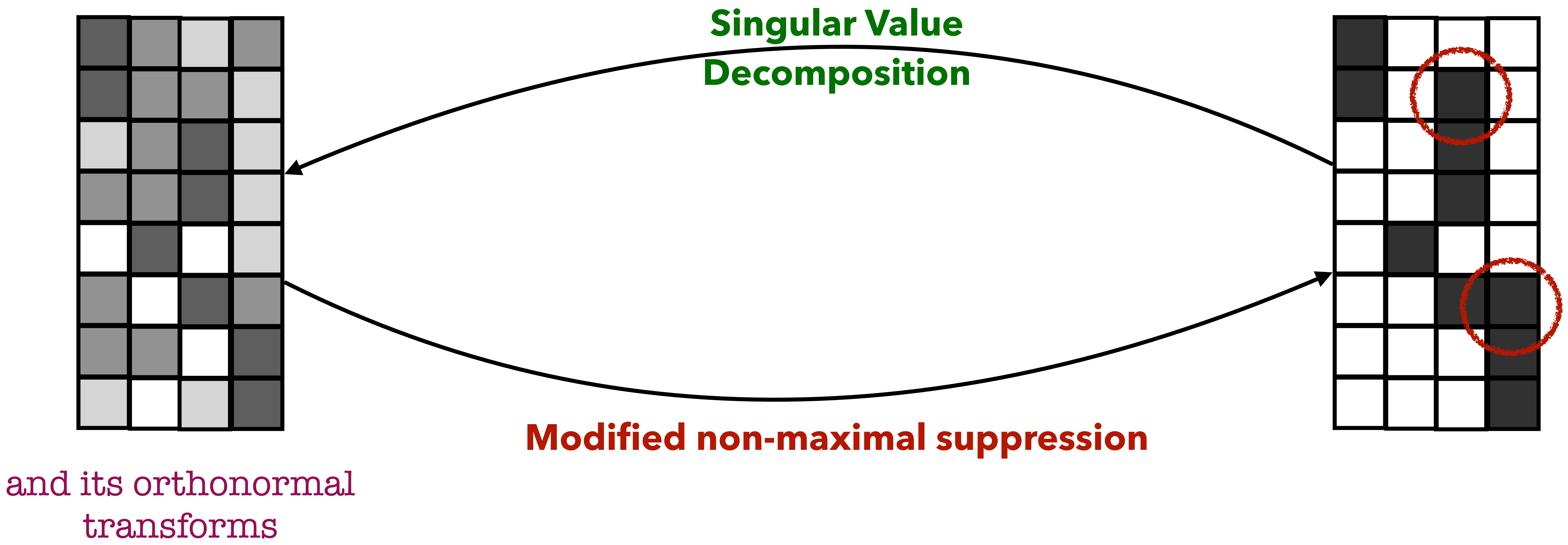
$$\text{subject to} \quad X \in \{0,1\}^{N\times K}, \quad X\mathbf{1}_K = \mathbf{1}_N + \boldsymbol{v}_{OL}.$$



