

INTERSPEECH 2026 TUTORIAL

Speaker Modeling beyond Recognition



Foundations, Cross-Task Applications, and Responsible Deployment



Shuai Wang
Nanjing University



Quan Wang
Google DeepMind



Kong Aik Lee
The Hong Kong
Polytechnic University

Tutorial team

Shuai Wang

Nanjing University

Speaker representation learning, target-speaker speech processing, and multimodal speaker perception.

Quan Wang

Google DeepMind

Production speaker recognition and diarization, personalization, and system harnessing.

Kong Aik Lee

The Hong Kong
Polytechnic University

Voice biometrics, speech privacy, speaker anonymization, and spoofing countermeasures.

One tutorial, three perspectives: modeling, system engineering, and responsible deployment.

Introduction

Speech carries cues to **who is speaking**, alongside what is said. Speaker modeling aims to capture this identity despite changes in content and recording conditions.

Speaker identity matters when systems distinguish between individuals or adapt to a person's voice. This makes speaker modeling a foundation for speech applications beyond recognition.

This tutorial connects **speaker representation learning** with the design of speech systems, including the security and privacy implications of using speaker identity.

Tutorial structure

CHAPTER 1

Representations and Applications

Foundations, training objectives, pre-trained models, robustness, and cross-task speaker modeling.

Shuai Wang
Nanjing University

CHAPTER 2

Harnessing Speaker Models

Routing, enrollment memory, constraints, post-processing, and long-form diarization around frozen backbones.

Quan Wang
Google DeepMind

CHAPTER 3

Responsible Deployment

Voice privacy, spoofing countermeasures, and speaker anonymization.

Kong Aik Lee
HK PolyU

The tutorial moves from what a speaker model learns, to how a complete system uses it, to the risks and safeguards that shape deployment.

Chapter 1 of 3

Speaker Modeling from Representations to Applications



Shuai Wang Nanjing University

About the lecturer



Shuai Wang
Nanjing University

Current appointments

Associate Professor at Nanjing University, with adjunct appointments at Shenzhen Loop Area Institute, Shenzhen.

Education and industry experience

Ph.D. from Shanghai Jiao Tong University in 2020.
Former Senior Researcher and speech team lead at Tencent LightSpeed Studios.

Research and open source

Speaker modeling, target-speaker speech processing, and multimodal speaker perception. Initiator of *WeSpeaker* and *WeSep*.

IEEE Senior Member and ISCA Member. Winner of VoxSRC 2019, DIHARD 2019, and NIST SRE 2024.

What this chapter is about

TUTORIAL TITLE

Speaker Modeling
beyond Recognition

FOCUS OF CHAPTER 1

Representations
and Applications

KEY TAKEAWAY

A speaker representation should retain task-relevant evidence and suppress irrelevant variation.

Agenda

01

Speaker Modeling
Foundations

02

Supervised Speaker
Learning

03

Pre-trained Models
and SSL

04

Speaker Modeling
Across Tasks

05

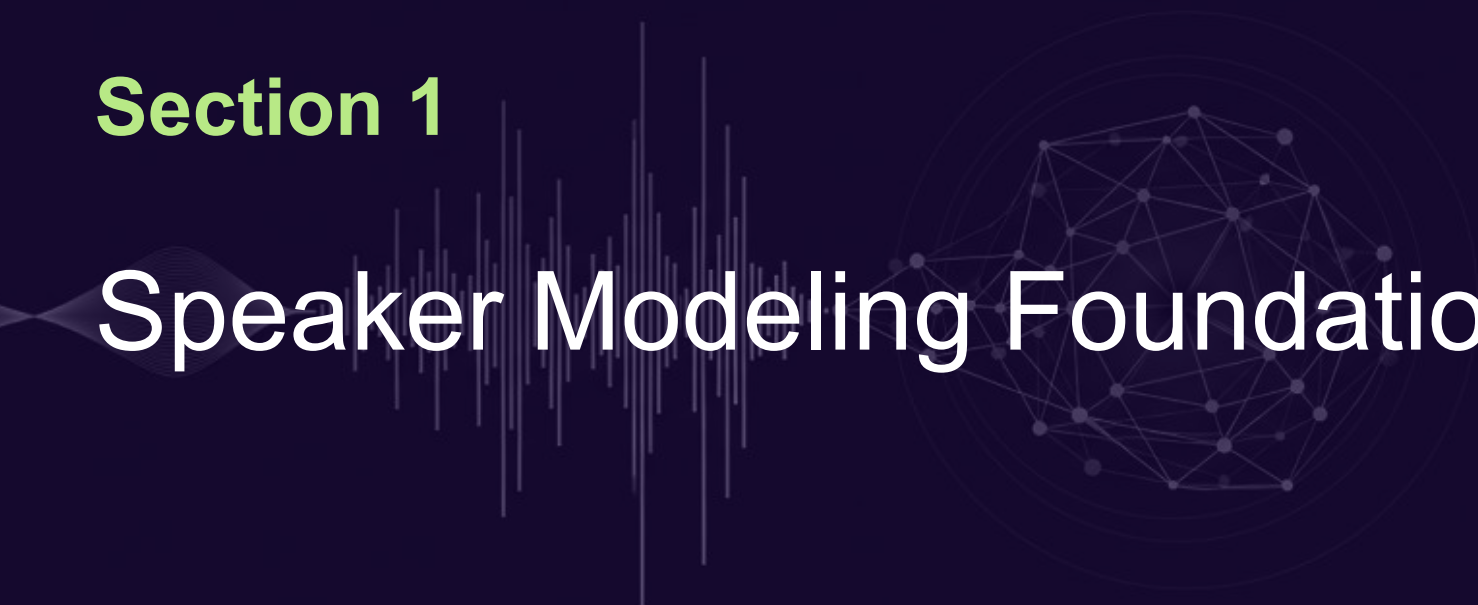
Multi-Modal Speaker
Modeling

06

Frontiers and
Resources

Section 1

Speaker Modeling Foundations



Speaker Modeling

Voice as a Biometric and Behavioral Signal



Speaker Modeling

Definition and Scope

Definition

Speaker modeling aims to characterize and recognize an individual's unique traits by analyzing patterns embedded in speech signals.

Main Applications:

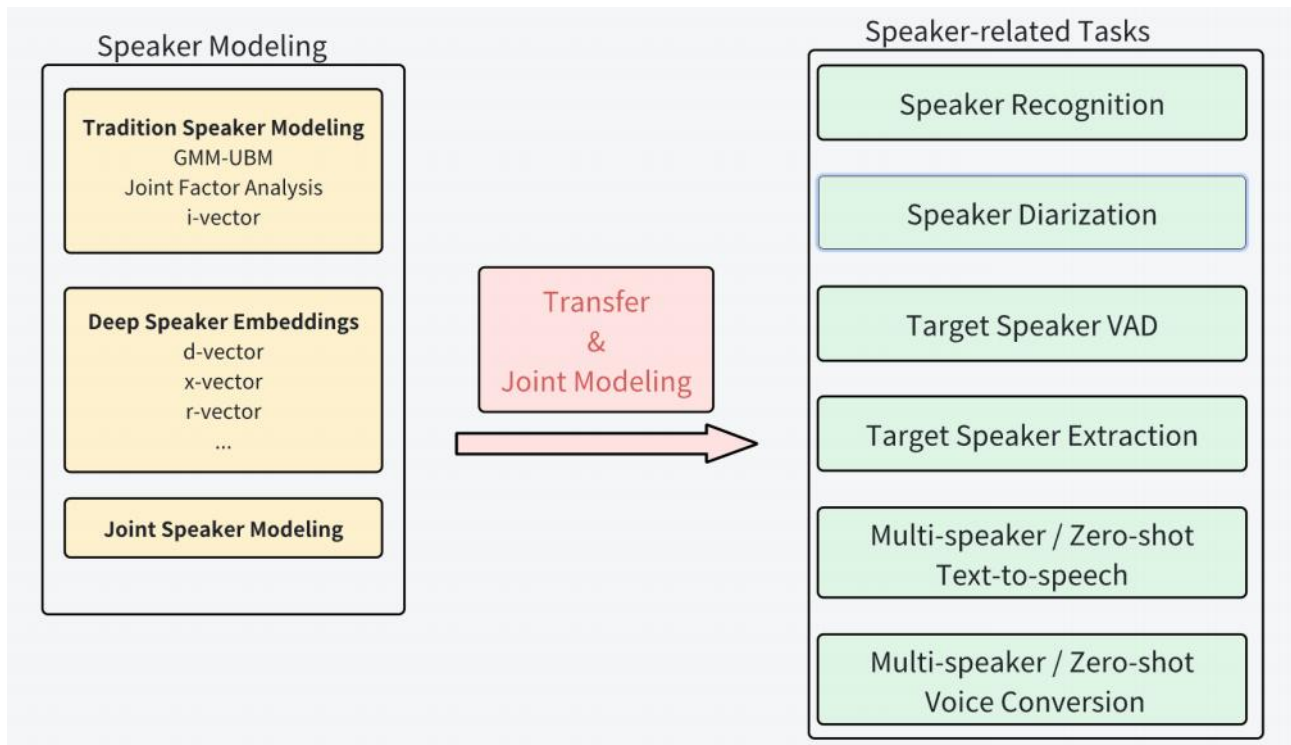
- Speaker recognition
- Speaker diarization
- Voice cloning
- Speech synthesis
- Target speaker extraction

Real-world Impact:

- Biometric authentication
- Surveillance systems
- Personalized services
- Forensic analysis
- Personalized interaction

Speaker Modeling

Application Landscape



Evolution of Speaker Representations

Four Paradigm Shifts

1. **From VQ to GMM** (1980s–1990s)

- Vector quantization → Gaussian mixture models
- Key advance: modeling uncertainty with covariance matrices

2. **From GMM-EM to GMM-UBM** (2000s)

- Universal background model to improve generalization
- MAP adaptation replaces EM-only training for speaker models

3. **From Supervectors to i-vectors** (2010s)

- Dimensionality reduction and channel compensation
- Low-dimensional speaker representations

4. **From Generative to Discriminative** (2015–present)

- Deep neural networks for speaker embeddings
- End-to-end discriminative training

Evolution of Speaker Representations

VQ to GMM: From Codebooks to Probabilistic Models

Methods and Representations

- Vector Quantization (VQ): discrete codebook representation; distance-based matching.
- Gaussian Mixture Models (GMM): continuous-density generative modeling, trained via EM.
- Likelihood-based scoring replaces heuristic distances; better fits acoustic feature distributions.

Limitations of VQ

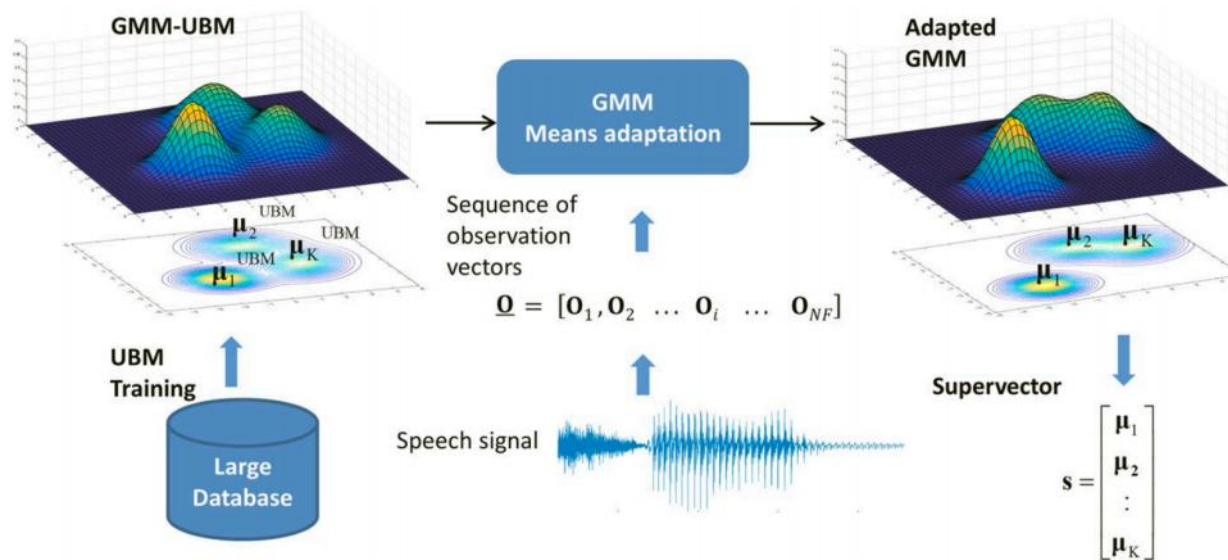
- No explicit uncertainty modeling; fragile under channel/noise variations.
- Fixed, hand-crafted codebooks; limited capacity and adaptability.

