Speaker diarization

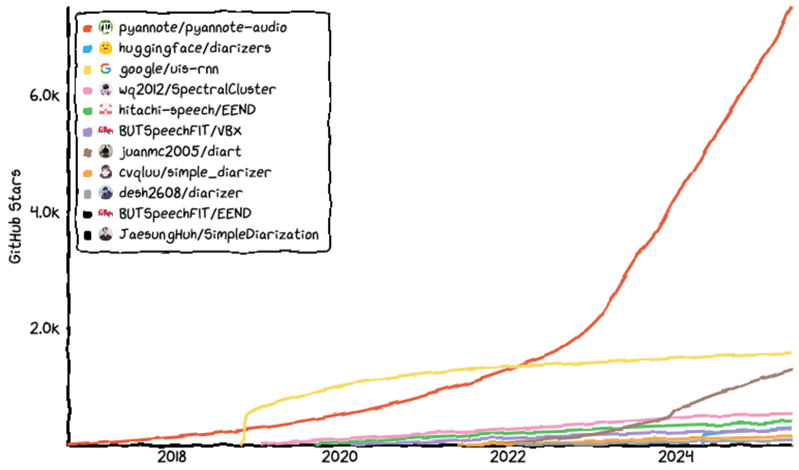
a love story

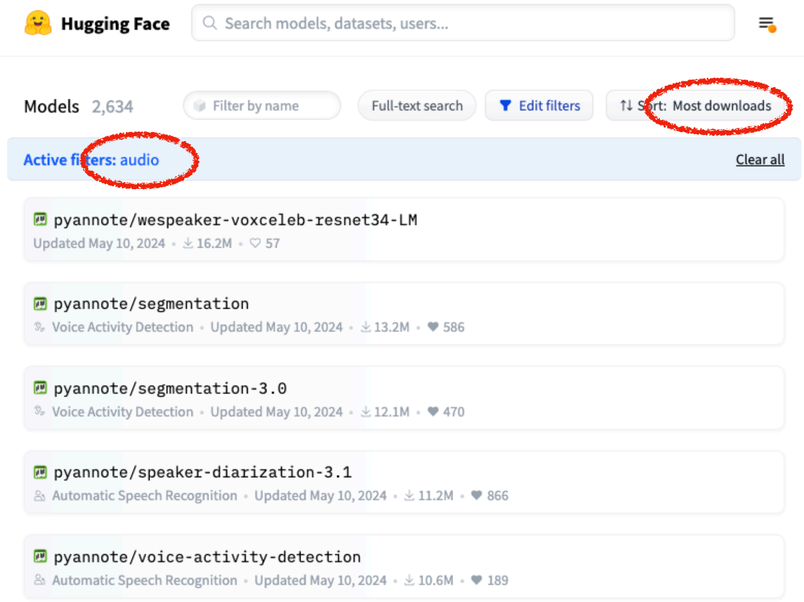
Hervé Bredin pyannoteAI

Hervé Bredin

the "pyannote" guy

- Tenured **CNRS** researcher since 2008
- Started working on **speaker diarization** back in 2012 (and never looked back)
- Started open-sourcing **pyannote** toolkit in 2015 (and don’t regret it)
- Co-founded **pyannoteAI** in 2024 (on leave from CNRS since 2025)



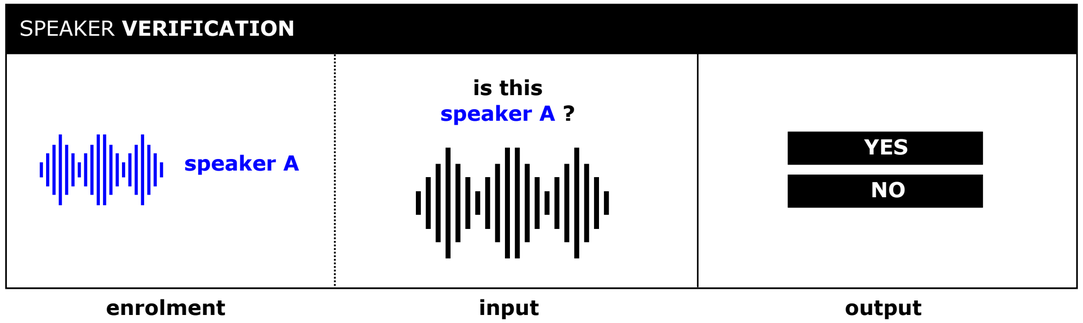


pyannoteAI

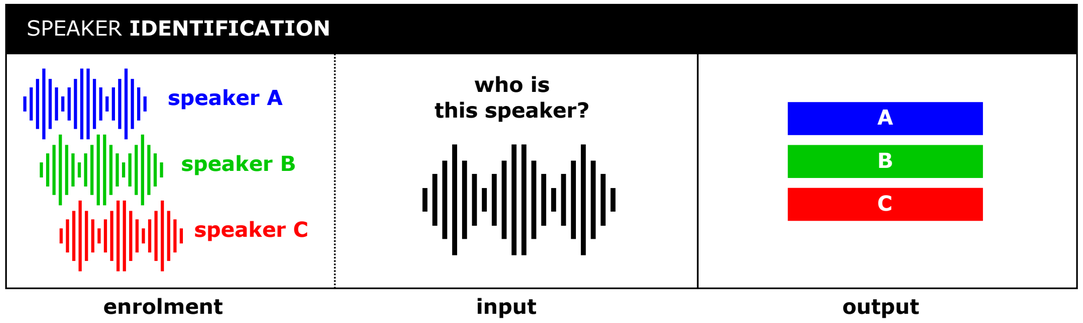
What is speaker diarization?

pyannoteAI

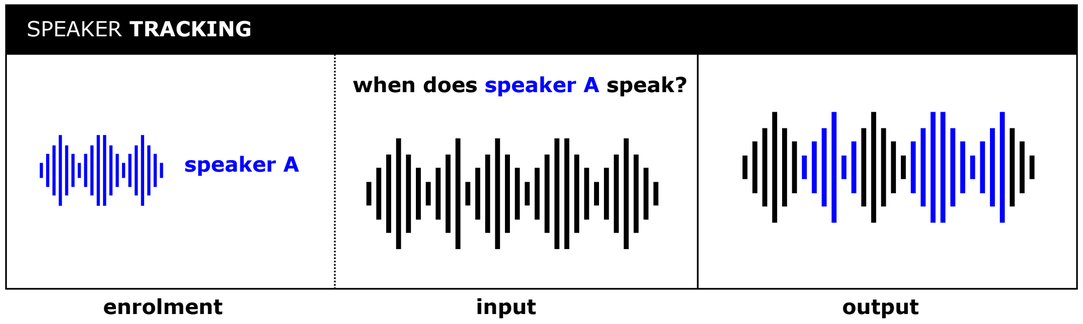
Speaker recognition



supervised, utterance-wise
binary classification



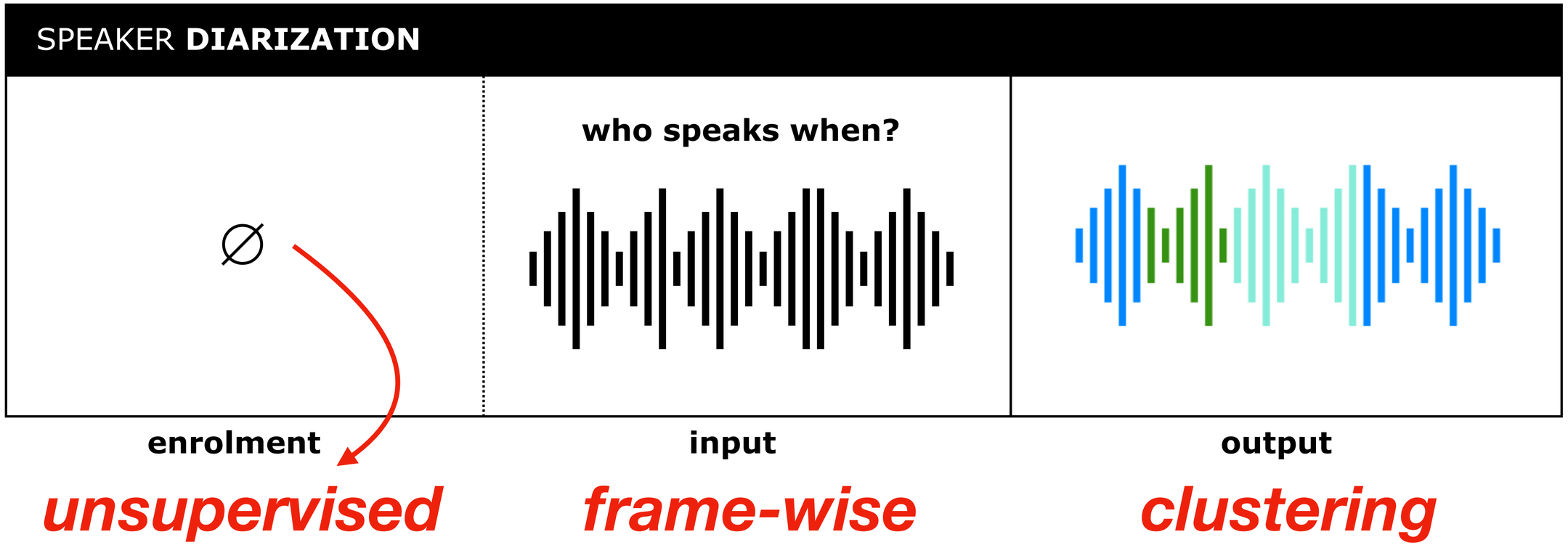
supervised, utterance-wise
multi-class classification



supervised, frame-wise
binary classification

Speaker diarization

"*who speaks when*"



pyannoteAI

Speaker diarization(s)

- **Batch** speaker diarization
  assumes the whole conversation is available at once
- **Streaming** speaker diarization
  processes the conversation in "real-time" as it unfolds
- **Longitudinal** speaker diarization
  tracks speakers across multiple conversations

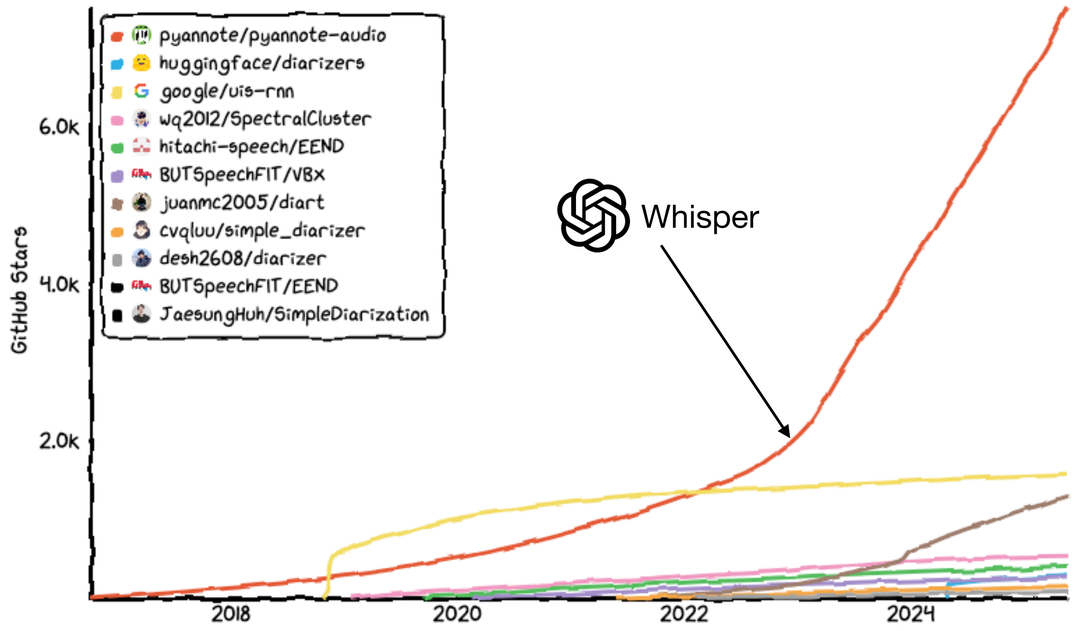
pyannoteAI

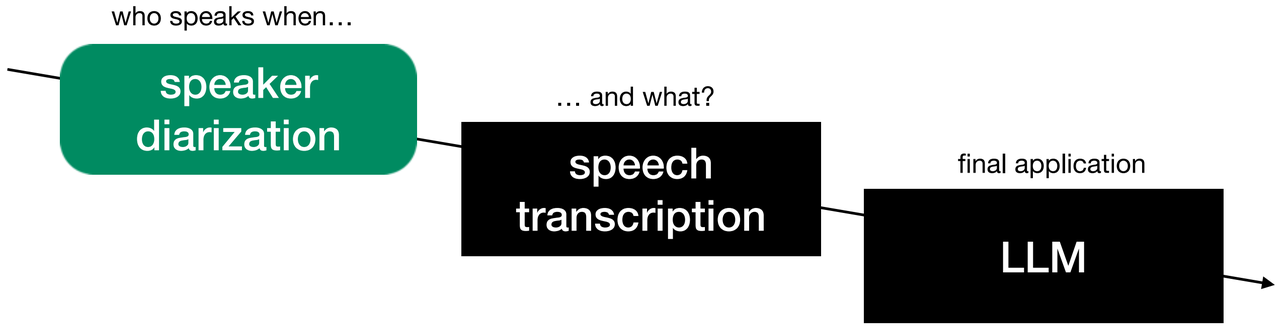
What for?