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



Iterate until convergence

GPU-accelerated GSS

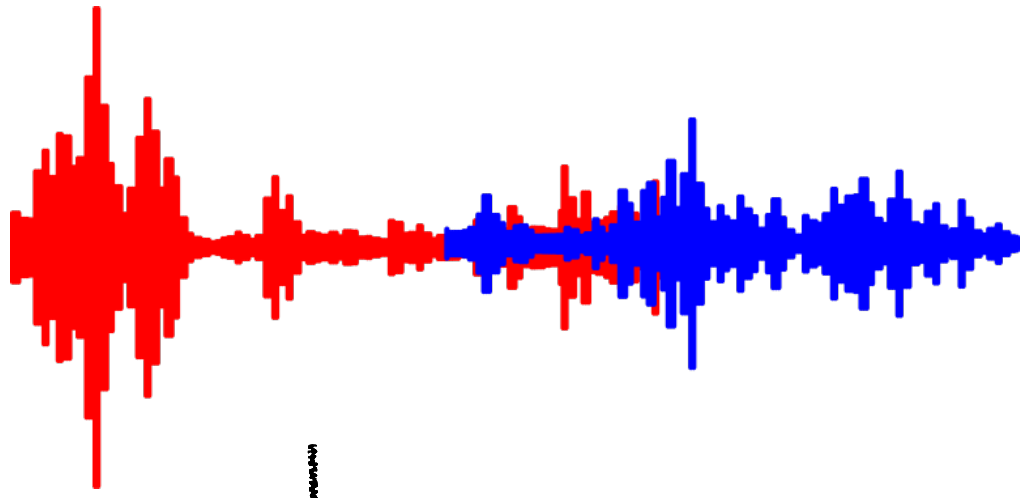
Guided source separation

Consists of 3 main steps

https://github.com/fgnt/pb_chime5

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

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

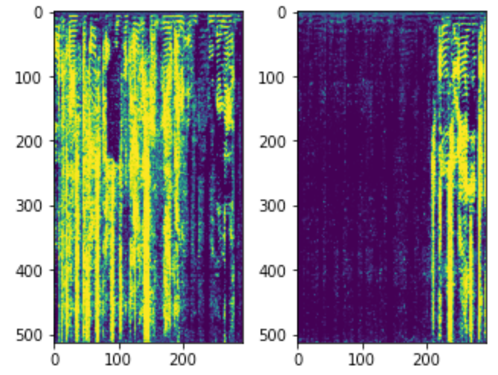


De-reverberation using Weighted Prediction Error (WPE)

Remove the late reverb

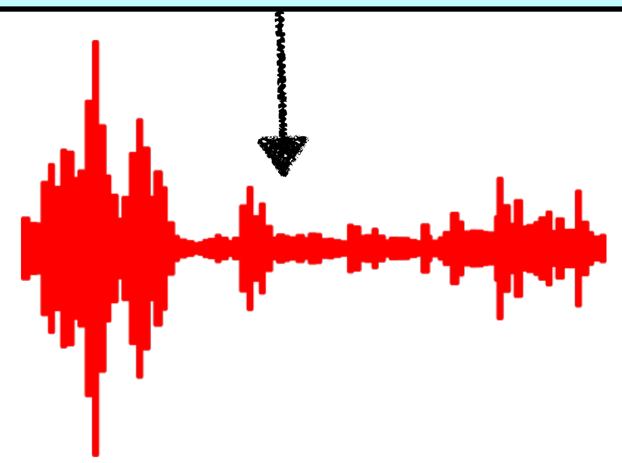
Mask estimation using mixture models

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

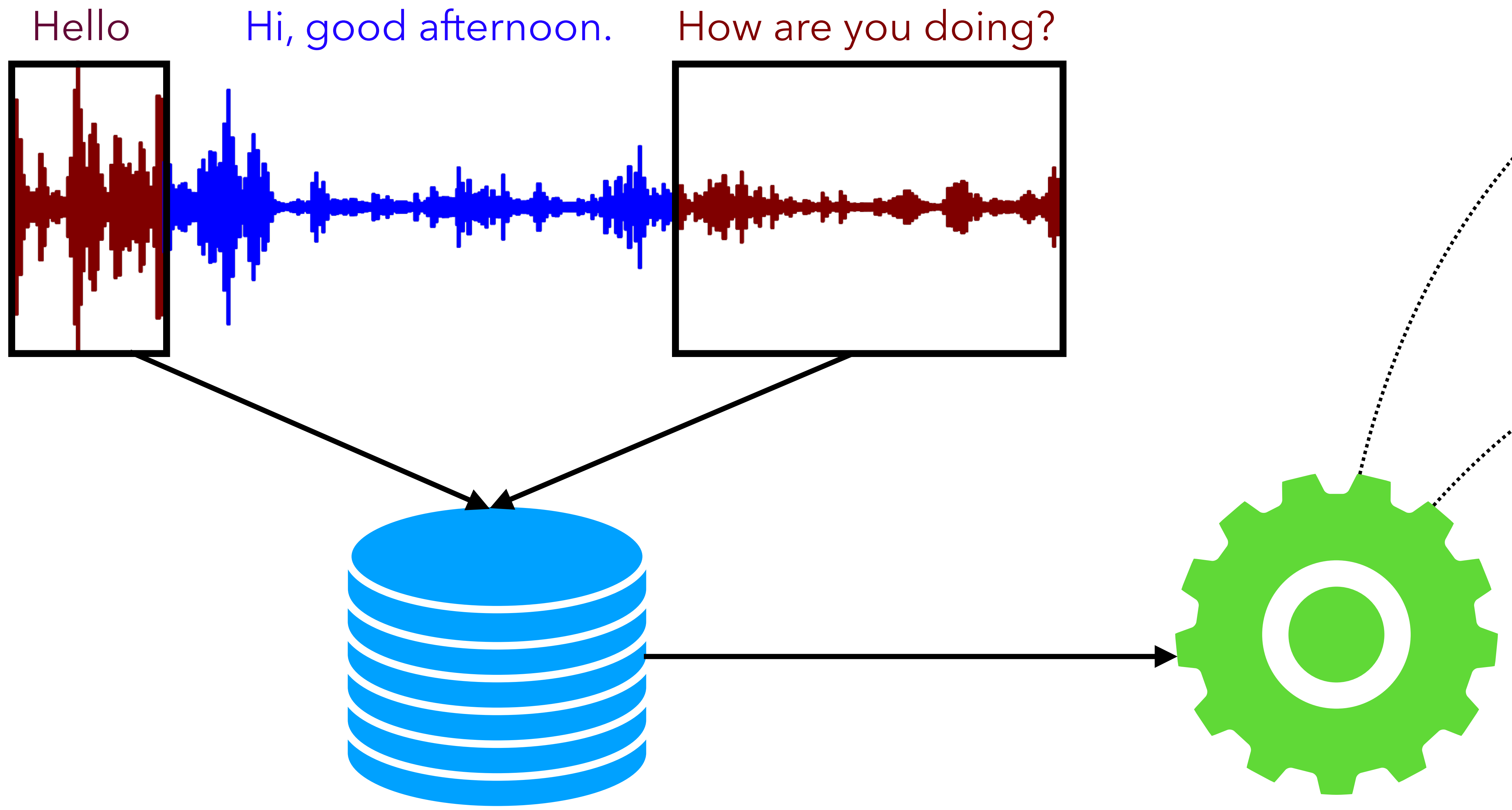
Limitations with original implementation