Why GMM Dominated

- Soft assignment and covariance modeling capture within-speaker variability.
- Principled maximum-likelihood training; extendable to adaptation and compensation.

Evolution of Speaker Representations

GMM-UBM and GMM Supervectors^[1]



From population model to speaker vector

A GMM models acoustic uncertainty. Adapt a universal background model to the enrolled speaker, then concatenate the mixture means into a supervector.

^[1]Reynolds et al., "Gaussian mixture models."

Evolution of Speaker Representations

i-vector Representation^[2]

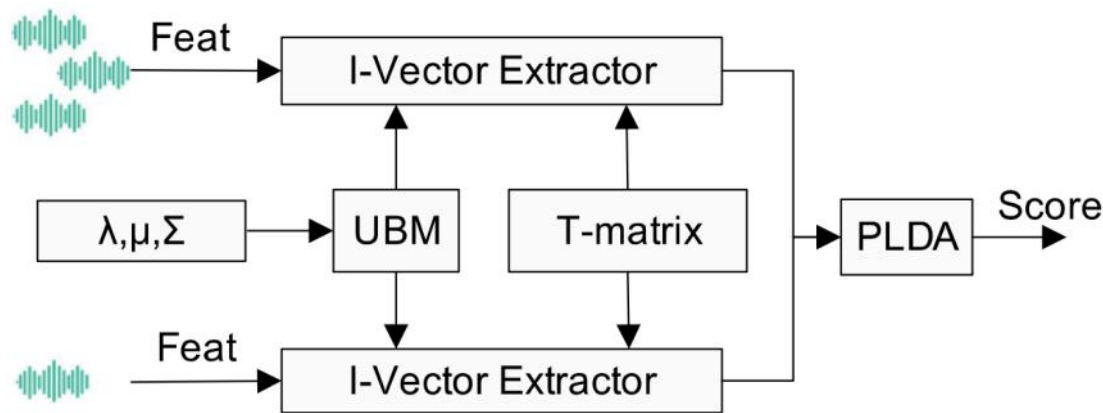


Figure 1: Block diagram of an i-vector based speaker recognition system

Low-dimensional factorization

$M(s) = m + T w(s)$ The i-vector $w(s)$ summarizes speaker and channel variability in a compact space.

^[2]Dehak et al., "Front-end factor analysis for speaker verification".

Evolution of Speaker Representations

Generative Models to Discriminative Embeddings

Neural Embeddings and Training

- x-vector/TDNN, ResNet/ECAPA encoders; attentive statistics pooling for temporal aggregation.
- Large-margin classification losses (AM-Softmax, AAM-Softmax) for discriminative speaker spaces.
- Data augmentation and domain adaptation for robustness (reverb, noise, codecs, channels).

Trends and Performance

- Self-supervised pretraining (e.g., wav2vec 2.0, HuBERT) as frontend or via end-to-end fine-tuning.
- Simple cosine/PLDA back-ends remain competitive with well-calibrated embeddings.
- Continuous improvements on open benchmarks (e.g., VoxCeleb) under constrained scoring (EER/minDCF).

Deep Speaker Encoder Design

Problem Formulation

Speaker representation learning:

Given an utterance $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_T\} \in \mathbb{R}^{T \times D}$, learn a mapping function F to extract a speaker representation:

$$\mathbf{v} = F(\mathbf{O}) \in \mathbb{R}^d$$

where

- $\mathbf{o}_t$: frame-level acoustic feature at time t
- T : number of frames
- D : feature dimension
- $\mathbf{v}$: fixed-length speaker embedding
- d : embedding dimension

Objective: embeddings from the same speaker should be close, and those from different speakers should be far apart.

Deep Speaker Encoder Design

Frame-level vs. Segment-level Optimization

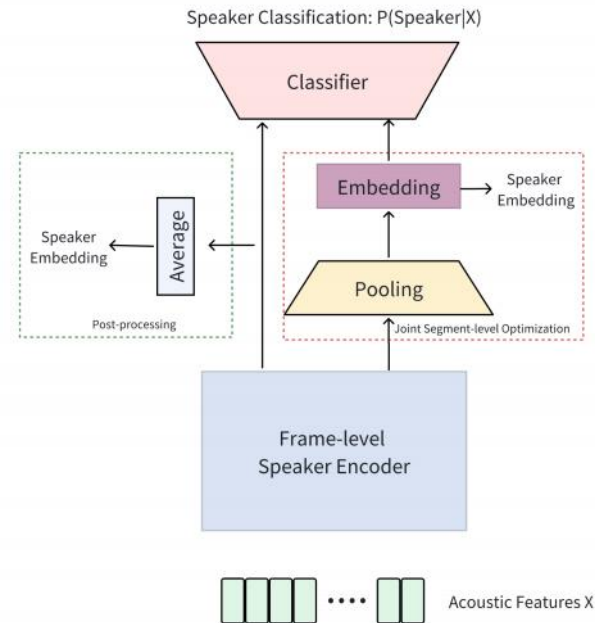
Frame-level (d-vector)

- Train with frame-level labels
- Aggregate after training
- Simple but limited performance

$$\mathbf{v} = \frac{1}{T} \sum_{t=1}^T \mathbf{f}_t \quad (1)$$

Segment-level (x-vector)

- Train with utterance-level labels
- Integrate aggregation in-network
- Better performance



Deep Speaker Encoder Design

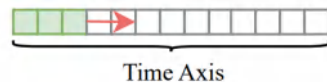
1D vs. 2D Convolution

1D Convolution

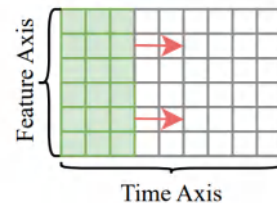
- Applied along time
- Lower computational cost
- Potentially large receptive field
- Simple architecture
- Limited frequency modeling

2D Convolution

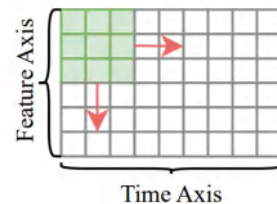
- Applied along time and frequency
- Better time-frequency modeling
- Higher computational cost
- Better performance potential



a) 1D Convolution for raw wav input



b) 1D Convolution for spectrogram input



c) 2D Convolution for spectrogram input

Deep Speaker Encoder Design

d-vector: Frame-Level Embeddings^[3]

Train a speaker classifier on short frames, then average frame embeddings into one utterance vector. Phonetic content can leak into the representation.

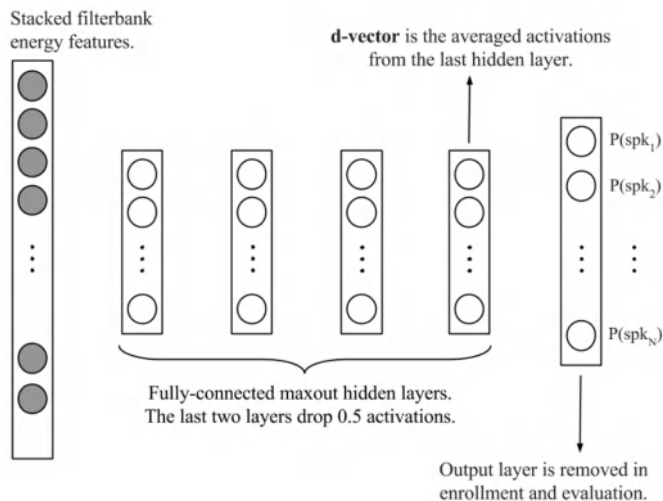


Figure 2: Architecture of d-vector

^[3]Variani et al., "Deep neural networks for small footprint text-dependent speaker verification".

Deep Speaker Encoder Design

x-vector: Segment-Level Embeddings^[4,5]

Train on complete utterances and pool frame features into $\mathbf{v} = [\mu; \sigma]$. ECAPA-TDNN extends this design.

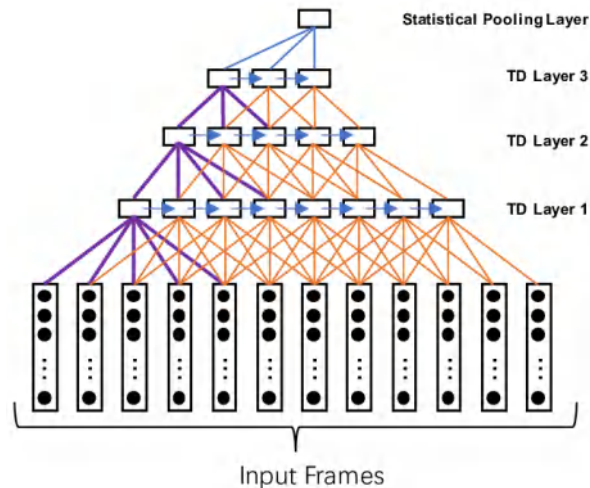


Figure 3: TDNN architecture used by x-vector

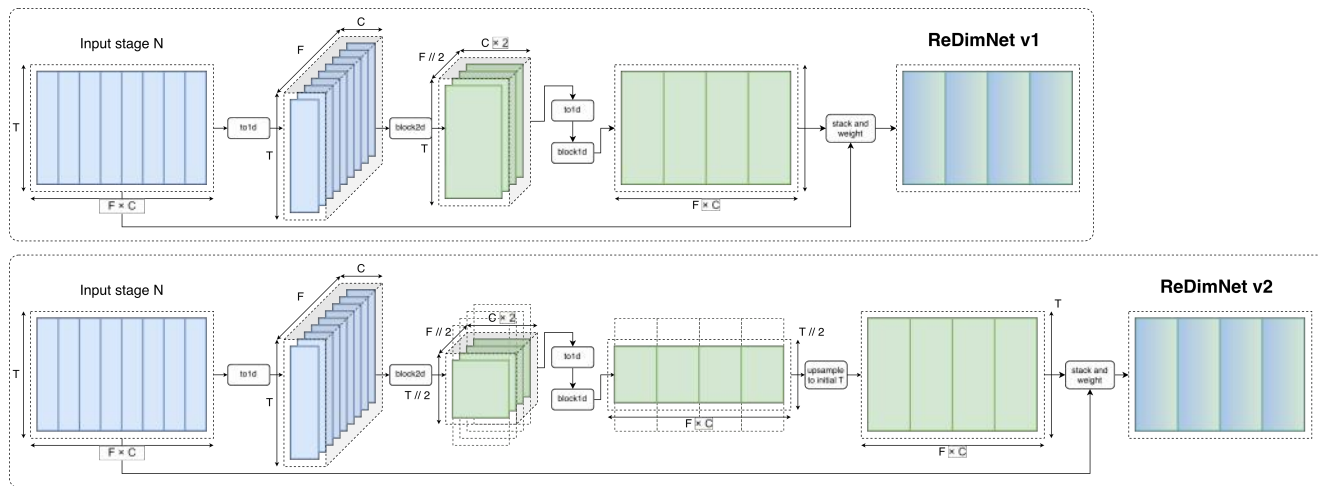
^[4]Snyder et al., “X-vectors: Robust dnn embeddings for speaker recognition”.