"*a means to an end*"

Speaker diarization is an **enabling technology** for **downstream applications.**

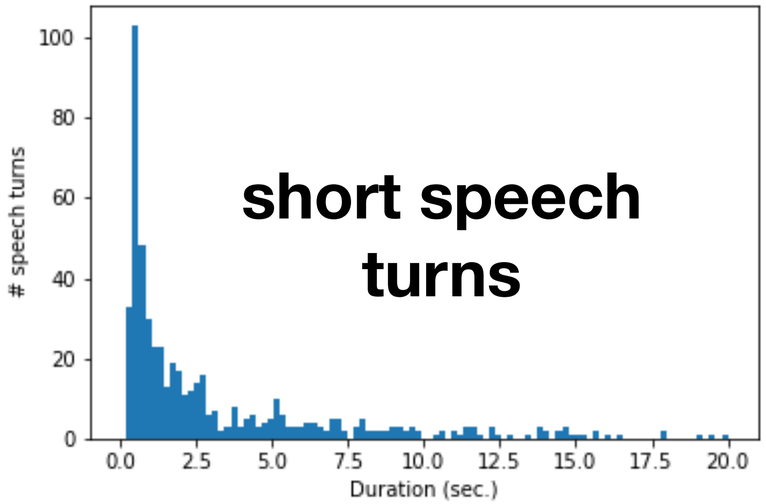
- Speaker-attributed transcription
  meeting, call center, healthcare, court





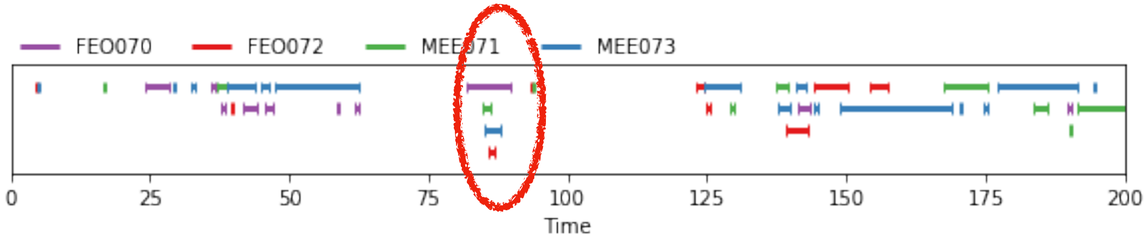
- Voice assistant training — kyutai Moshi
- Text-to-speech training — Meta Audiobox
- Video translation with voice cloning
- Life logging (wearables)

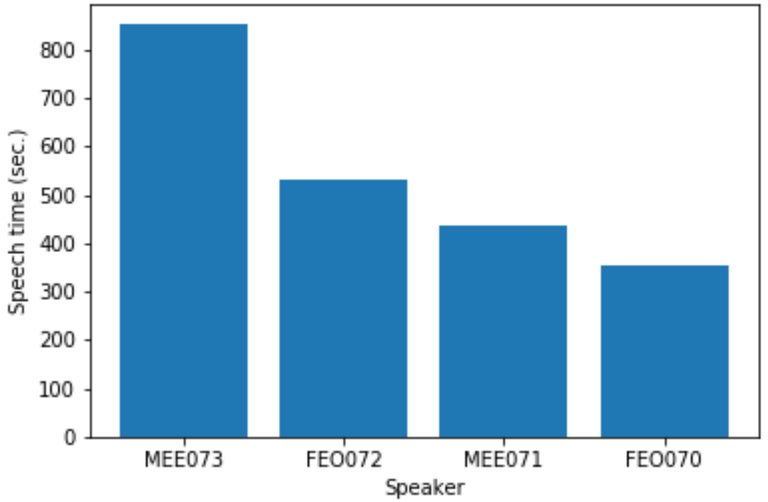
What makes it difficult?



short speech turns

overlapping speech (up to 40%)





unbalanced speaker time

unknown number of speakers

pyannoteAI

How does it work?

pyannoteAI

"Historical" approach

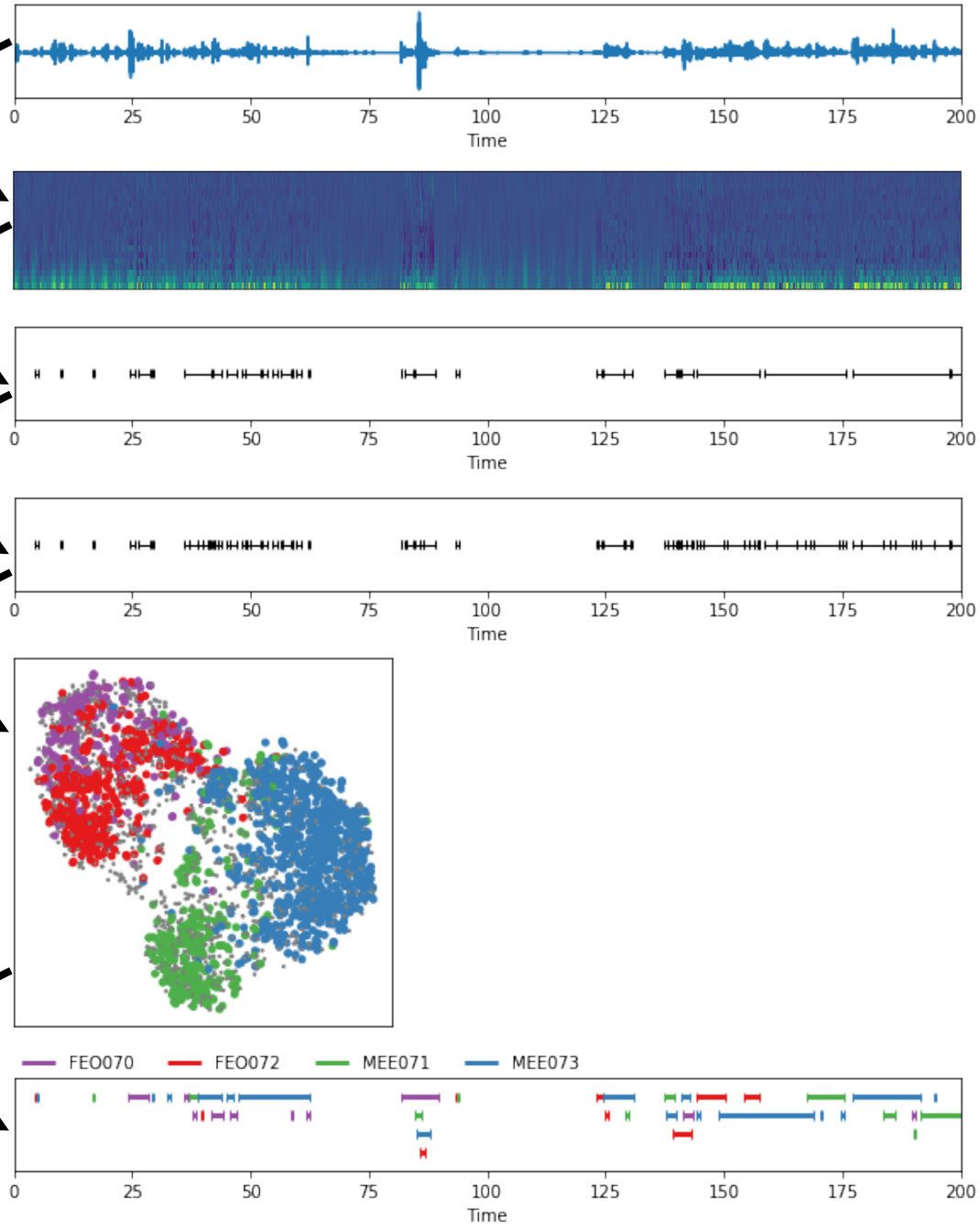
feature extraction

speech activity detection

speaker change detection

speech turn representation

speech turn clustering

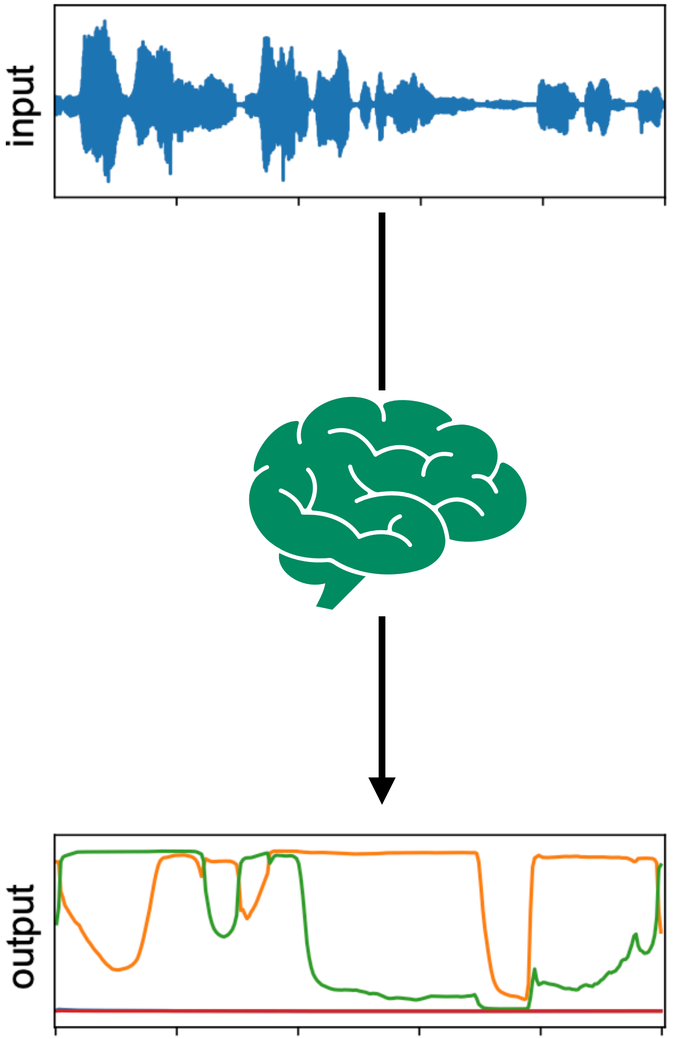


pyannoteAI

End-to-End Neural Diarization

- **Sequence-to-sequence**
  input: audio
  output: speaker probabilities

- **Multi-label classification**
  fixed number of speakers
  overlap-aware by design



pyannoteAI

Pros and cons

Multi-Stage Diarization

❌ propagation of errors

❌ hacky handling of overlapping speech

✅ (relatively) easy to train

✅ good at handling variable number of speakers

✅ scale (almost) linearly with file duration

End-to-End Neural Diarization

✅ one single step

✅ overlapping speech as first class citizen

❌ difficult to train

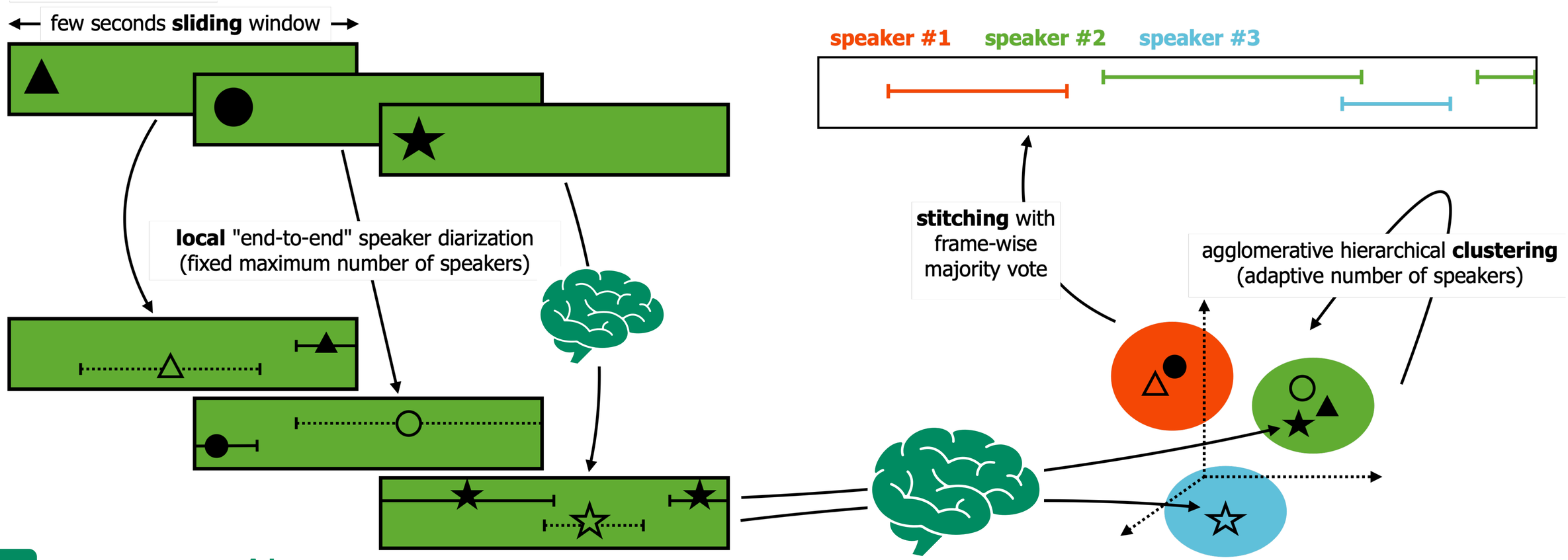
❌ not so good at generalizing to previously unseen number of speakers