- 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

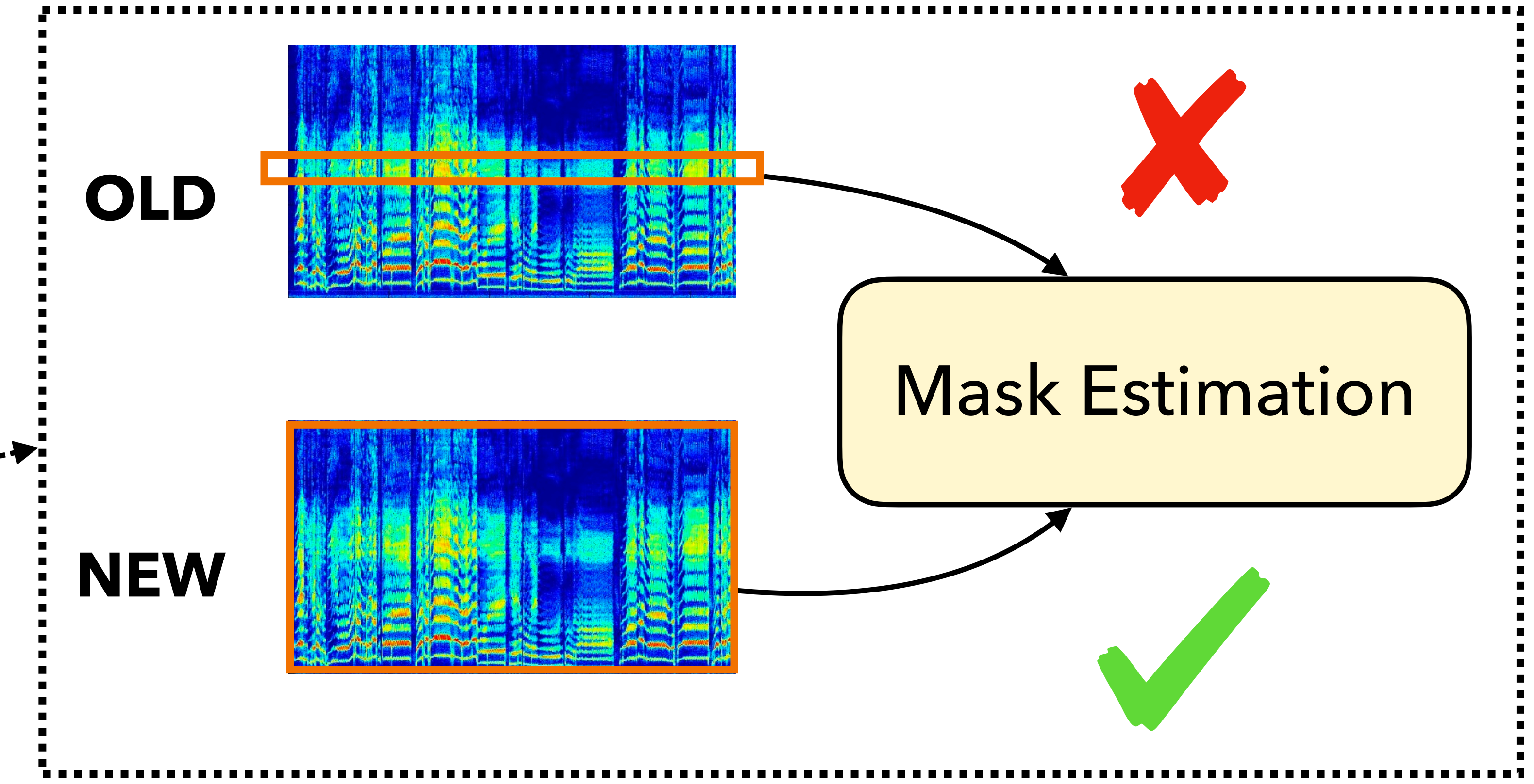
https://github.com/desh2608/gss



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

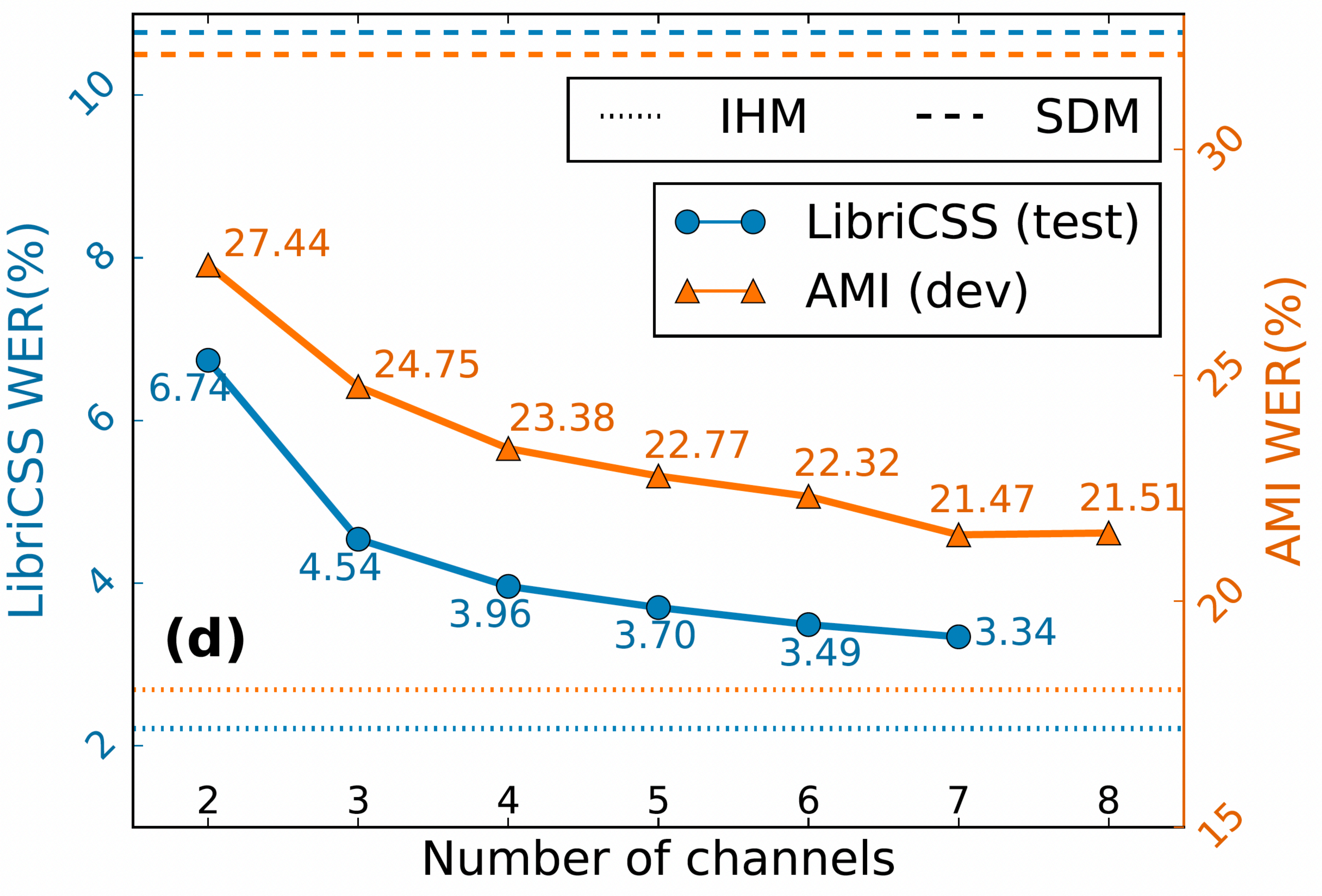
- Comparison on CHiME-6 dev set
- Old GSS: Takes **19.3** hours using 80 jobs
- New GSS: Takes **1.3** hours using 4 GPUs

CHiME-7 DASR Baseline

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

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

Speaker attribution with SURT

Speaker attribution with SURT

Some other considerations

- How to train the two branches, i.e., joint vs. sequential?
- Where to branch out of the ASR encoder?