^[5]Desplanques, Thienpondt, and Demuynck, “Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification”.

Deep Speaker Encoder Design

ReDimNet2: Time–Frequency Reshaping^[6]

d-vector $\rightarrow$ x-vector $\rightarrow$ r/ ξ -vector $\rightarrow$ hybrid 1-D/2-D encoders



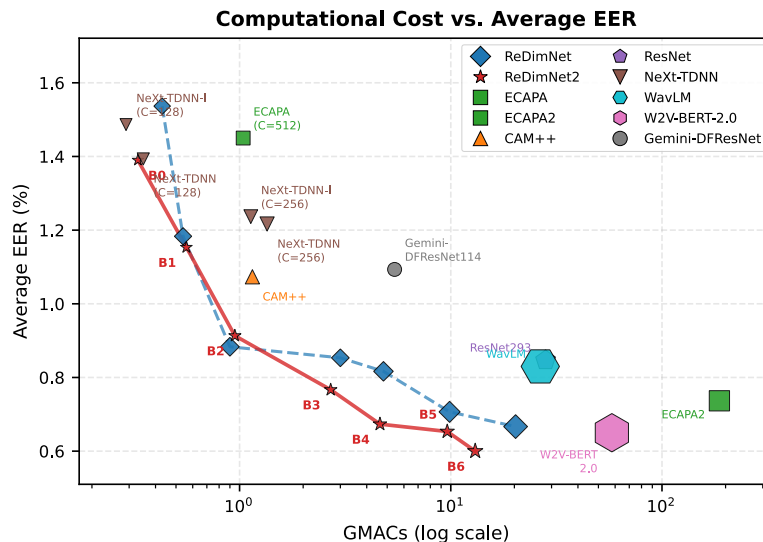
Architecture principle

Time pooling shortens the 1-D path; upsampling restores temporal resolution before stage-wise aggregation.

^[6]Yakovlev and Okhotnikov, ReDimNet2: Scaling Speaker Verification via Time-Pooled Dimension Reshaping.

Deep Speaker Encoder Design

Accuracy–Efficiency Trade-offs^[7]



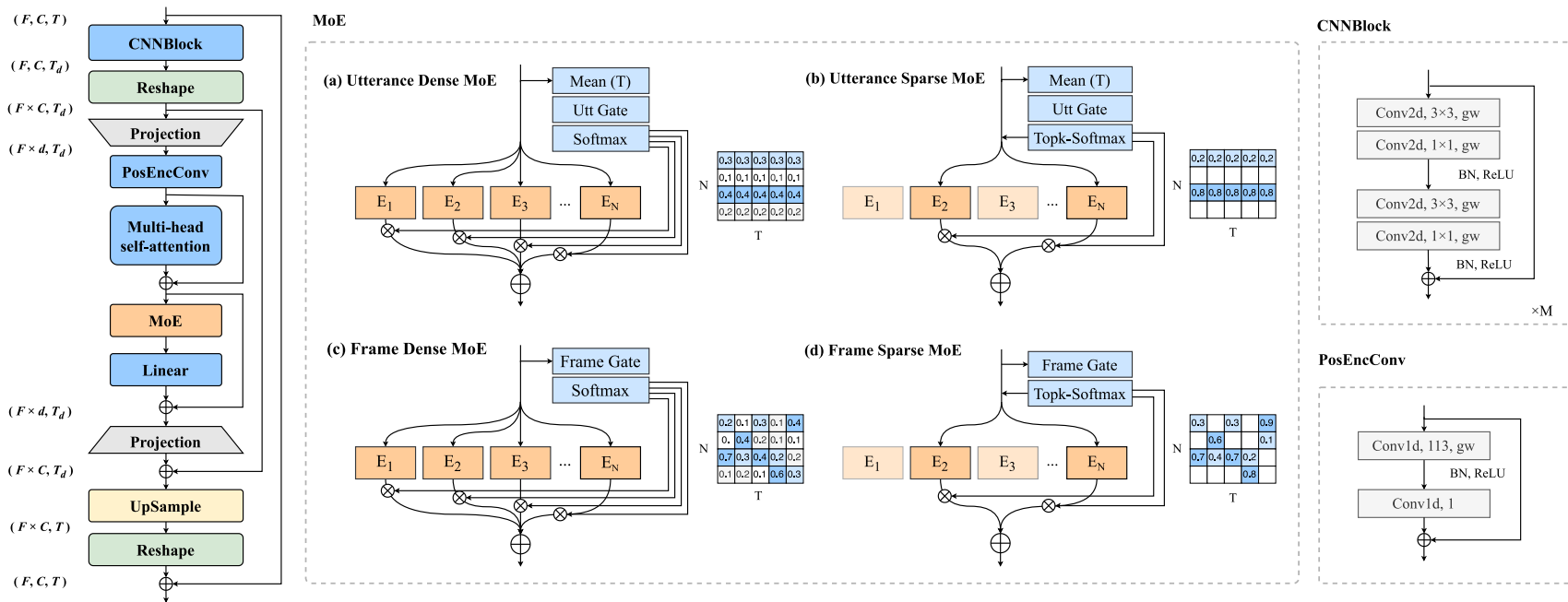
Model selection

The cost–accuracy frontier matters more than a single best result; deployment selects an operating point according to latency, memory, and error cost.

^[7]Yakovlev and Okhotnikov, *ReDimNet2: Scaling Speaker Verification via Time-Pooled Dimension Reshaping*.

Deep Speaker Encoder Design

MECT: CNN–Transformer with Expert Routing^[8]



Four MoE routers augment a reshaped CNN–Transformer encoder. MECT-B2 reports 0.27% Vox1-O EER with VoxCeleb2-only training (9.57M parameters).

Training Data and Supervision

Labeled and Unlabeled Data

Labeled data:

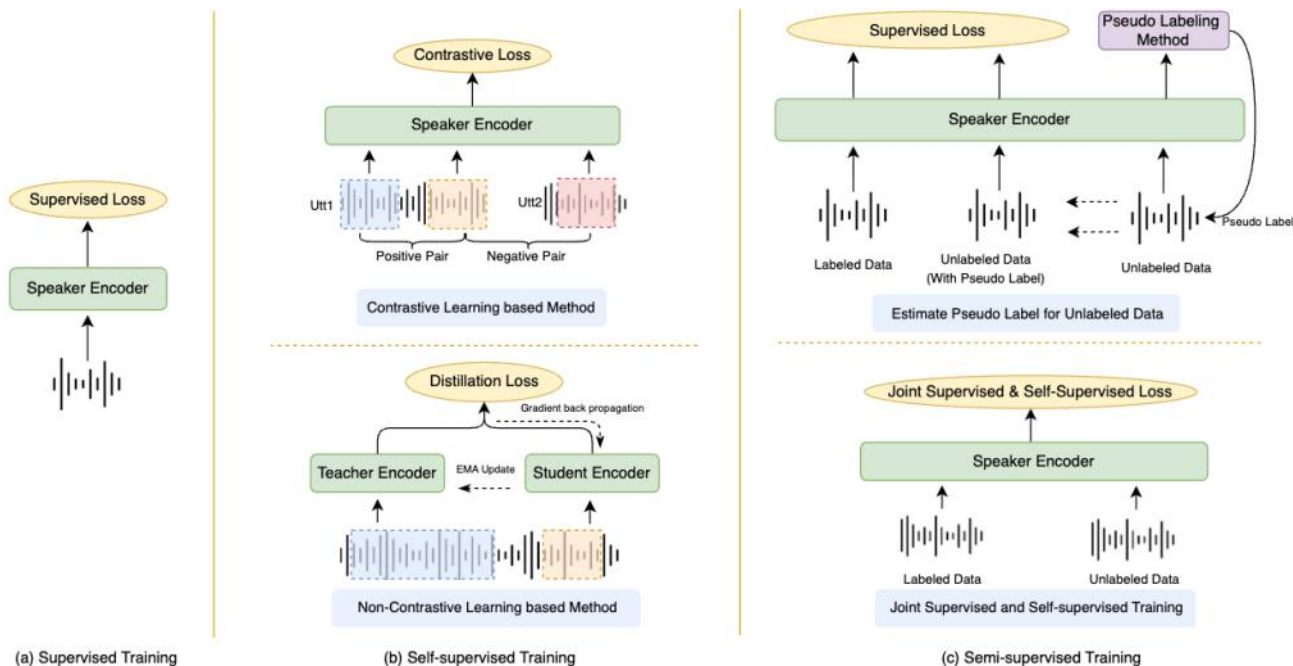
- Supports speaker classification and margin-based objectives
- Requires identity annotation, curation, and careful dataset design
- Coverage is often limited by language, channel, and recording domain

Unlabeled data:

- Scales to broader acoustic and linguistic conditions
- Requires a learning signal based on augmentation, context, or pseudo-labels
- Removes identity annotation while preserving large-scale acoustic diversity

Training Data and Supervision

Supervised, Semi-Supervised, and Self-Supervised Paradigms



Section 2

Supervised Speaker Learning

Supervised Training Objectives

Goal and Principle

Problem setting

Given an utterance $\mathbf{O} = \{\mathbf{o}_t\}_{t=1}^T \in \mathbb{R}^{T \times D}$, learn F that produces a fixed-length embedding $\mathbf{v} = F(\mathbf{O}) \in \mathbb{R}^d$.

- Same-speaker embeddings should be close; different-speaker embeddings should be far.
- Use classification-driven objectives that align with cosine/angle used at verification time.
- Introduce explicit margins to strengthen open-set generalization.

Supervised Training Objectives

Speaker Verification vs. ASR: Open-Set Motivation

ASR (classic acoustic modeling)

- Closed-set at inference: phoneme/senone labels are fixed.
- Objective: maximize classification accuracy on known label set.

Speaker verification

- Open-set at inference: speakers are unseen during training.
- Need compact clusters and clear margins under cosine/angle.

Supervised Training Objectives

Softmax Baseline for Speaker Classification

Model

Embedding $\mathbf{v} = F(\mathbf{O}) \in \mathbb{R}^d$, classifier $W = [\mathbf{w}_1, \dots, \mathbf{w}_C] \in \mathbb{R}^{d \times C}$ with bias $\mathbf{b}$.

- Logits: $s_j = \mathbf{w}_j^\top \mathbf{v} + b_j$.

$$L_{\text{CE}} = -\log \frac{e^{s_y}}{\sum_{j=1}^C e^{s_j}}.$$

Limitation for open-set

Encourages separation but does not explicitly enforce intra-speaker compactness or a margin in the cosine metric used at test time.

Supervised Training Objectives

Normalized Softmax on the Unit Hypersphere

Normalization

Enforce $\|\mathbf{v}\| = 1$, $\|\mathbf{w}_j\| = 1$, and set $b_j = 0$. Then

$$s_j = s \cos \theta_j,$$

where θ_j is the angle between $\mathbf{v}$ and $\mathbf{w}_j$, and $s > 0$ is a scale (inverse temperature).

$$L_{\text{Norm-CE}} = -\log \frac{e^{s \cos \theta_y}}{\sum_{j=1}^C e^{s \cos \theta_j}}.$$

Geometric view

Classification equals picking the smallest angle on the unit hypersphere, consistent with cosine scoring for $\mathbf{v}$.

Supervised Training Objectives

SphereFace: Multiplicative Angular Margin

Form

$$L = -\log \frac{e^{s \cos(m\theta_y)}}{e^{s \cos(m\theta_y)} + \sum_{j \neq y} e^{s \cos \theta_j}}, \quad m \geq 1.$$

- Decision boundary: from $\theta_1 = \theta_2$ to $m\theta_1 = \theta_2$.
- Pros: pioneering angular-margin formulation.