❌ does not scale with file duration

pyannoteAI

EEND-VC

the best of both worlds



Speaker diarization

a ~~love~~ loss story

Hervé Bredin

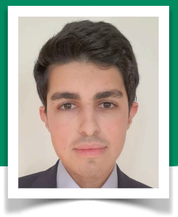
pyannoteAI

Powerset multi-class cross entropy loss for neural speaker diarization

Alexis Plaquet & Hervé Bredin
Interspeech 2023

Multi-latency look-ahead for streaming speaker diarization

Bilal Rahou & Hervé Bredin
Interspeech 2024



PixIT: Joint Training of Speaker Diarization and Speech Separation from Real-world Multi-speaker Recordings

Joonas Kalda, Clément Pagés, Ricard Marxer, Tanel Alumäe & Hervé Bredin
Speaker Odyssey 2024

Speaker diarization

a ~~love~~ loss story

Hervé Bredin

pyannoteAI

Powerset multi-class cross entropy loss for neural speaker diarization

Alexis Plaquet & Hervé Bredin
Interspeech 2023

Multi-latency look-ahead for streaming speaker diarization

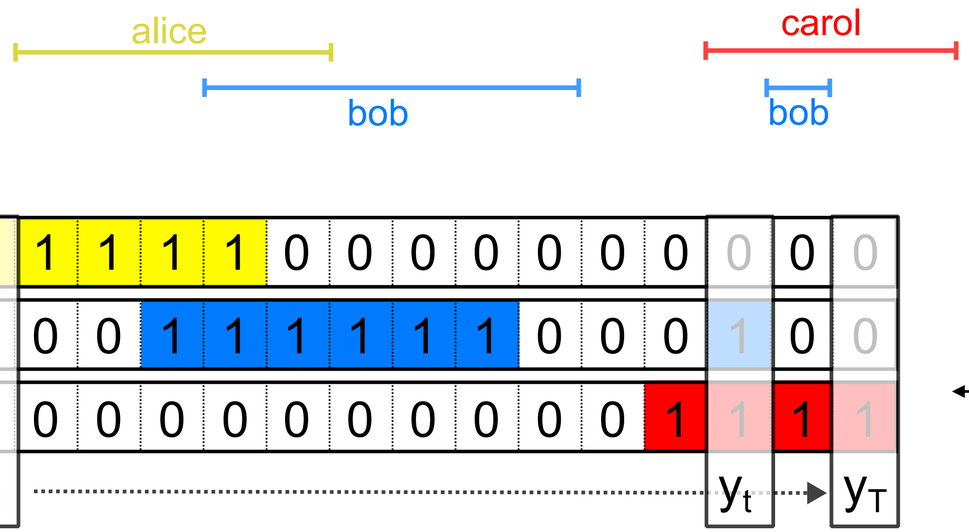
Bilal Rahou & Hervé Bredin
Interspeech 2024

PixIT: Joint Training of Speaker Diarization and Speech Separation from Real-world Multi-speaker Recordings

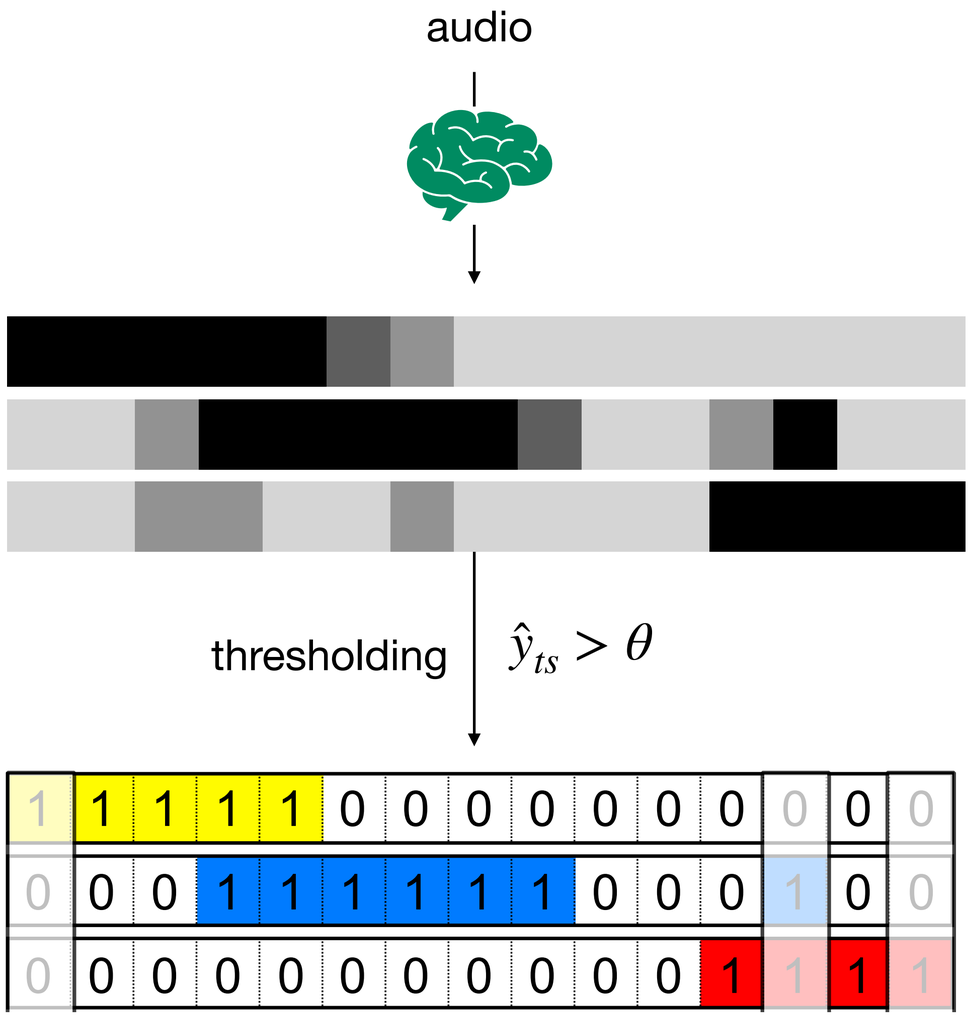
Joonas Kalda, Clément Pagés, Ricard Marxer, Tanel Alumäe & Hervé Bredin
Speaker Odyssey 2024