Speaker attribution with SURT

Joint vs. sequential training

Experiments on simulated LibriSpeech mixtures

Method	ORC-WER	WDER	cpWER
Sequential	8.5	**4.0**	**15.0**
Joint	**8.4**	4.5	**15.0**
Sequential + joint	9.2	4.3	15.3

Speaker attribution with SURT

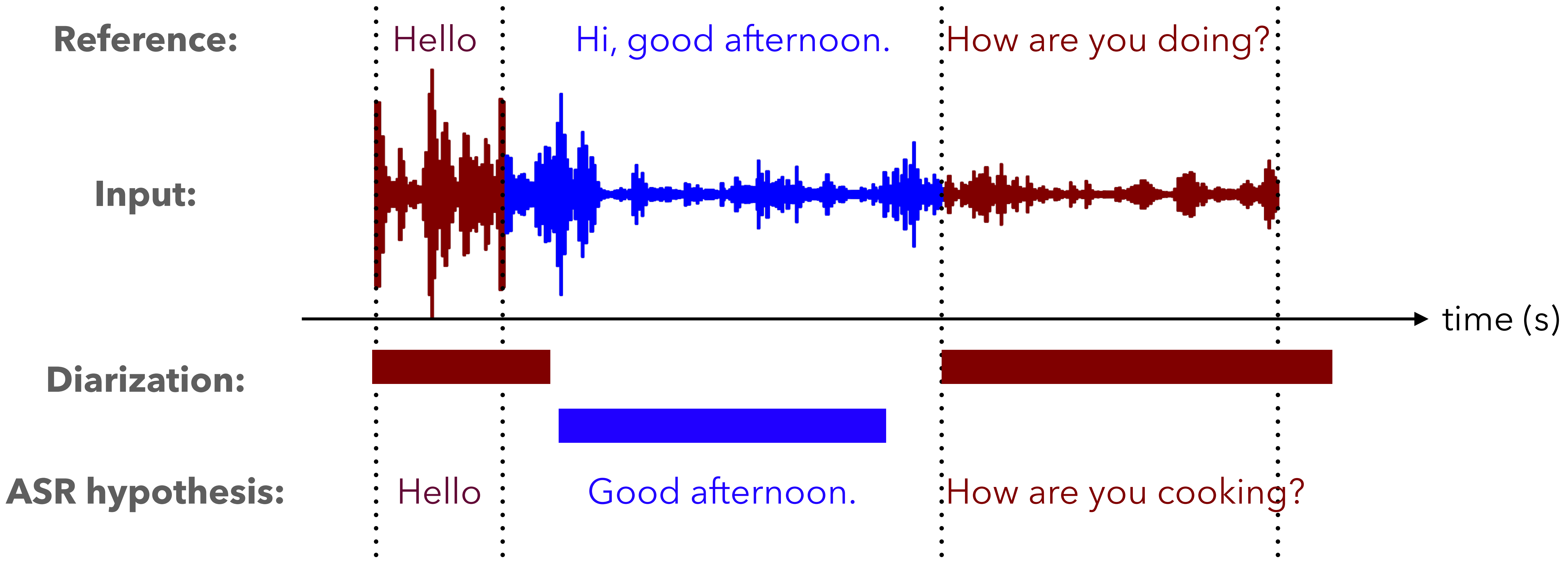
Where to branch out of the main encoder?

Experiments on simulated LibriSpeech mixtures

Main Encoder Block	WDER	cpWER
Block 0 (after embedding layer)	5.4	16.7
✓ Block 1	**4.0**	**15.0**
Block 2	6.7	19.6
Block 3	8.4	23.4

Problem Statement

Evaluation Metrics



Diarization Error Rate (DER)

$$\frac{\text{Missed speech + False alarms + Speaker confusion}}{\text{Total speaking time}}$$

Concatenated minimum permutation Word Error Rate (cpWER)

Concatenated reference:	Hello How are you doing?	Hi, good afternoon.
Concatenated hypothesis:	Hello How are you cooking?	Good afternoon.

Compute average WER for all permutations of speakers and return minimum