Caveat

Highly non-linear; training can be unstable and often needs annealing/special scheduling.

Supervised Training Objectives

CosFace (AM-Softmax): Additive Cosine Margin

Form

$$L = -\log \frac{e^{s(\cos \theta_y - m)}}{e^{s(\cos \theta_y - m)} + \sum_{j \neq y} e^{s \cos \theta_j}}, \quad m > 0.$$

- Interpretation: require $\cos \theta_y \geq \cos \theta_j + m$ (constant margin in cosine space).
- Pros: simple and stable optimization; no annealing.

Note

Margin is constant in cosine but not constant in angle; the angular gap varies with θ .

Supervised Training Objectives

ArcFace (AAM-Softmax): Additive Angular Margin

Form

$$L = -\log \frac{e^{s \cos(\theta_y + m)}}{e^{s \cos(\theta_y + m)} + \sum_{j \neq y} e^{s \cos \theta_j}}, \quad m > 0.$$

- Interpretation: require $\theta_y + m \leq \theta_j$ (constant angular gap).
- Pros: clean geometry; strong open-set performance.

Implementation tip

Ensure numerical stability (clip $\cos \theta$ to $[-1, 1]$; avoid unstable inverse trigonometric ops).

Supervised Training Objectives

Three Margins at a Glance

Method	Margin domain	Decision boundary (1 vs 2)
SphereFace	Angle (multiplicative)	$\cos(m\theta_1) = \cos(\theta_2)$
CosFace	Cosine (additive)	$\cos(\theta_1) - m = \cos(\theta_2)$
ArcFace	Angle (additive)	$\cos(\theta_1 + m) = \cos(\theta_2)$

Constancy

CosFace: constant in cosine.

ArcFace: constant in angle.

SphereFace: depends on θ .

Practice

Prefer ArcFace or CosFace for stability and performance; SphereFace mainly for historical context.

Supervised Training Objectives

Optional: Adding Center Loss

Form

$$L_{\text{center}} = \frac{1}{2} \sum_i \|\mathbf{v}_i - \mathbf{c}_{y_i}\|_2^2.$$

- Complements margin softmax by explicitly reducing intra-speaker variance.
- Combine as $L = L_{\text{ArcFace/CosFace}} + \lambda L_{\text{center}}$ with small λ .

Section 3

Pre-trained and Self-Supervised Learning

The background features a dark blue gradient. On the left, there is a faint, light-colored waveform. On the right, there is a faint, light-colored network graph with nodes and connecting lines, overlaid on concentric circles.

Label-Efficient Speaker Learning

Two Routes Beyond Speaker Labels

Route 1: Transfer from pre-trained speech models

- Start from SSL or ASR encoders
- Select and aggregate useful layers
- Adapt the full model or use PEFT

Route 2: Learn speaker structure without labels

- Construct positive pairs from utterances
- Apply contrastive or teacher–student learning
- Refine embeddings with pseudo-labels

Key distinction

Speech-model pre-training transfers a general acoustic representation; speaker-specific SSL directly defines which samples should share speaker identity.

Pre-trained Speech Models

Representative Encoder Families

Wav2Vec^[9], HuBERT^[10], WavLM^[11], UniSpeech^[12]

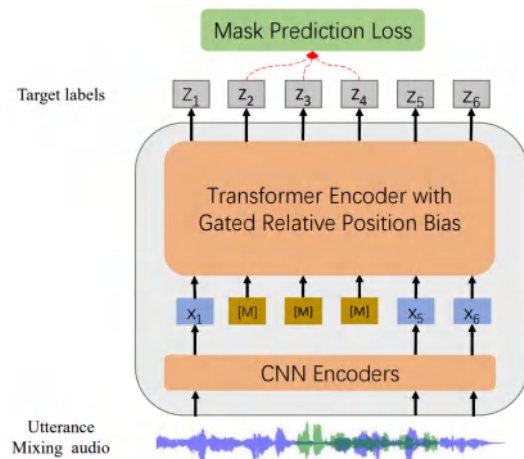


Figure 4: WavLM architecture

^[9]Baevski et al., "wav2vec 2.0: A framework for self-supervised learning of speech representations".

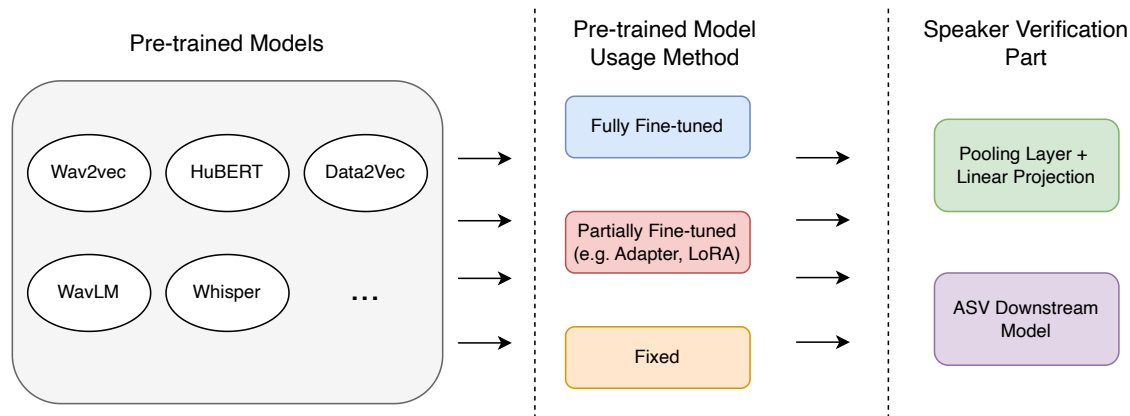
^[10]Hsu et al., "Hubert: Self-supervised speech representation learning by masked prediction of hidden units".

^[11]Chen et al., "Wavlm: Large-scale self-supervised pre-training for full stack speech processing".

^[12]Chen et al., "Unispeech-sat: Universal speech representation learning with speaker aware pre-training".

Pre-trained Speech Models

Feature Extraction, Fine-tuning, and Multi-task Learning



Integration Strategies:

1. **Feature Extraction:** use pretrained features as input
2. **Fine-tuning:** adapt pretrained models to speaker tasks
3. **Multi-task Learning:** jointly optimize multiple tasks

Pre-trained Speech Models

Layer Aggregation for Speaker Verification

Fine-tune **SSL encoders** for speaker verification^[13]

Replace Fbank with pre-trained features and learn a weighted mixture of layers.

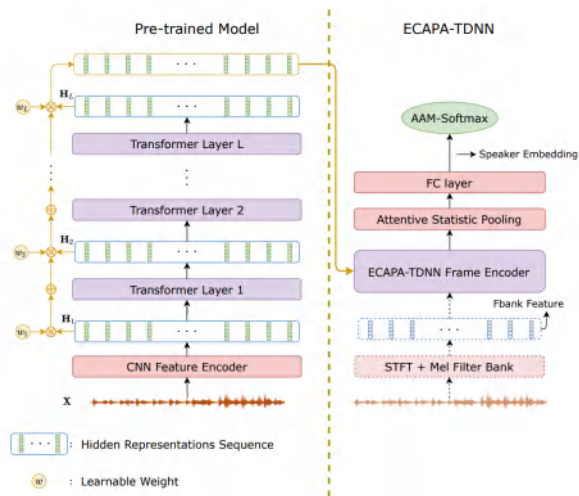


Figure 5: Using pretrained representations

^[13]Chen et al., “Large-scale self-supervised speech representation learning for automatic speaker verification”.

Pre-trained Speech Models

Adapter-Based Parameter-Efficient Tuning

Efficient SSL adaptation for speaker verification^[14]
Freeze the encoder and train small speaker adapters.

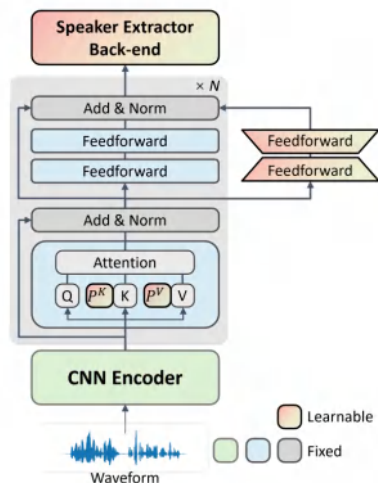


Figure 6: Efficient fine-tuning with adapters

^[14]Peng et al., "Parameter-efficient transfer learning of pre-trained Transformer models for speaker verification using adapters".

Pre-trained Speech Models

Large Encoders with Distillation and Structured Pruning

Use w2v-BERT 2.0 as the speaker encoder, then reduce deployment cost with distillation-guided structured pruning^[15].

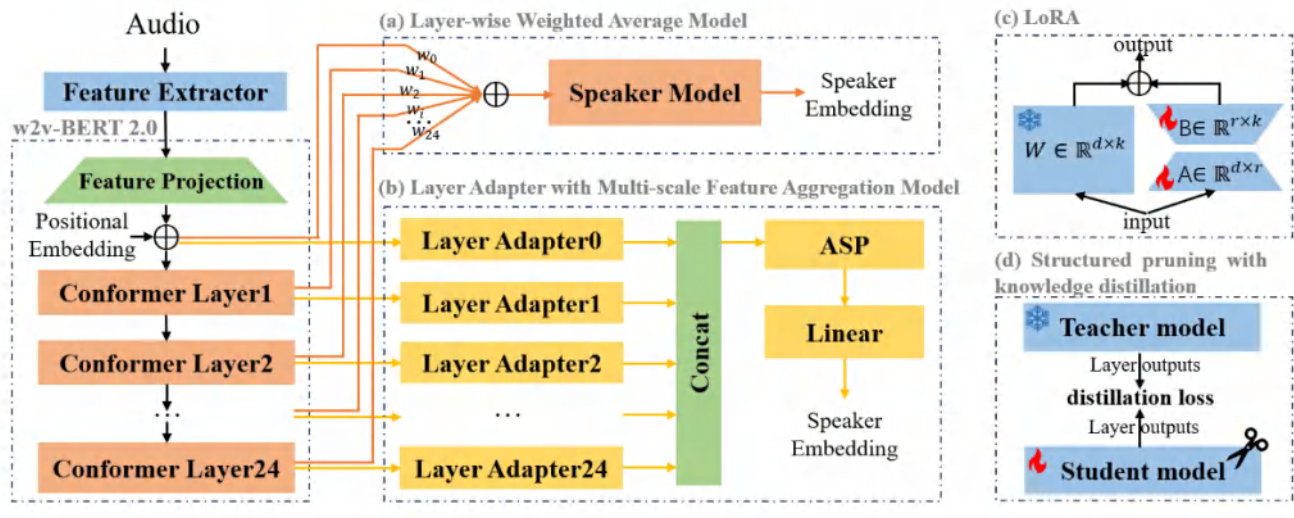


Figure 7: Enhancing SV with w2v-BERT 2.0 and distillation-guided structured pruning

^[15]Li, Cheng, and Li, "Enhancing Speaker Verification with w2v-BERT 2.0 and Knowledge Distillation guided Structured Pruning".

Pre-trained Speech Models

Transfer from ASR Encoders

Fine-tune **ASR models** on speaker verification^[16,17]

Pretrain with ASR datasets, then initialize the encoder for speaker tasks.