End-to-End Neural Diarization

Frame-wise multi-label classification



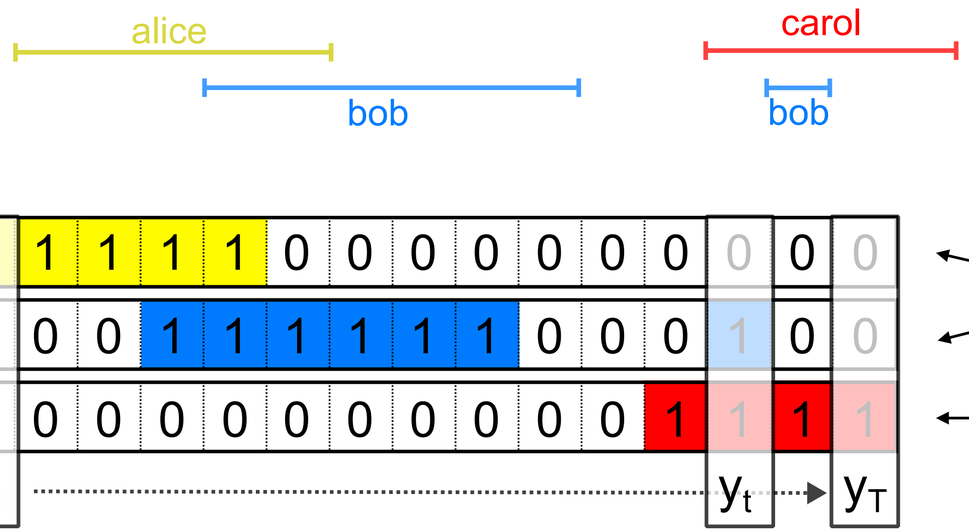
$$\mathscr{L}_{BCE}$$

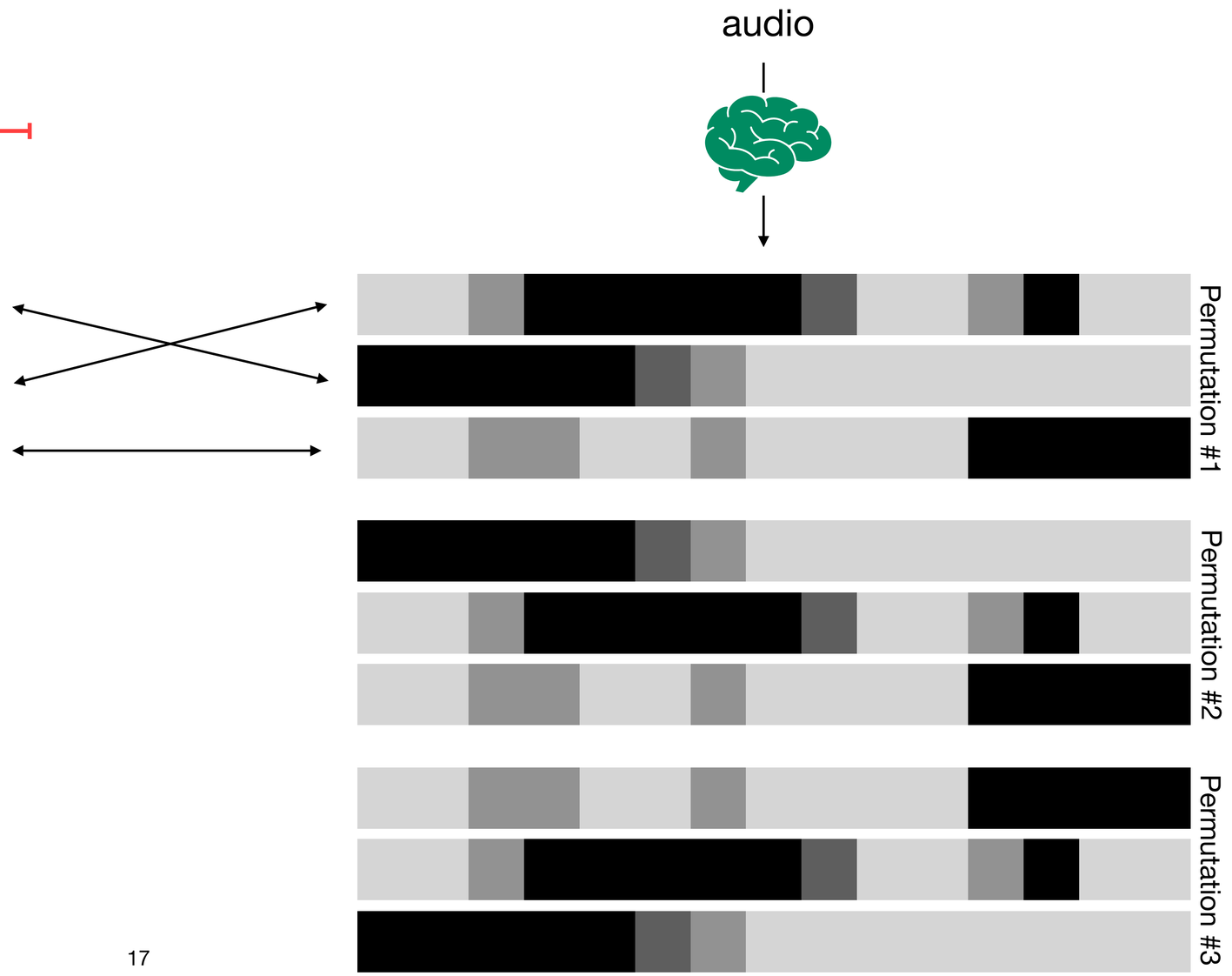


pyannoteAI

End-to-End Neural Diarization

Invariance to permutation



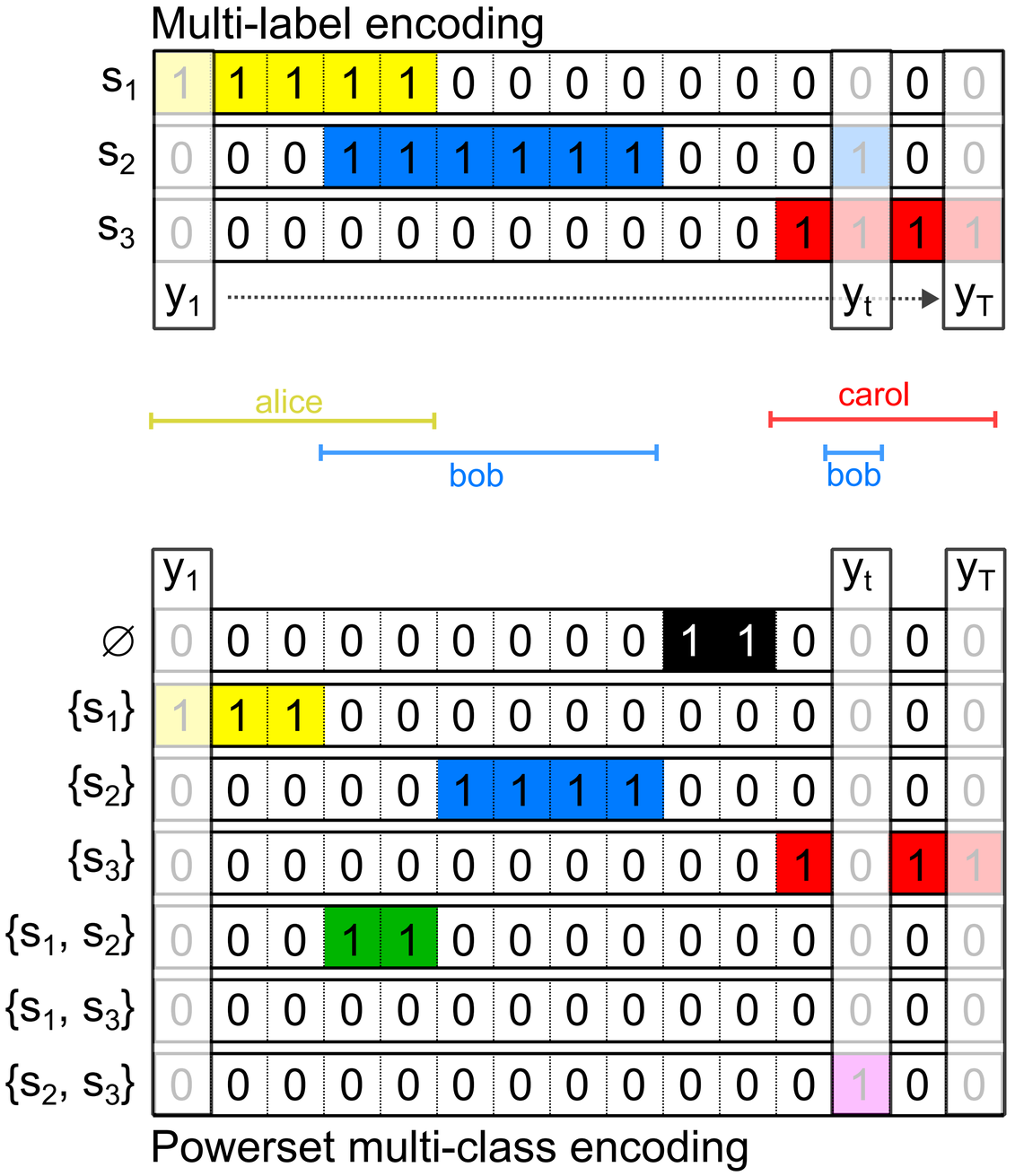


$$\mathscr{L} = \min_{p \in \mathscr{P}} \mathscr{L}_{BCE}\left(y, p(\hat{y})\right)$$

pyannoteAI

From multi-label to powerset encoding

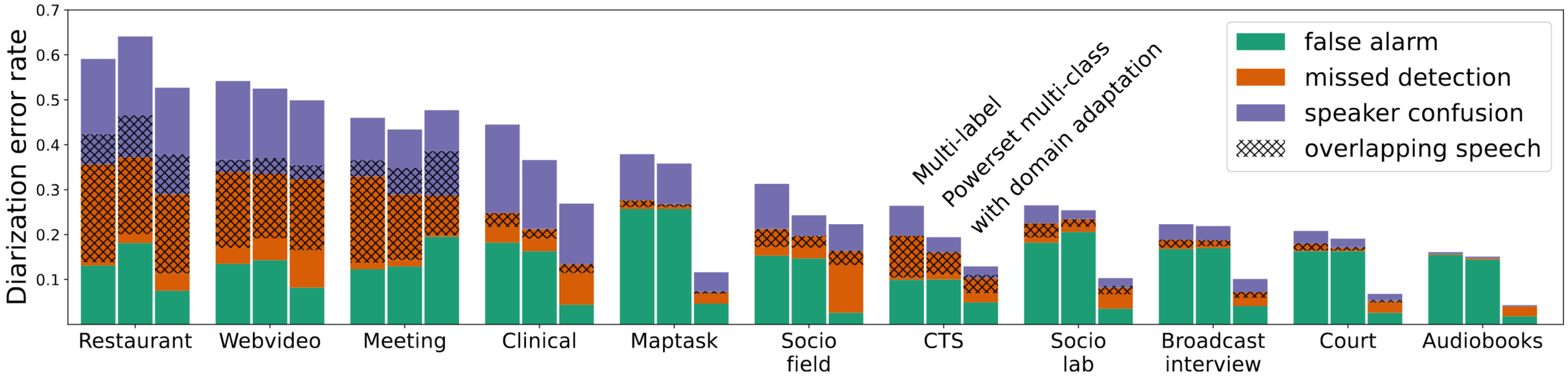
- **dedicated multi-speakers classes**
  number of classes growing fast
- **"regular" multi-class classification**
  mutually exclusive classes
  cross-entropy loss (easier to train)
- **thresholding replaced by argmax**
  one less hyper-parameter to tune
- **dedicated non-speech class**
  to control VAD aggressiveness



pyannoteAI

Results

DIHARD 3 benchmark (11 application domains)



$$\text{DER} = \frac{\text{false alarm} + \text{missed detection} + \text{speaker confusion}}{\text{total speech}}$$

pyannoteAI

Speaker diarization

a ~~love~~ loss story

Hervé Bredin

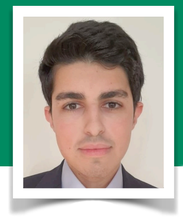
pyannoteAI

Powerset multi-class cross entropy loss for neural speaker diarization

Alexis Plaquet & Hervé Bredin
Interspeech 2023

Multi-latency look-ahead for streaming speaker diarization

Bilal Rahou & Hervé Bredin
Interspeech 2024



PixIT: Joint Training of Speaker Diarization and Speech Separation from Real-world Multi-speaker Recordings

Joonas Kalda, Clément Pagés, Ricard Marxer, Tanel Alumäe & Hervé Bredin
Speaker Odyssey 2024

Streaming speaker segmentation

Causal architecture?

Offline DER = 17.3%

mono-directional

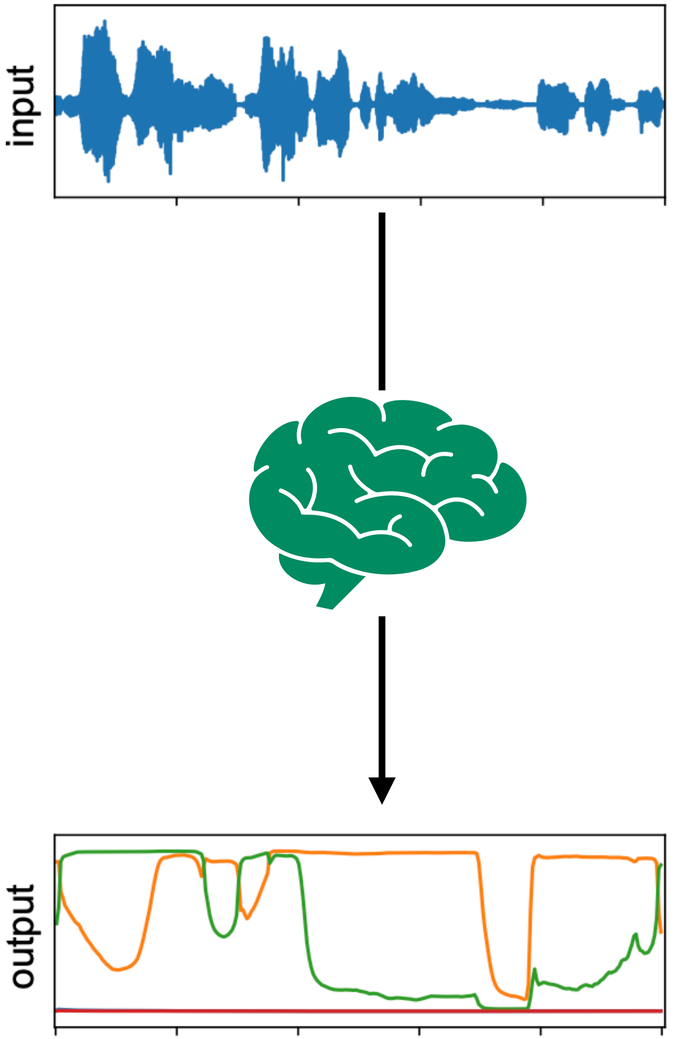
- **Sequence modeling**
  bi-directional LSTM ❌ → 20.3%

no normalization

- **Feature extraction**
  Sincnet trainable filterbank
  (with instance normalization) ❌ → 21.1%

- **Final classification**
  frame by frame ✅

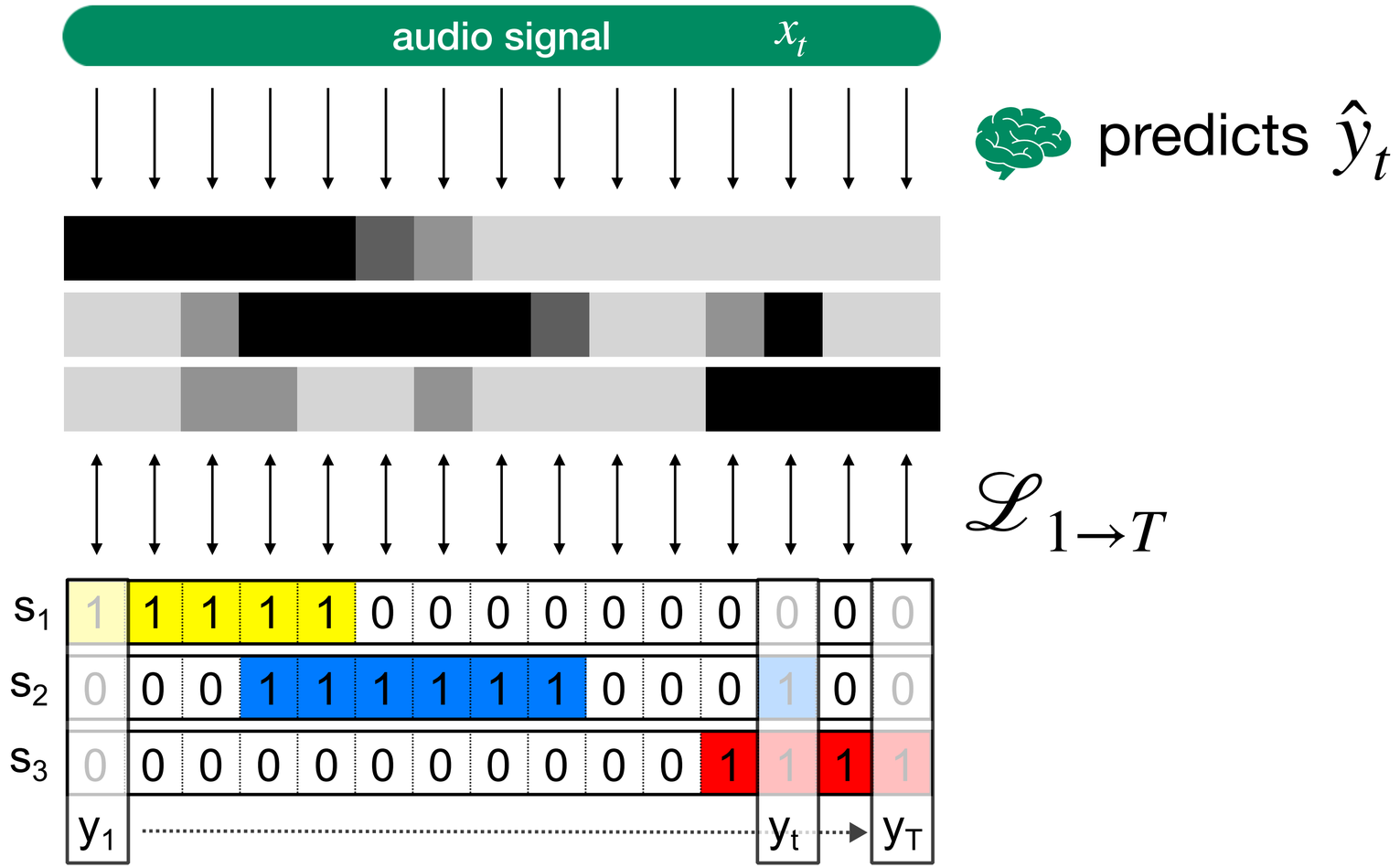
how to go back to 17.3% ?