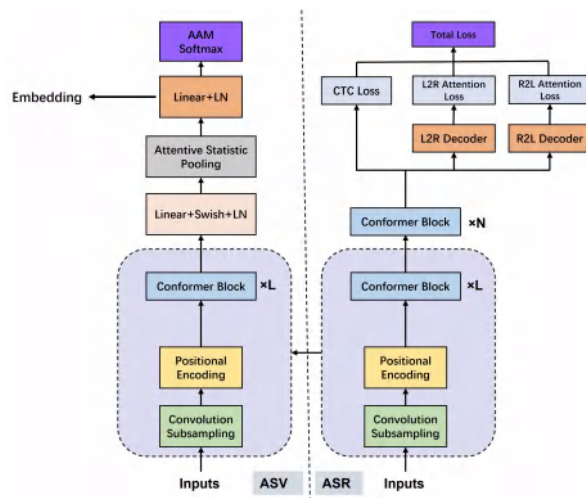


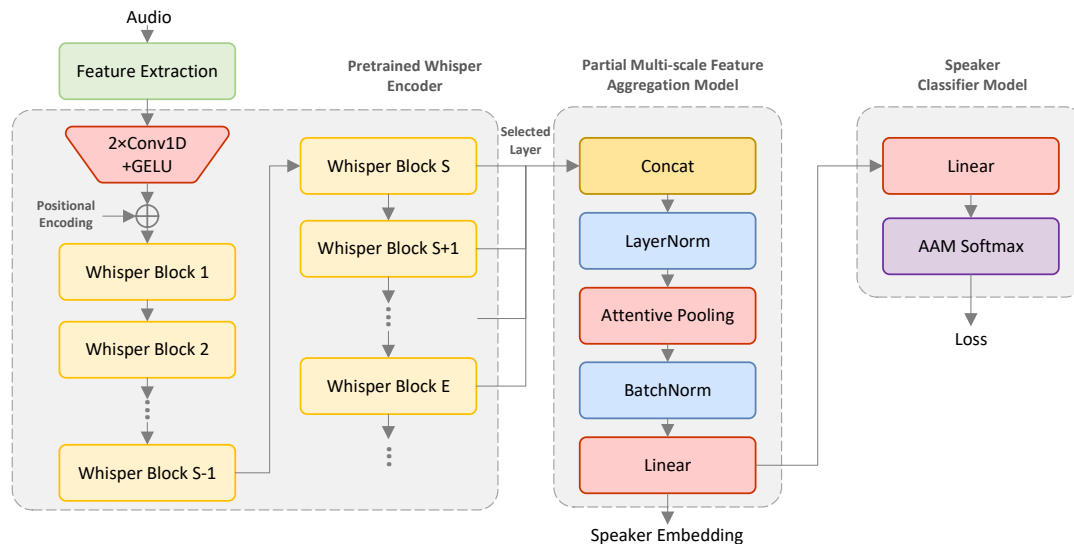
Figure 8: ASR transfer illustration

^[16]Liao et al., "Towards a unified conformer structure: from asr to asv task".

^[17]Cai et al., "Pretraining Conformer with ASR for Speaker Verification".

Pre-trained Speech Models

Whisper-PMFA: Layer Selection and Aggregation^[18]



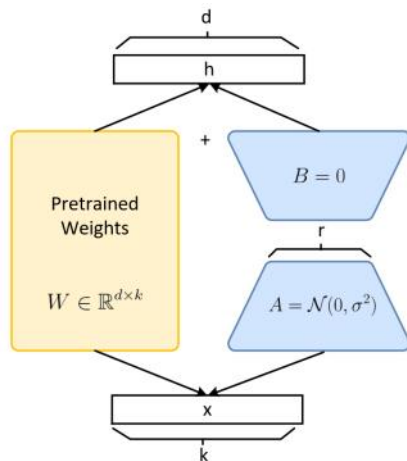
Partial multi-scale aggregation

Select a contiguous subset of Whisper blocks, aggregate them with attentive pooling, and train the speaker head with AAM-Softmax.

^[18] Zhao et al., "Whisper-PMFA: Partial Multi-Scale Feature Aggregation for Speaker Verification Using Whisper Models".

Pre-trained Speech Models

LoRA for Parameter-Efficient Adaptation^[19]



Low-rank update

- Freeze the pretrained matrix W
- Learn the smaller factors A and B
- Merge the update into the target projection

Evaluation requirement

Parameter efficiency does not guarantee calibration, robustness, or cross-language transfer.

^[19] Zhao et al., "Whisper-PMFA: Partial Multi-Scale Feature Aggregation for Speaker Verification Using Whisper Models".

Self-Supervised Speaker Learning

Metric-Learning Objectives

Metric learning losses provide contrastive supervision, e.g., Triplet, Prototypical, GE2E^[20], and Angular Prototypical^[21].

$$L_{\text{Triplet}} = \frac{1}{N} \sum_{j=1}^N \max(0, \|\mathbf{x}_{j,0} - \mathbf{x}_{j,1}\|_2^2 - \|\mathbf{x}_{j,0} - \mathbf{x}_{k \neq j,1}\|_2^2 + m)$$

$$L_{\text{Prototypical}} = -\frac{1}{N} \sum_{j=1}^N \log \frac{e^{\mathbf{S}_{j,j}}}{\sum_{k=1}^N e^{\mathbf{S}_{j,k}}}, \text{ where } \mathbf{S}_{j,k} = \|\mathbf{x}_{j,M} - \mathbf{c}_k\|_2^2$$

^[20]Wan et al., “Generalized end-to-end loss for speaker verification”.

^[21]Chung et al., “In defence of metric learning for speaker recognition”.

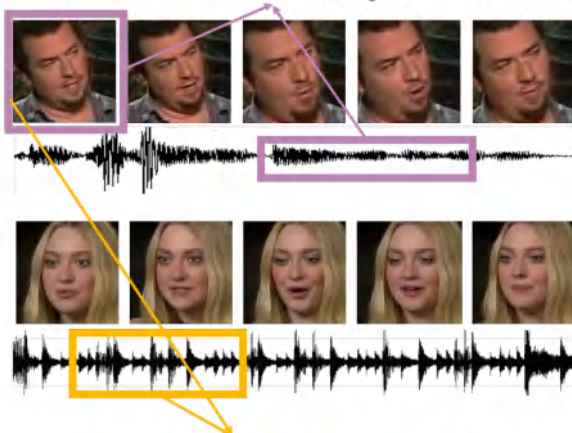
Self-Supervised Speaker Learning

Positive-Pair Assumption

Assumption for SSL on speaker verification^[22]

Same utterance: same speaker. Different utterances: different speakers.

Within the same track = same identity but different content



Different tracks = different identity and different content

Figure 9: Illustration of the assumption

^[22]Huh et al., “Augmentation adversarial training for self-supervised speaker recognition”.

Self-Supervised Speaker Learning

SimCLR for Speaker Embeddings^[23,24]

Two crops of one utterance form a positive pair. Crops from other utterances provide negatives.

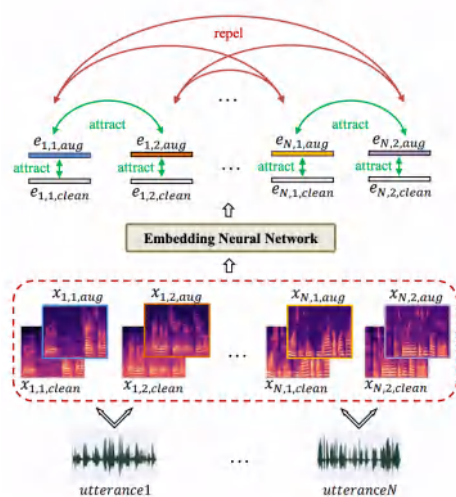


Figure 10: SimCLR for speaker task

^[23]Chen et al., “A simple framework for contrastive learning of visual representations”.

^[24]Zhang, Zou, and Wang, “Contrastive self-supervised learning for text-independent speaker verification”.

Self-Supervised Speaker Learning

DINO for Speaker Embeddings^[25,26,27]

Multiple crops of one utterance learn teacher–student consistency without explicit negatives.

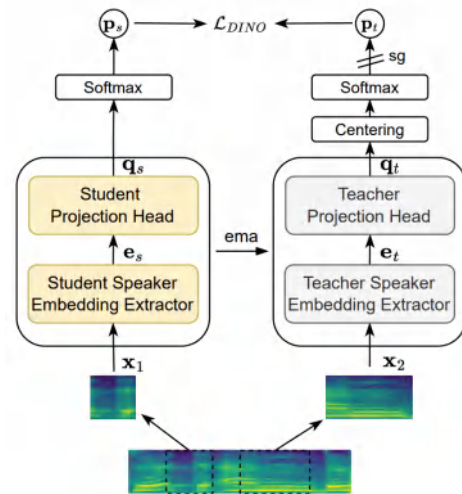


Figure 11: DINO for speaker task

^[25]Caron et al., “Emerging properties in self-supervised vision transformers”.

^[26]Han, Chen, and Qian, “Self-supervised speaker verification using dynamic loss-gate and label correction”.

^[27]Chen et al., “A comprehensive study on self-supervised distillation for speaker representation learning”.

Self-Supervised Speaker Learning

Iterative Pseudo-Label Refinement

Two-stage iterative framework^[28,29,30]

Stage I: contrastive training. Stage II: iterative clustering and representation learning.

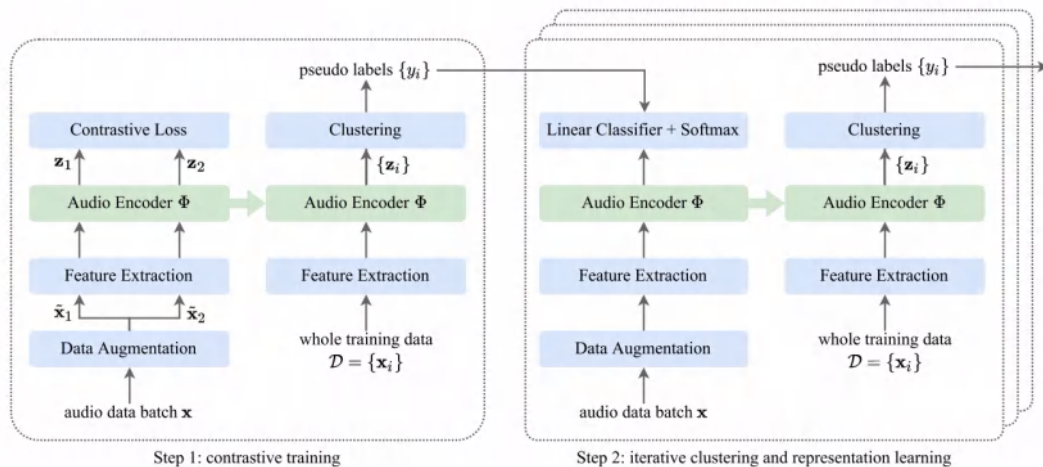


Figure 12: Iterative framework for SSL speaker verification

^[28] Cai, Wang, and Li, “An iterative framework for self-supervised deep speaker representation learning”.

^[29] Han, Chen, and Qian, “Self-Supervised Learning with Cluster-Aware-DINO for High-Performance Robust Speaker Verification”.

^[30] Tao et al., “Self-supervised speaker recognition with loss-gated learning”.

Section 4

Speaker Modeling Across Tasks



Cross-Task Speaker Modeling

Three Roles of Speaker Information

Role	Representative tasks	Required speaker evidence
Attribution	Diarization, speaker-attributed ASR	Separate speakers and track activity over time
Conditioning	TTS, voice conversion, personalization	Preserve identity, style, and controllable variation
Selection	Target-speaker extraction, target ASR, personal VAD	Match enrollment to the target inside a mixture

Central question

Which speaker information can be shared across tasks, and which evidence must be learned for the downstream output and loss?

Cross-Task Speaker Modeling

Shared Encoders and Task-Specific Evidence

Recognition

Maximize identity separation and suppress within-speaker nuisance variation.

Generation

Preserve timbre, prosody, emotion, and style for faithful or controllable synthesis.

Target processing

Distinguish the enrolled target from a small competing set inside the current mixture.

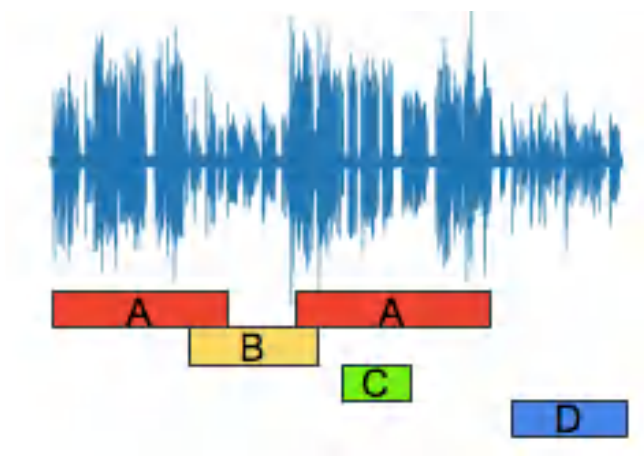
Design implication

A reusable encoder is valuable, but pooling, conditioning, adaptation, and evaluation must follow the task's required evidence.

Multi-Speaker Understanding

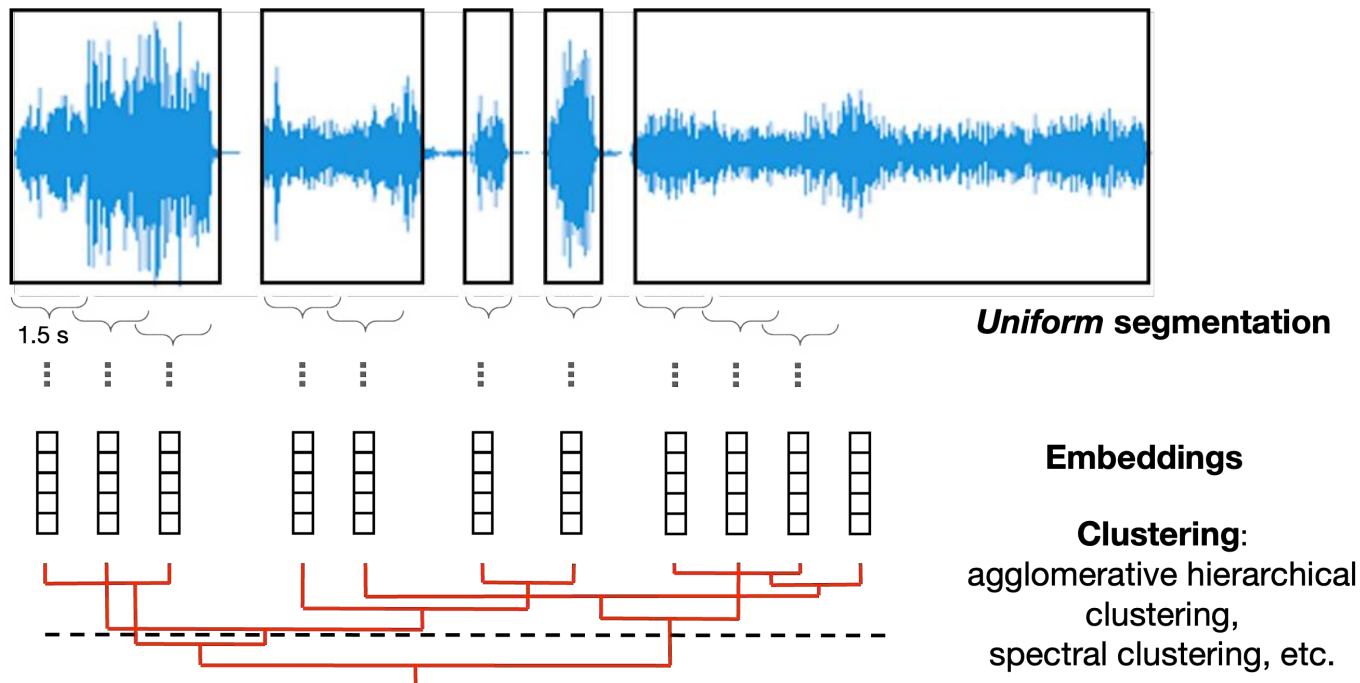
Speaker Diarization

- Speaker diarization is also known as speaker segmentation and clustering
- Answers the question: "Who spoke when?"
- In an audio, speakers A, B, C, D all speak; the goal is to:
 - Output the start and end time of each person's speech segments
- After diarization, per-speaker segments are easier for downstream processing



Multi-Speaker Understanding

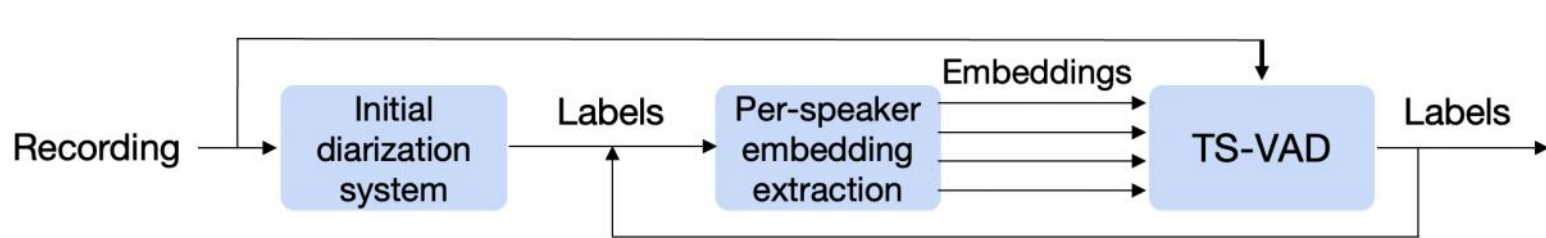
Clustering-Based Speaker Diarization^[31]



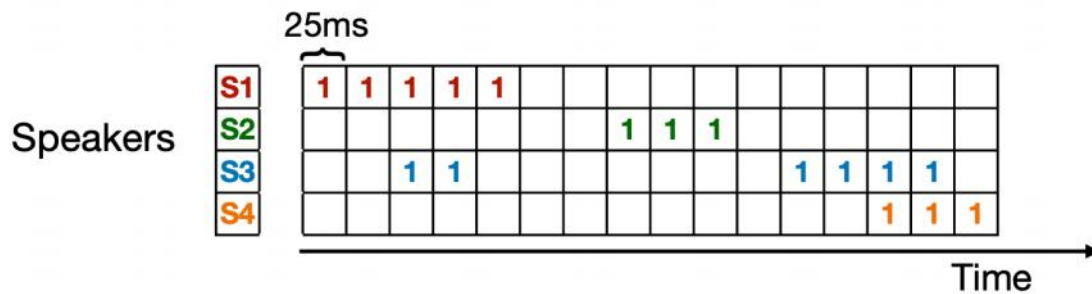
^[31]Thanks to Mireia Diez Sánchez for the figure

Multi-Speaker Understanding

TS-VAD-Based Speaker Diarization^[32]^[33]



Binary VAD decisions are made independently for each speaker per-frame with the same NN



^[32]Medennikov et al., "Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario".

^[33]Thanks to Mireia Diez Sánchez for the figure

Multi-Speaker Understanding

Speaker-Attributed ASR^[37]

- Multi-speaker ASR: in the same audio, recognize **content** from all speakers and **attribute** it (who says what)
- Typical scenarios: meetings, group discussions, call-center logs, classroom interactions, podcast interviews
- Related to the "cocktail party effect": extracting target info from overlapping speech

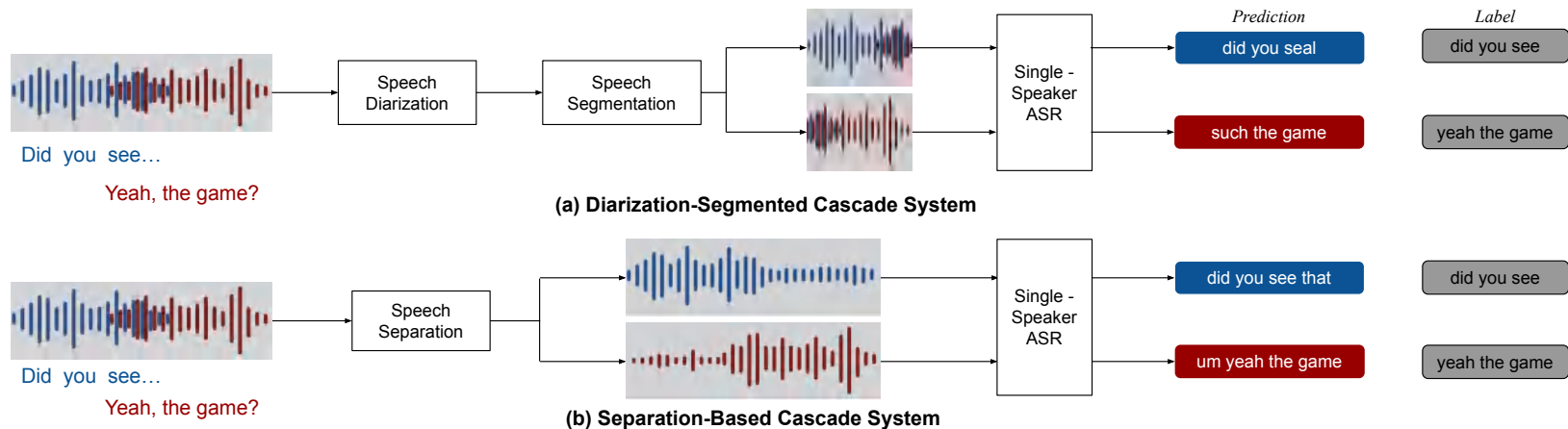
Challenges

- **Overlapped speech**: overlapping speakers cause mixing
- **Speaker attribution**: not only "what was said" but also "who said it"
- **Data scarcity**: scarce large-scale, fine-grained multi-speaker annotations
- **Coupled subtasks**: recognition, separation, segmentation, attribution, overlap/boundary detection

^[37]He and Whitehill, "Survey of End-to-End Multi-Speaker Automatic Speech Recognition for Monaural Audio".

Multi-Speaker Understanding

Speaker-Attributed ASR: Cascaded Systems^[38]

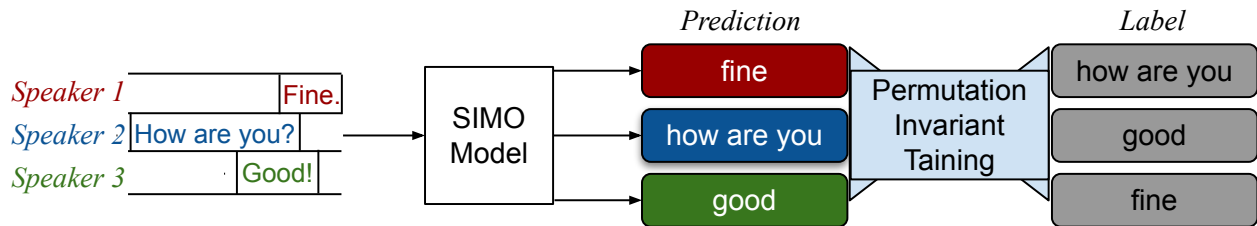


- Diarization-Segmented: first "speaker segmentation," then single-speaker ASR
- Separation-based: first "speech separation," then single-speaker ASR

^[38]He and Whitehill, "Survey of End-to-End Multi-Speaker Automatic Speech Recognition for Monaural Audio".