pyannoteAI

Streaming speaker segmentation

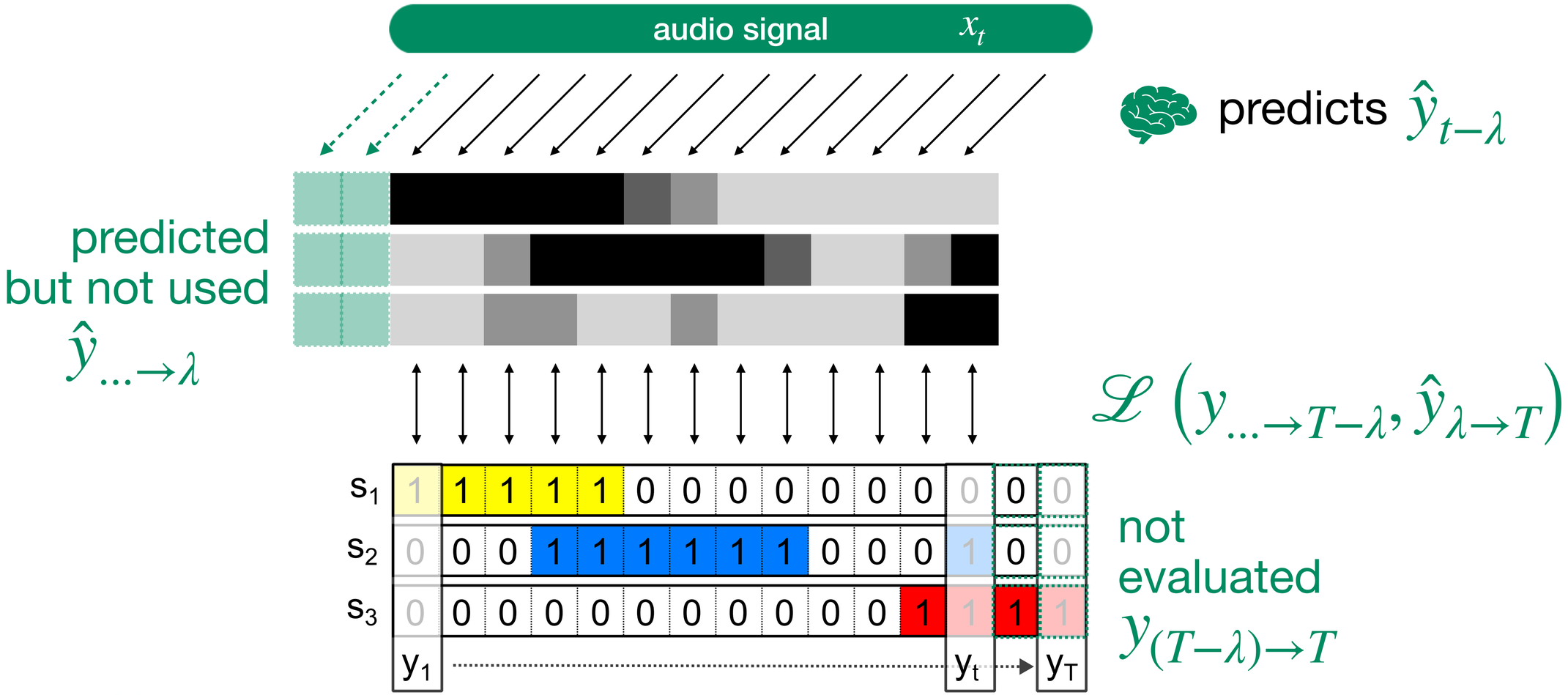
Look-ahead loss



pyannoteAI

Streaming speaker segmentation

Look-ahead loss

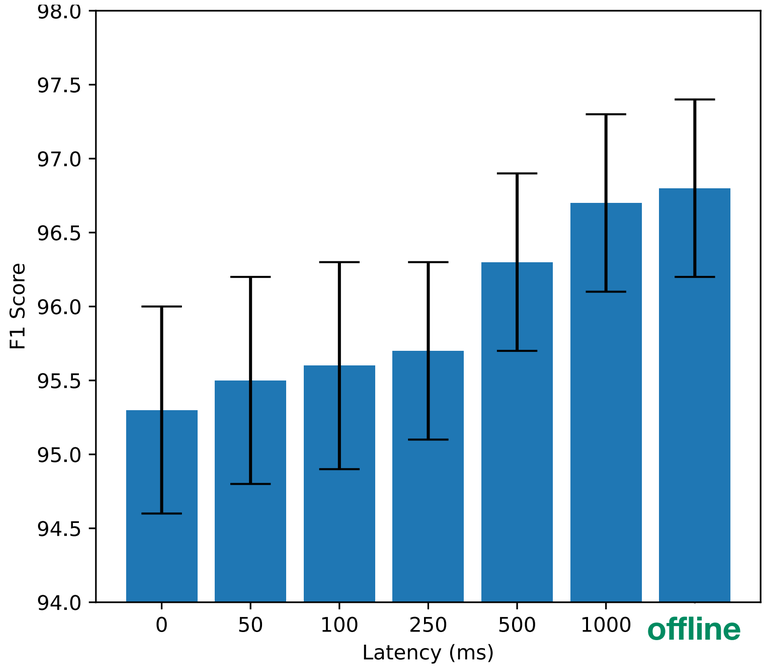


pyannoteAI

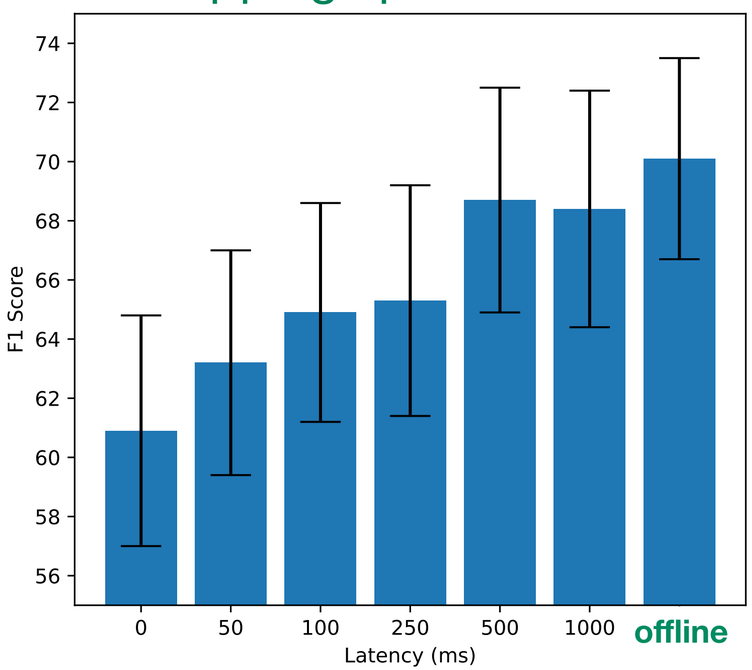
Streaming speaker segmentation

Latency/accuracy tradeoff

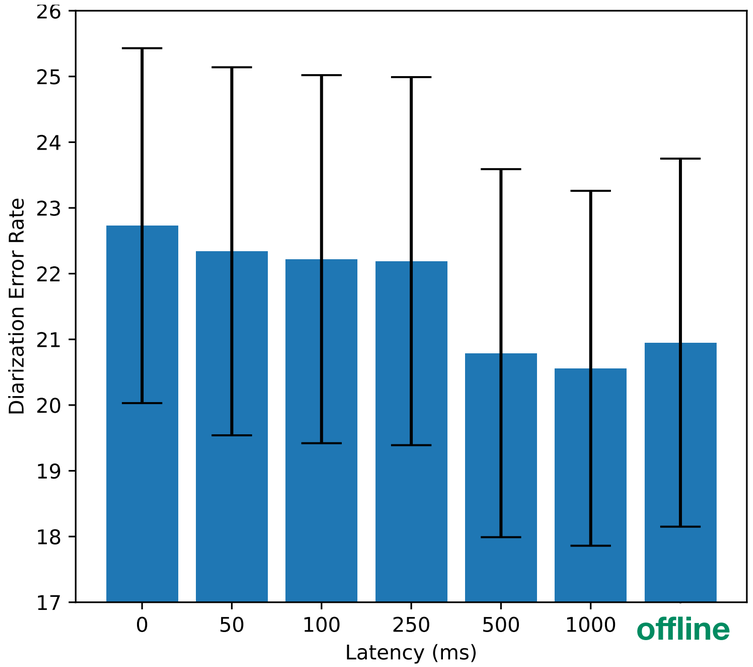
voice activity detection



overlapping speech detection



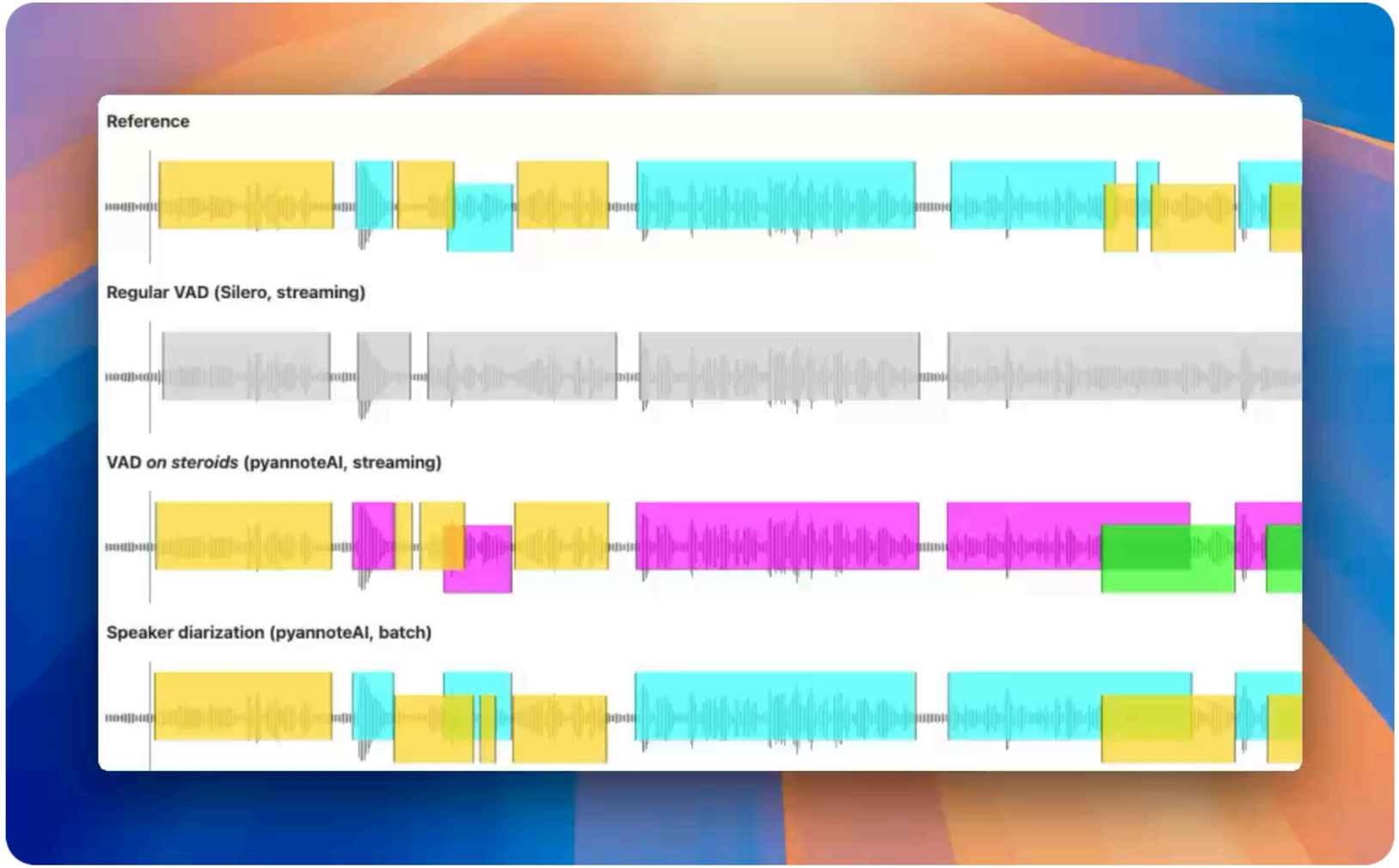
speaker change detection



pyannoteAI

Streaming speaker segmentation

VAD on steroids



pyannoteAI

Speaker diarization

a ~~love~~ loss story

Hervé Bredin pyannoteAI

some figures courtesy of Joonas Kalda

Powerset multi-class cross entropy loss for neural speaker diarization

Alexis Plaquet & Hervé Bredin
Interspeech 2023

Multi-latency look-ahead for streaming speaker diarization

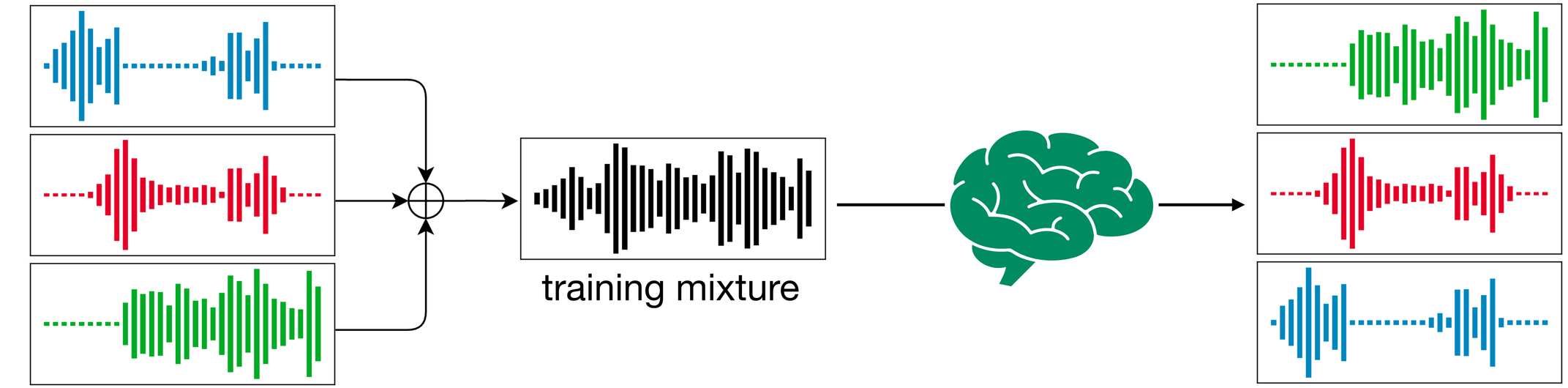
Bilal Rahou & Hervé Bredin
Interspeech 2024

PixIT: Joint Training of Speaker Diarization and Speech Separation from Real-world Multi-speaker Recordings

Joonas Kalda, Clément Pagés, Ricard Marxer, Tanel Alumäe & Hervé Bredin
Speaker Odyssey 2024

Speech separation

Supervised permutation-invariant training (PIT)



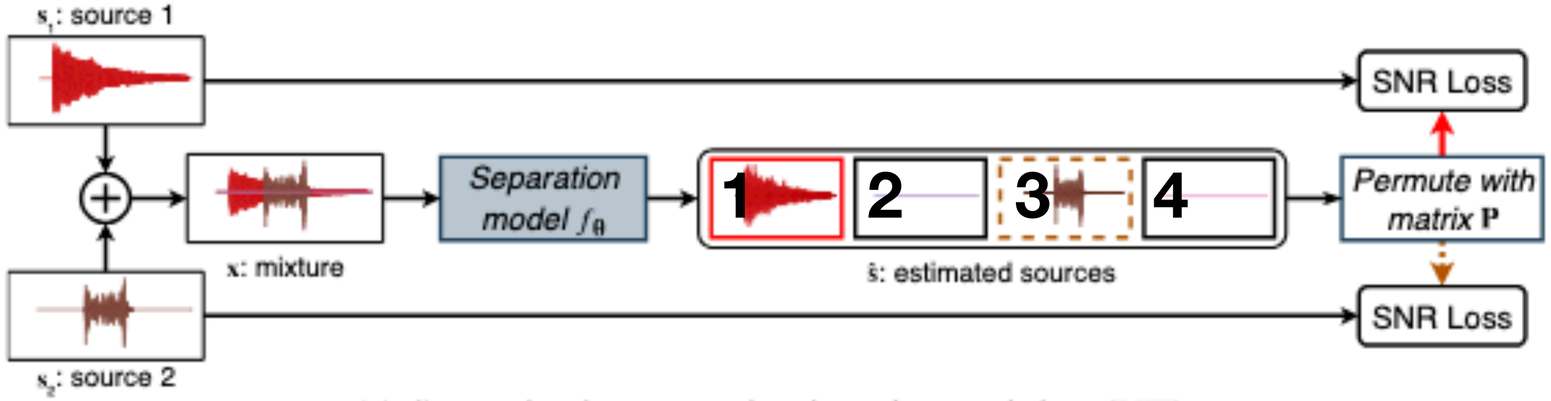
$$\min_{p \in \mathcal{P}} \mathcal{L}\left(s, p(\hat{s})\right)$$

pyannoteAI

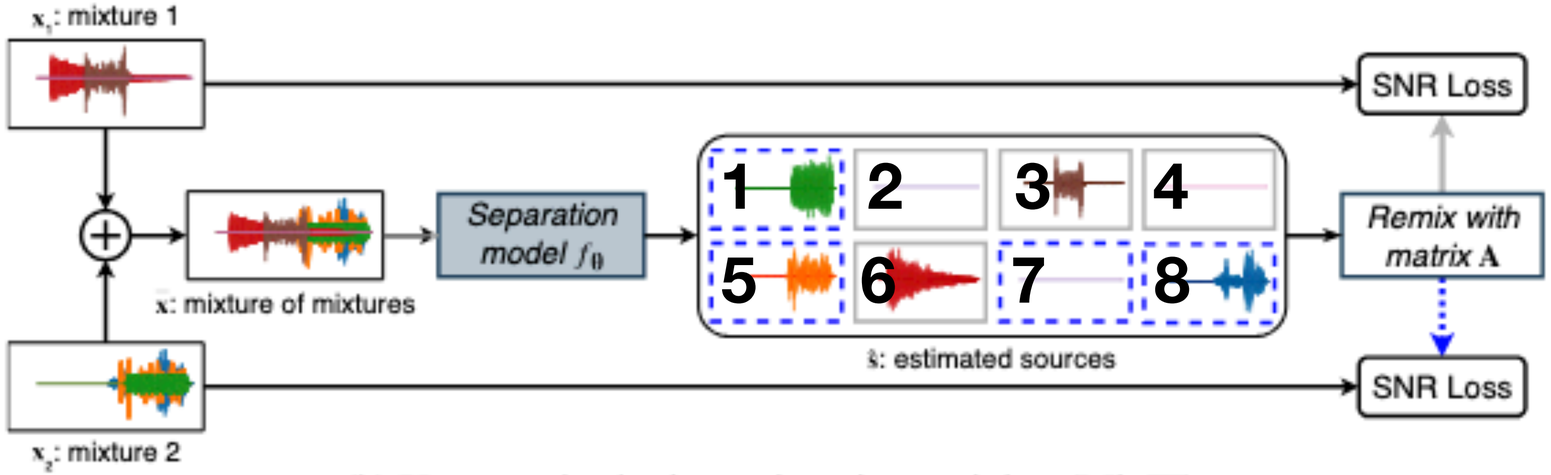
"*Unsupervised Sound Separation using Mixture Invariant Training*"
Wisdom et al. NeurIPS 2020

Speech separation

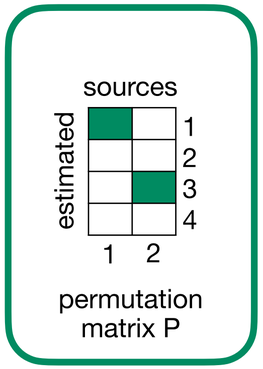
Unsupervised mixture-invariant training (MixIT)

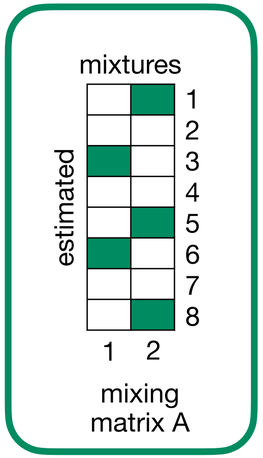


(a) Supervised permutation invariant training (PIT).



(b) Unsupervised mixture invariant training (MixIT).

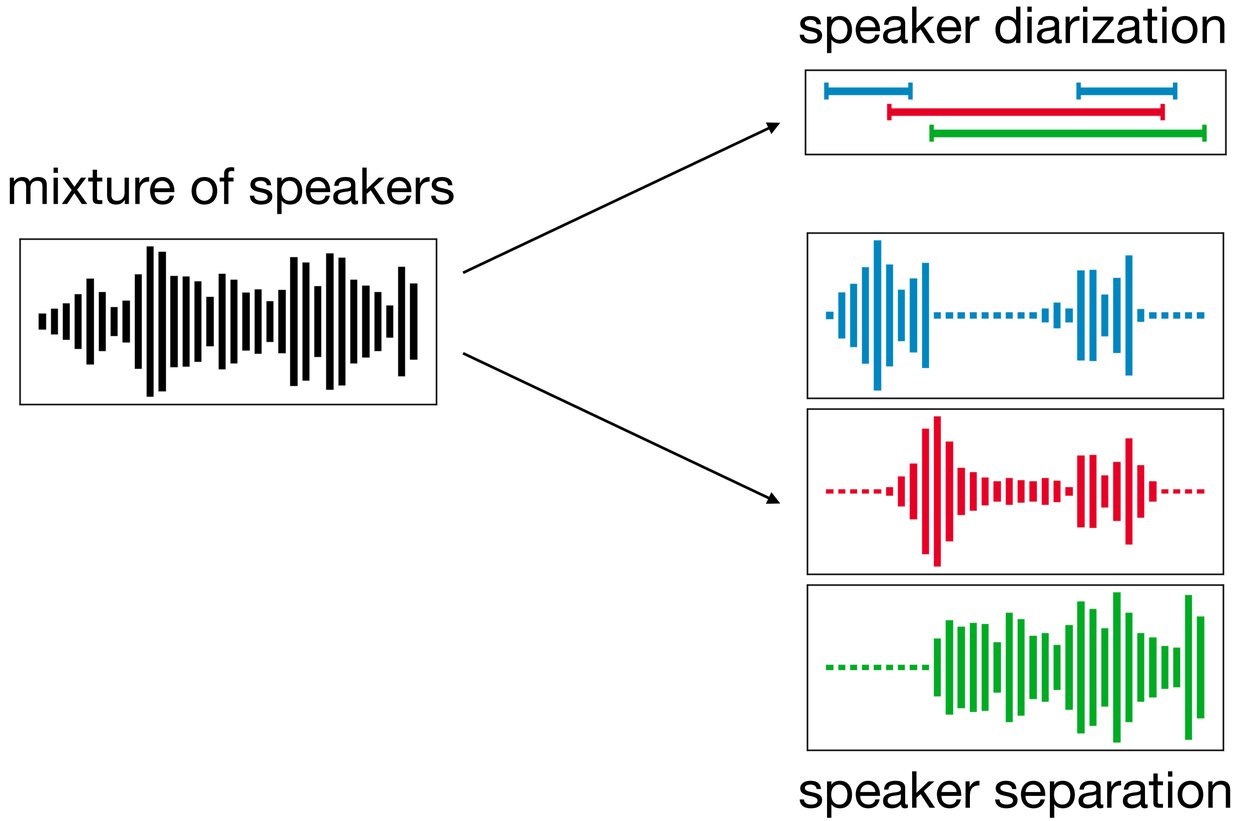




pyannoteAI

Joint speaker diarization & separation

two complementary tasks



- **speaker activations** can be used to guide separation
- **separated sources** can help assign overlapping speech to the right speakers
- why not **train them jointly** in a **semi-supervised** manner?

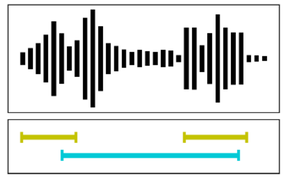
  - supervised diarization
  - unsupervised separation