Multi-Speaker Understanding

Speaker-Attributed ASR: End-to-End Systems



(a) Single-Input Multiple-Output Framework

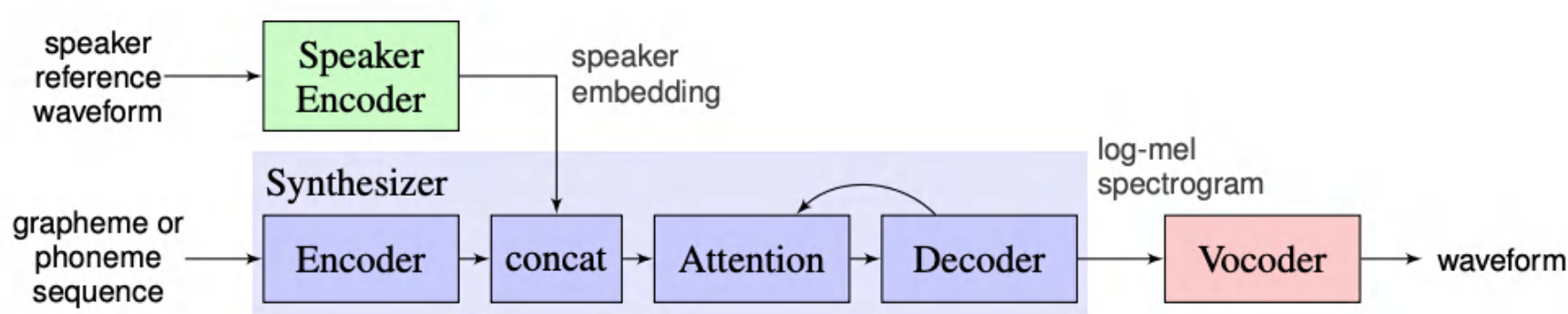


(b) Single-Input Single-Output Framework

- Directly map from **mixtures** to **speaker-attributed transcripts**, avoiding cascaded errors
- Support **joint optimization** of "who spoke" and "what was said"
- Two paradigms: **SIMO** (single-input multi-output) and **SISO** (single-input single-output)

Speaker-Conditioned Generation

Explicit Speaker Conditioning for Zero-Shot TTS^[39,40,41]



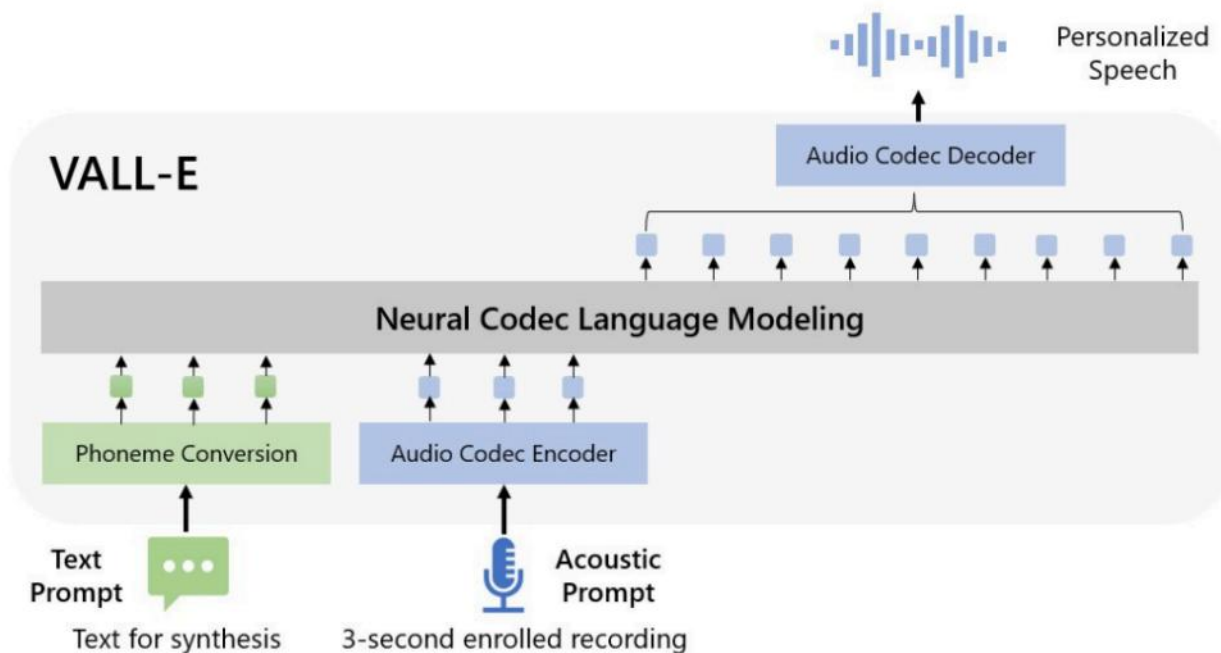
^[39] Jia et al., “Transfer learning from speaker verification to multispeaker text-to-speech synthesis”.

^[40] Casanova et al., “Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone”.

^[41] Wu et al., “Adaspeech 4: Adaptive text to speech in zero-shot scenarios”.

Speaker-Conditioned Generation

Implicit Speaker Modeling for Zero-Shot TTS^[42,43,44]



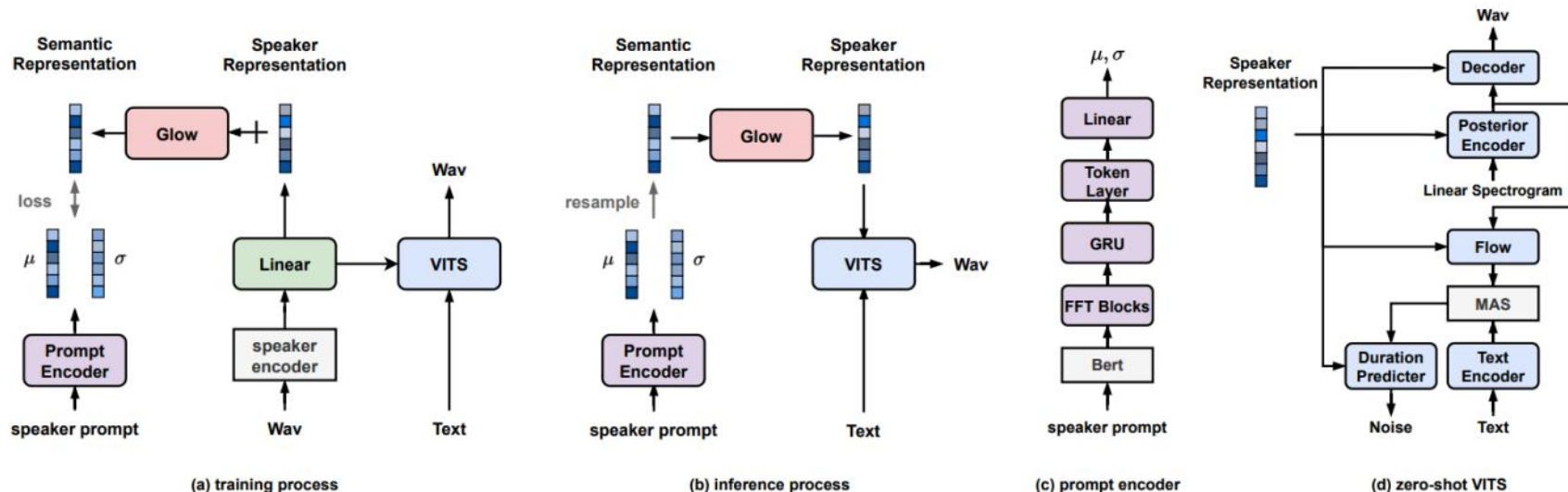
^[42]Wang et al., "Neural codec language models are zero-shot text to speech synthesizers".

^[43]Du et al., "UniCATS: A Unified Context-Aware Text-to-Speech Framework with Contextual VQ-Diffusion and Vocoding".

^[44]Le et al., "Voicebox: Text-guided multilingual universal speech generation at scale".

Speaker-Conditioned Generation

Controllability and Novel Voice Generation^[45,46,47,48]



^[45] Zhang et al., "PromptSpeaker: Speaker Generation Based on Text Descriptions".

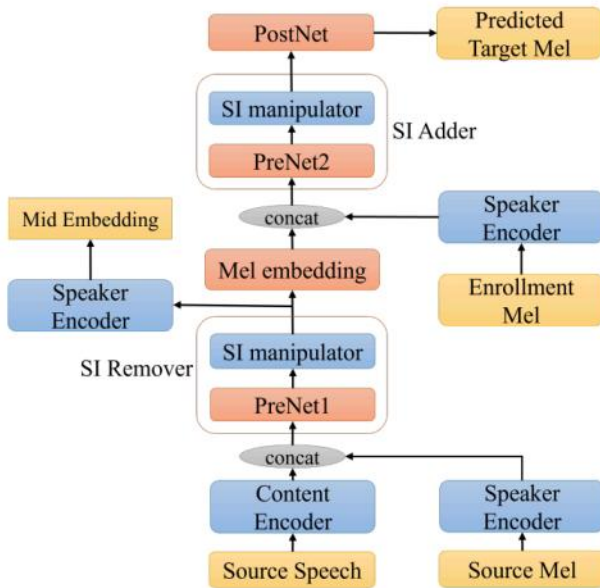
^[46] Stanton et al., "Speaker generation".

^[47] Shimizu et al., "PromptTTS++: Controlling Speaker Identity in Prompt-Based Text-to-Speech Using Natural Language Descriptions".

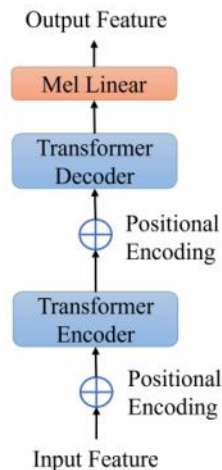
^[48] Bilinski et al., "Creating new voices using normalizing flows".

Speaker-Conditioned Generation

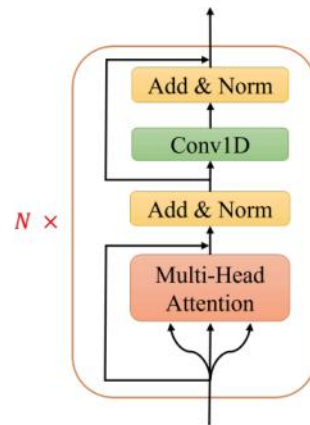
Explicit Speaker Modeling for Zero-Shot Voice Conversion^[49,50,51]



(a) Proposed System



(b) SI Manipulator



(c) Encoder and decoder architecture

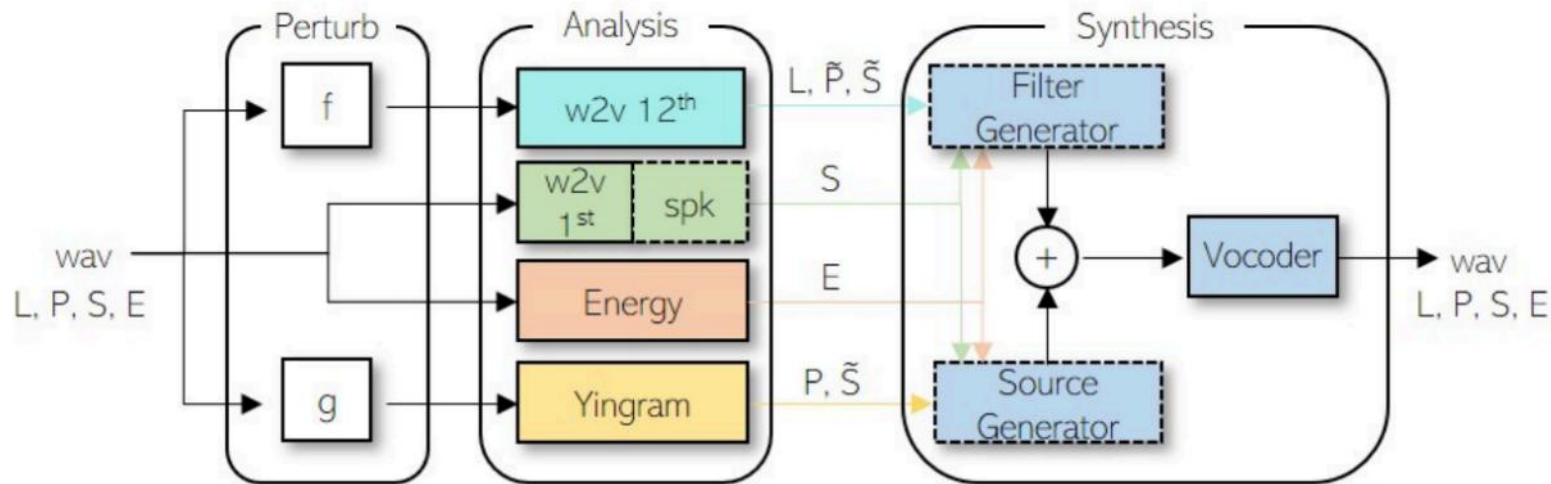
^[49] Zhang et al., "SIG-VC: A Speaker Information Guided Zero-Shot Voice Conversion System for Both Human Beings and Machines".