pyannoteAI

Joint speaker diarization & separation

PIT + MixIT = PixIT loss

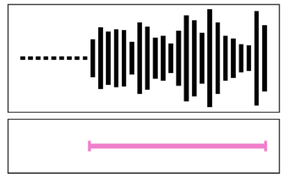
$\mathbf{x}^1$



$\mathbf{y}^1$

5 seconds chunks
with diarization labels

$\mathbf{x}^2$

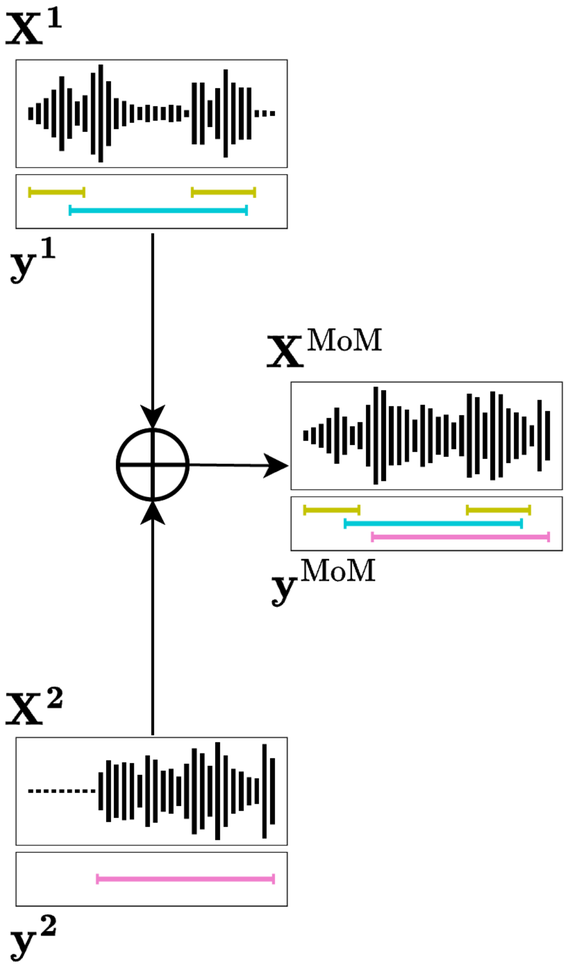


$\mathbf{y}^2$

pyannoteAI

Joint speaker diarization & separation

PIT + MixIT = PixIT loss

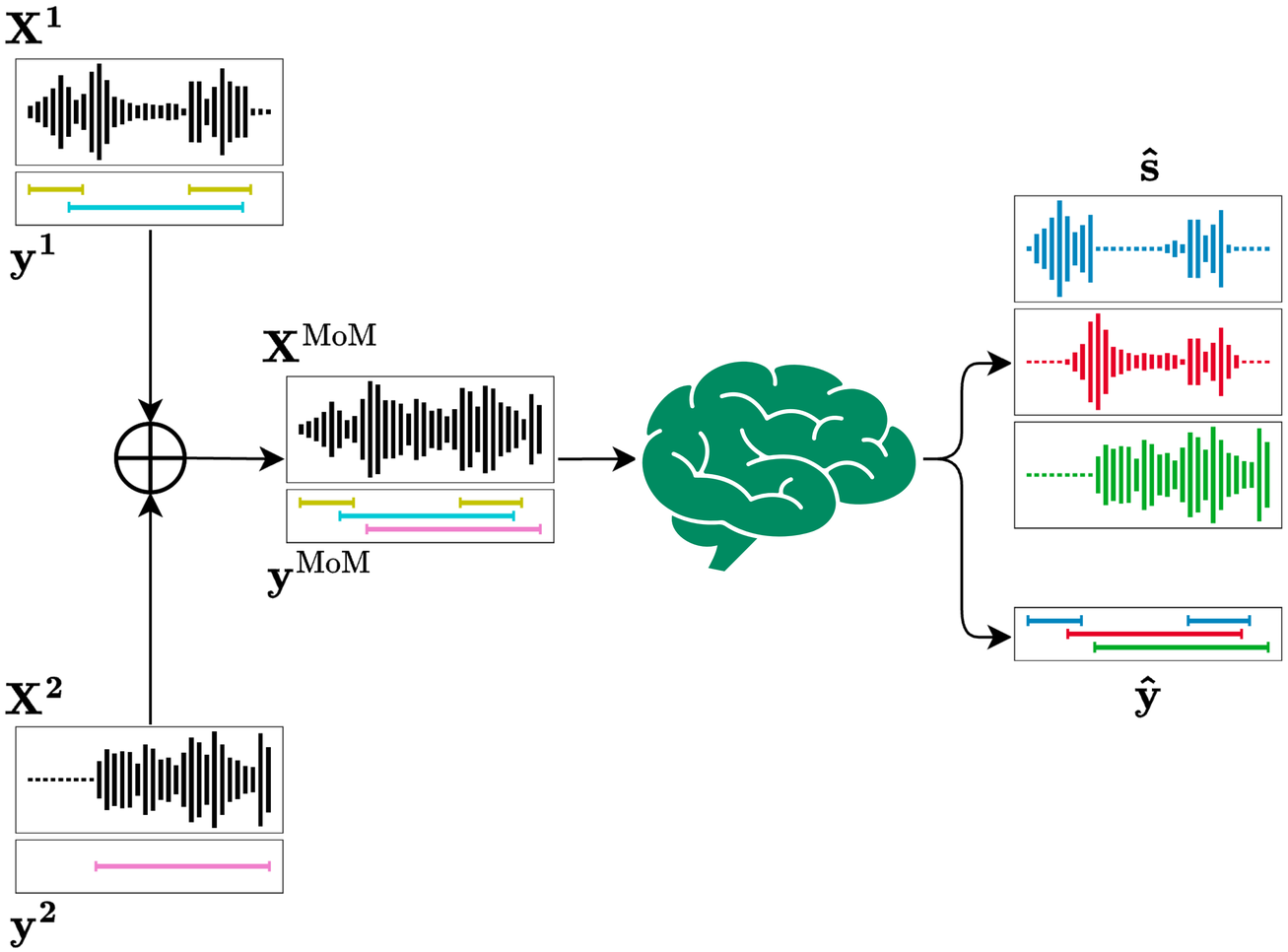


3 speakers max

pyannoteAI

Joint speaker diarization & separation

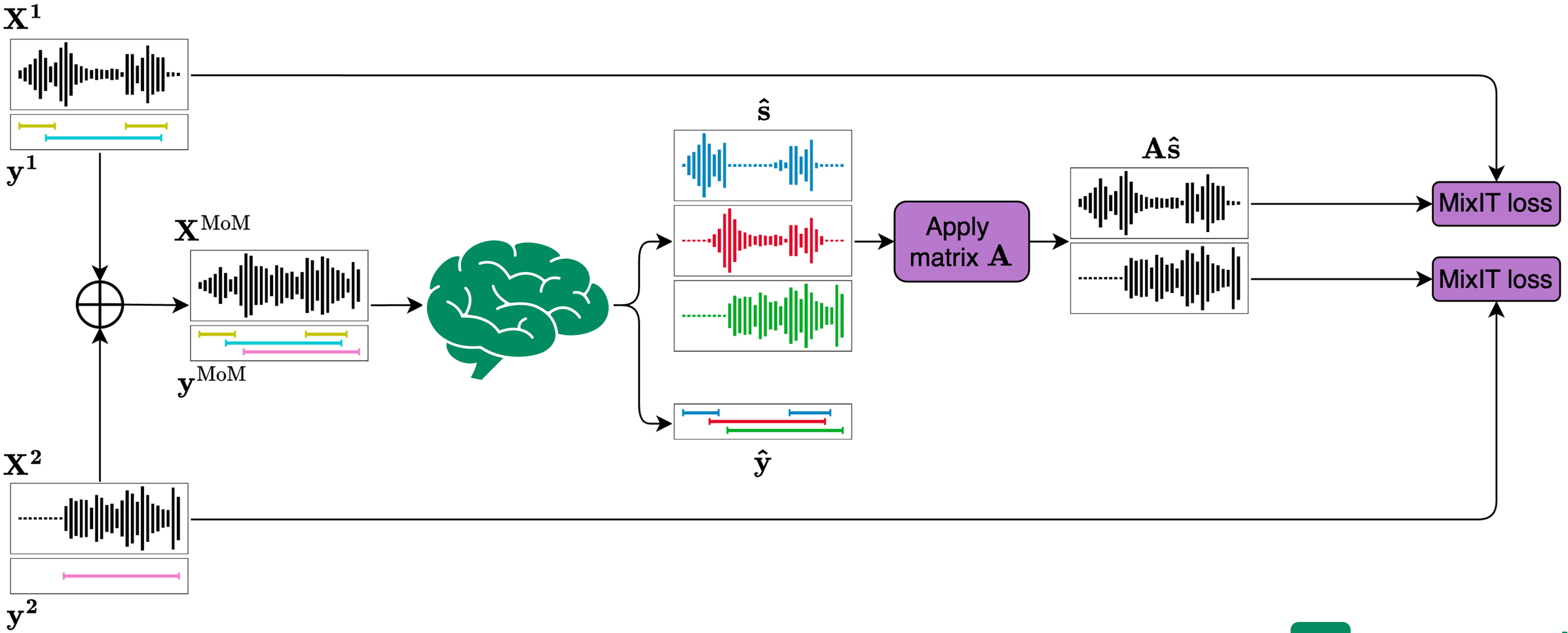
PIT + MixIT = PixIT loss



pyannoteAI

Joint speaker diarization & separation

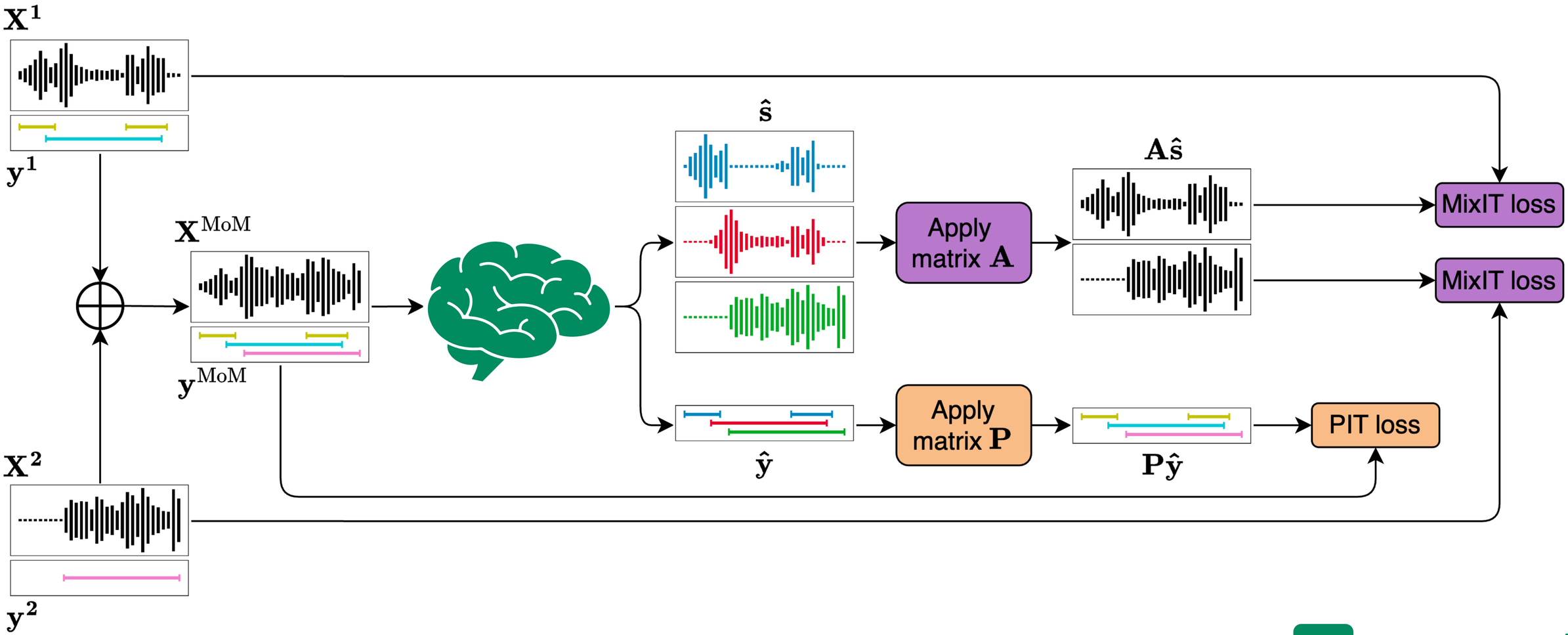
PIT + MixIT = PixIT loss



pyannoteAI

Joint speaker diarization & separation

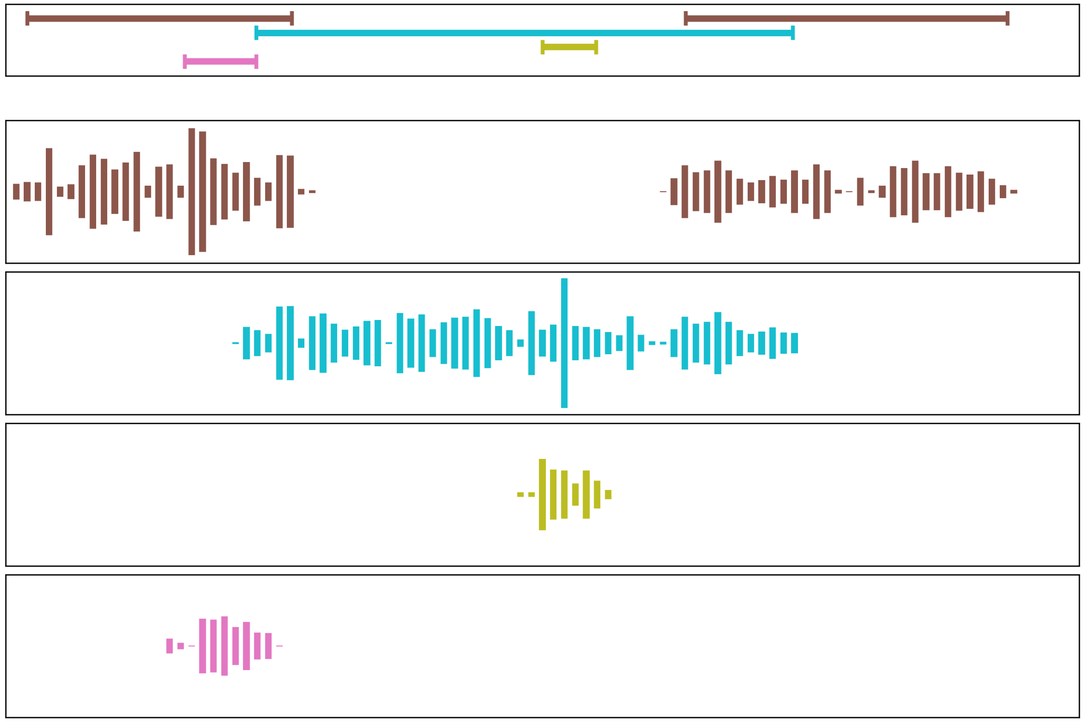
PIT + MixIT = PixIT loss



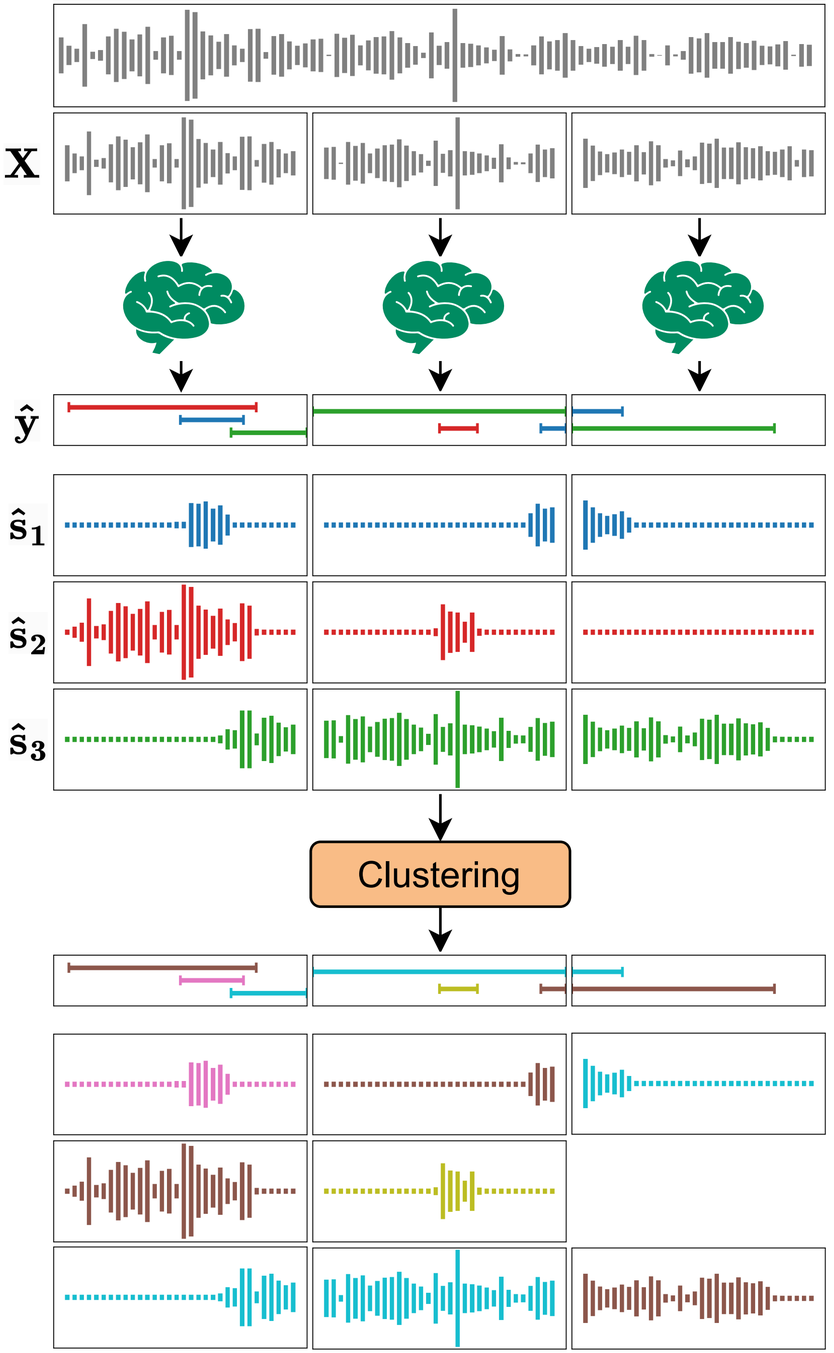
pyannoteAI

Diarization & separation

Inference on long-form audio

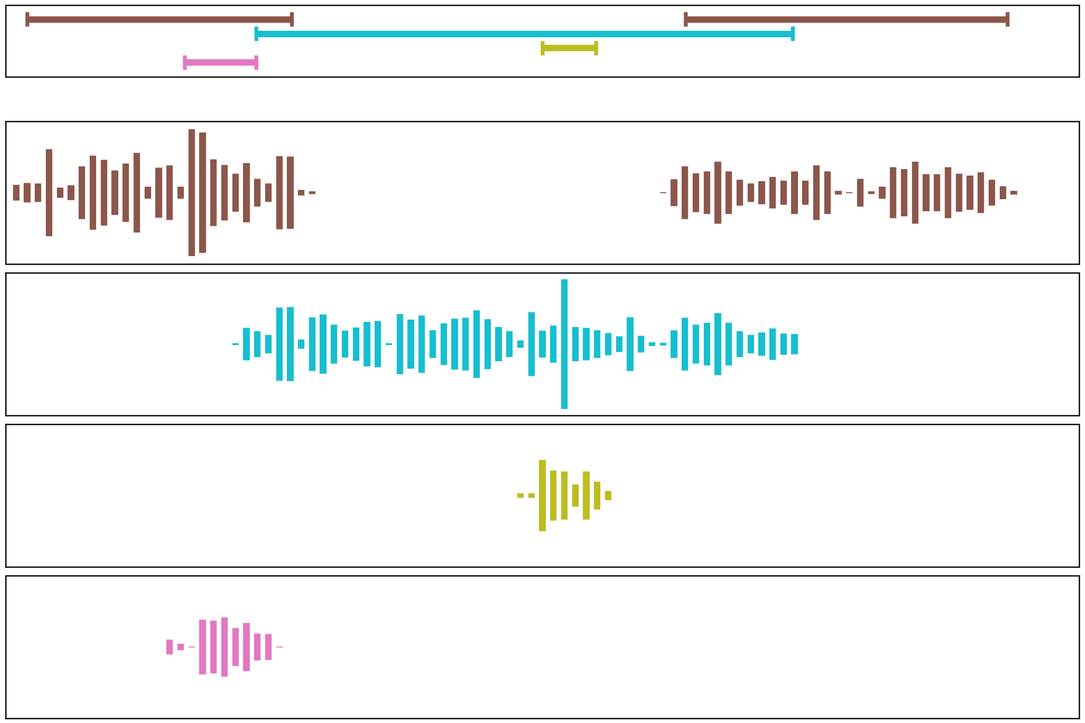


stitched, and transcribed separately



Diarization & separation

Inference on long-form audio



stitched, and transcribed separately

accounts for transcription
and diarization errors

Table 2: The cpWER (%) on AMI-SDM for various ASR systems with speaker attribution (SA) done through diarization or the joint model

ASR model	SA method	SA system	cpWER(%) sub	del	ins	**total**	Relative Change
Whisper small.en	Diarization	pyannote 3.1	8.7	27.2	3.7	**39.6**	
	Diarization	PixIT	8.5	27.3	2.1	**37.9**	-4.3%
	Separation	PixIT	6.7	25.8	1.4	**33.9**	-14.4%
Whisper medium.en	Diarization	pyannote 3.1	7.4	28.0	3.4	**38.8**	
	Diarization	PixIT	7.3	27.8	2.0	**37.1**	-4.4%
	Separation	PixIT	5.9	25.8	1.2	**32.8**	-15.4%
Whisper large-v2	Diarization	pyannote 3.1	7.1	29.3	1.8	**38.3**	
	Diarization	PixIT	6.9	26.6	2.1	**35.7**	-6.7%
	Separation	PixIT	5.6	24.7	1.3	**31.7**	-17.2%
NeMo conformer large	Diarization	pyannote 3.1	11.5	36.0	1.4	**48.9**	
	Diarization	PixIT	13.3	33.9	1.3	**48.5**	-0.8%
	Separation	PixIT	13.4	24.6	1.4	**39.4**	-19.4%

pyannoteAI

Democratizing seamless human interaction,
grounded in scientific excellence

pyannoteAI labs

All things "speaker"

HIRING

- speaker diarization
- speaker identification
- speaker separation
- speaker-attributed transcription
- streaming speaker diarization
- streaming speaker identification
- interactive speaker diarization

pyannoteAI labs

Resources

🧠 brains

Hervé Bredin, PhD
CSO, creator of pyannote.audio

Juan Coria, PhD
CTO, creator of diart

Antoine Laurent, PhD
researcher, co-inventor of pyannote diarization model

your name goes here

💻 compute

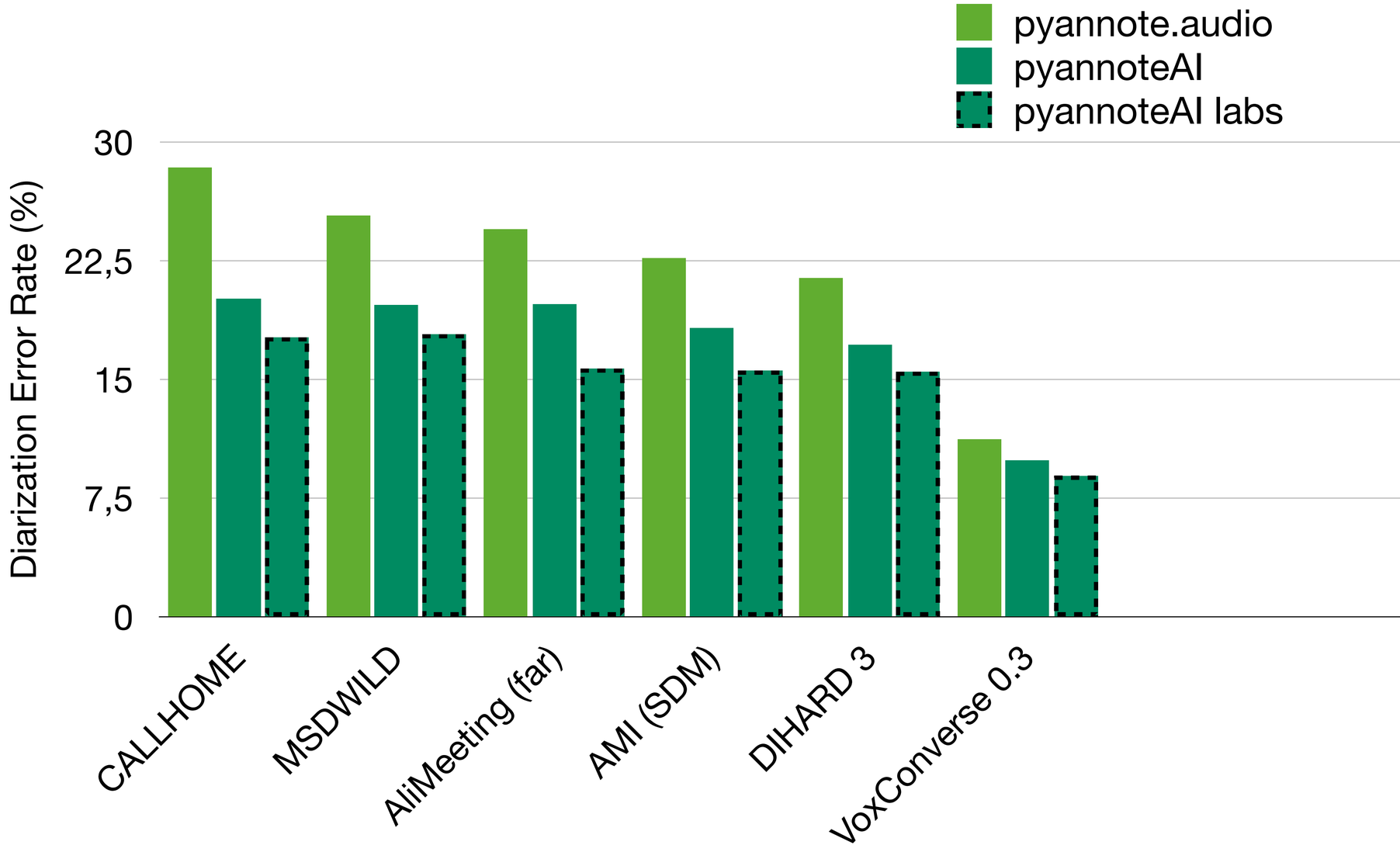
Jean Zay supercomputer
364 x 4 x H100 80Go

💰 money

pyannoteAI labs

Speaker diarization

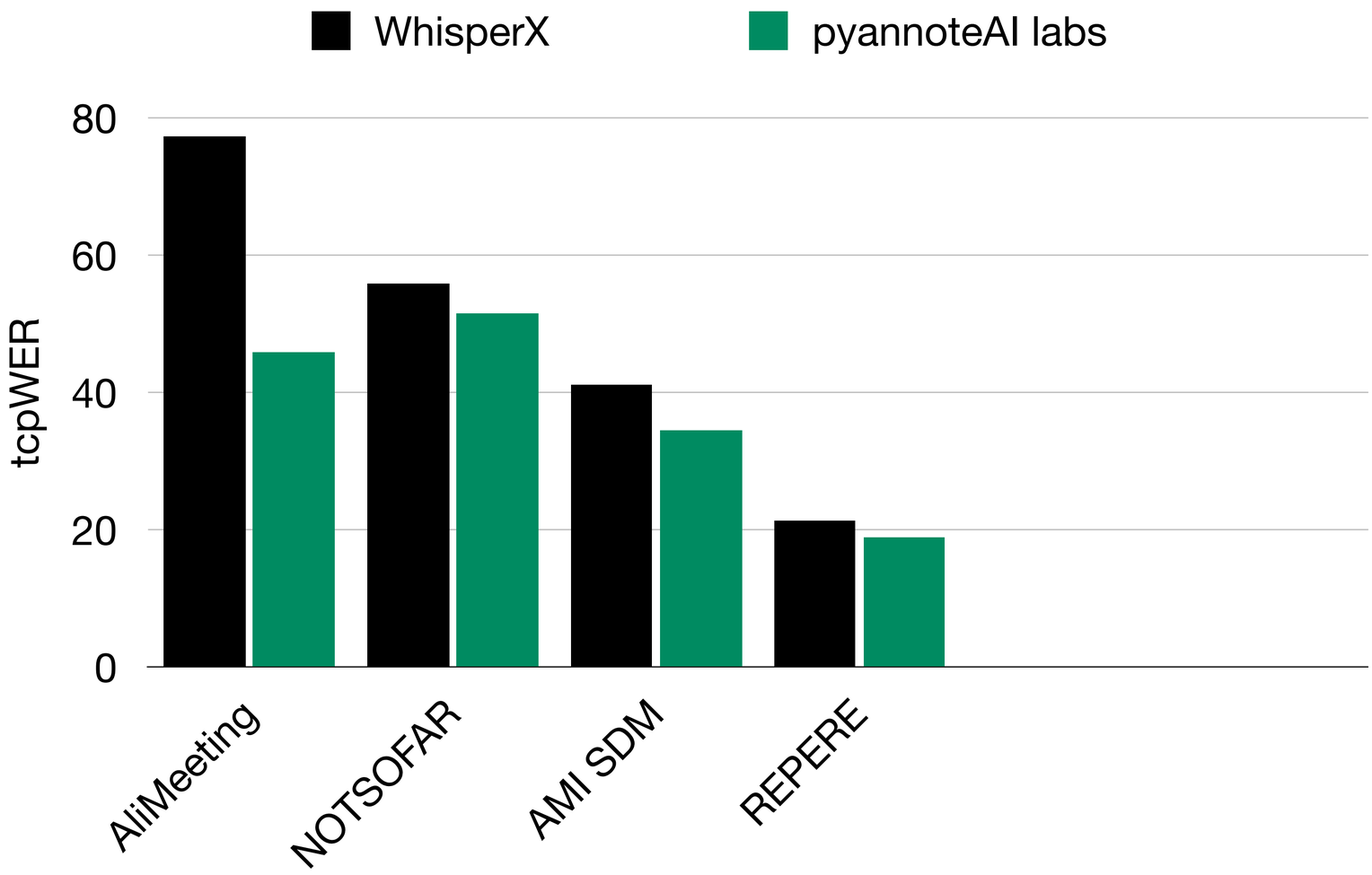
Without finetuning



pyannoteAI labs

Speaker-attributed transcription

Off the shelf
faster-whisper-turbo



pyannoteAI

Hervé Bredin
herve@pyannote.ai