^[50] Chen and Duan, "ControlVC: Zero-Shot Voice Conversion with Time-Varying Controls on Pitch and Rhythm".

^[51] Hussain et al., "ACE-VC: Adaptive and Controllable Voice Conversion Using Explicitly Disentangled Self-Supervised Speech Representations".

Speaker-Conditioned Generation

Implicit Speaker Modeling for Zero-Shot Voice Conversion^[52,53,54]



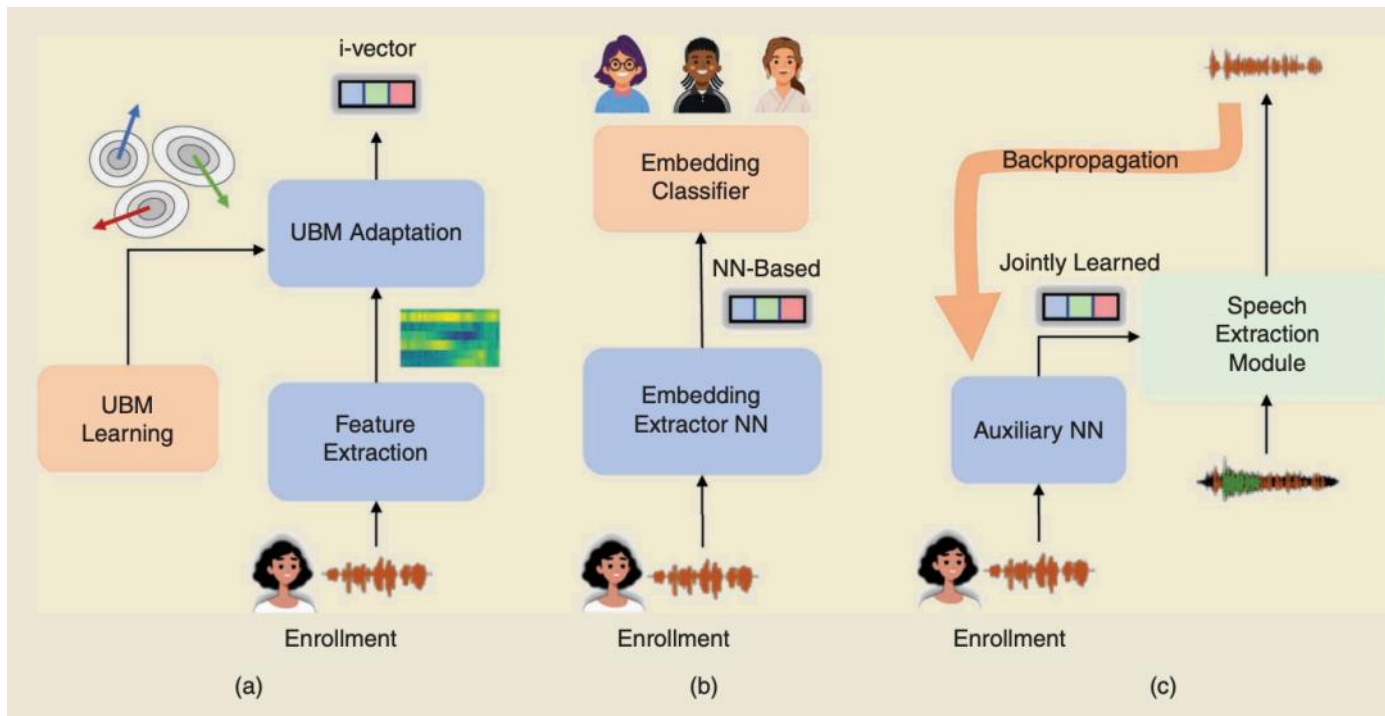
^[52]Choi et al., “Neural analysis and synthesis: Reconstructing speech from self-supervised representations”.

^[53]Wu, Chen, and Lee, “Vqvc+: One-shot voice conversion by vector quantization and u-net architecture”.

^[54]Wu and Lee, “One-shot voice conversion by vector quantization”.

Target-Speaker Speech Processing

Explicit Speaker Modeling for Target-Speaker Extraction^[55,56,57]



^[55] Zmolikova et al., “Neural Target Speech Extraction: An overview”.

^[56] Delcroix et al., “Single channel target speaker extraction and recognition with speaker beam”.

^[57] Ge et al., “Spex+: A complete time domain speaker extraction network”.

Target-Speaker Speech Processing

Implicit Speaker Modeling for Target-Speaker Extraction^[58,59]

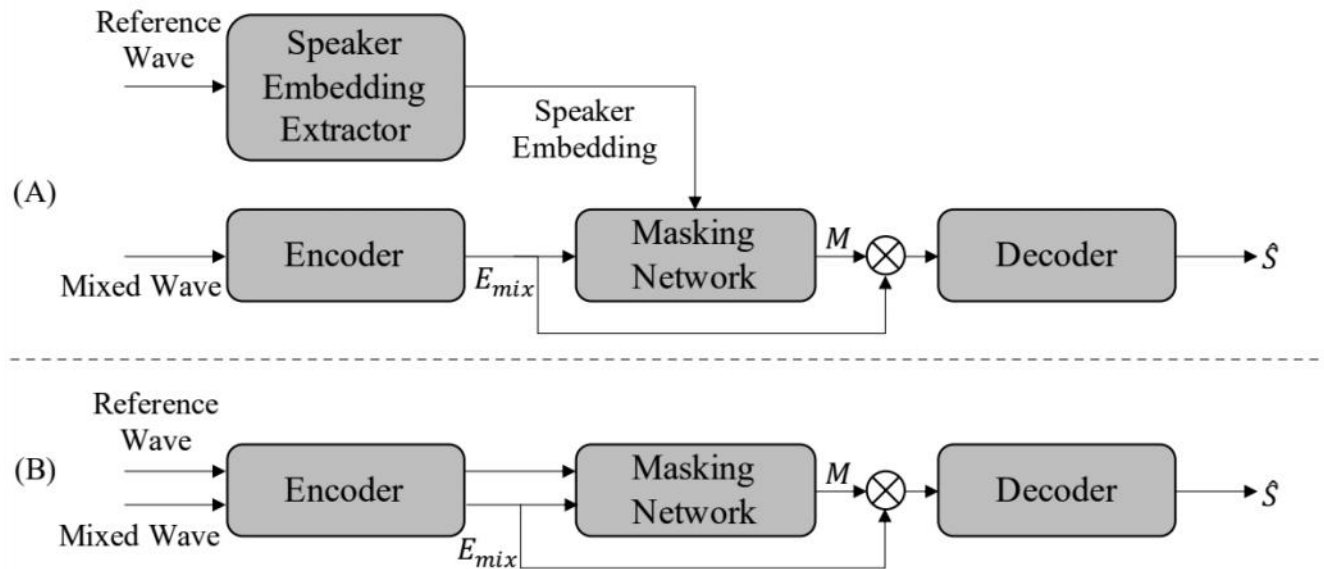


Figure 13: Embedding-conditioned (A) versus embedding-free (B) extraction

^[58]Zeng et al., "SEF-Net: Speaker Embedding Free Target Speaker Extraction Network".

^[59]Yang et al., "Target Speaker Extraction with Ultra-Short Reference Speech by VE-VE Framework".

Target-Speaker Speech Processing

Shared Conditioning and Task-Specific Outputs

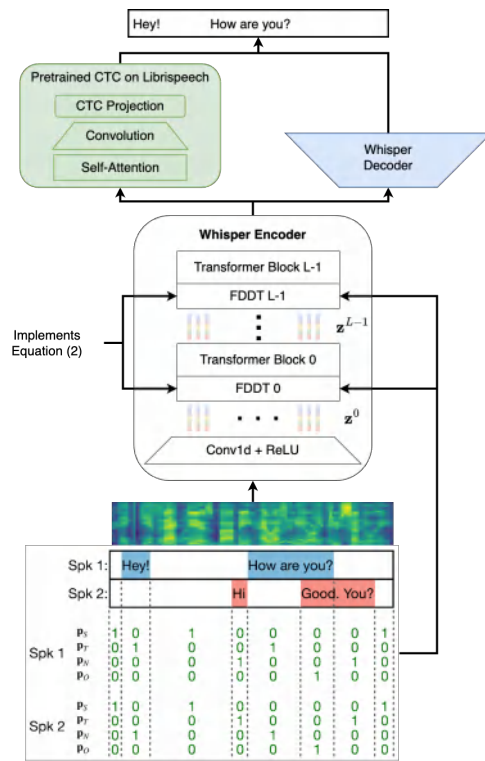
Task	Output	Speaker-conditioned decision
Target speaker extraction	Waveform or mask	Which acoustic energy belongs to the target?
Target-speaker ASR	Tokens	Which frames should drive recognition?
Personal VAD	Frame labels	Is this target, non-target, or non-speech?
TS-VAD diarization	Activity tracks	When is each enrolled speaker active?

Same enrollment, different error surface

Waveform distortion, WER, activity error, and diarization error reward different representations and thresholds.

Target-Speaker Speech Processing

Diarization-Conditioned Whisper^[60]



Frame-level conditioning

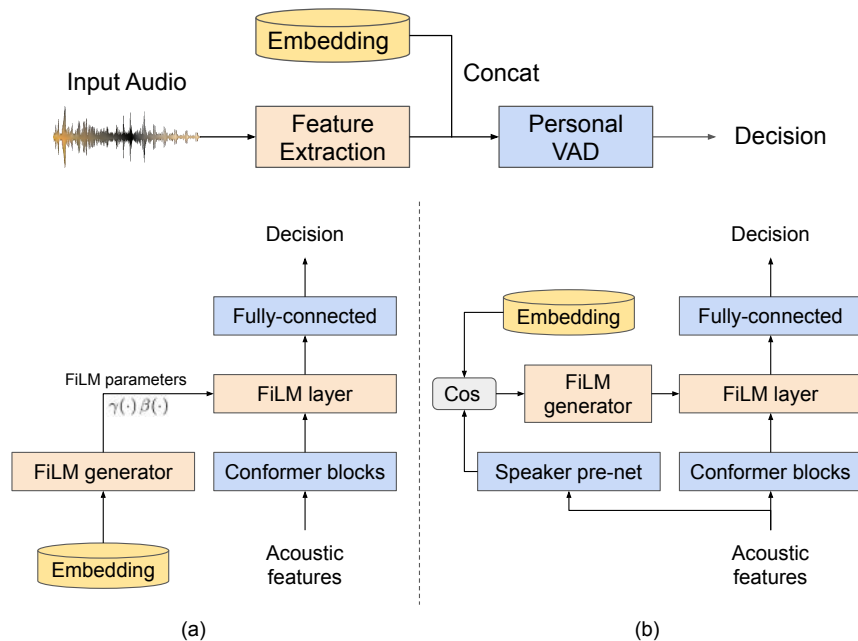
- Diarization supplies speaker-state probabilities
- Affine terms condition Whisper blocks
- Decode each selected speaker

Why it matters

The model avoids a separate waveform-extraction front end.

Target-Speaker Speech Processing

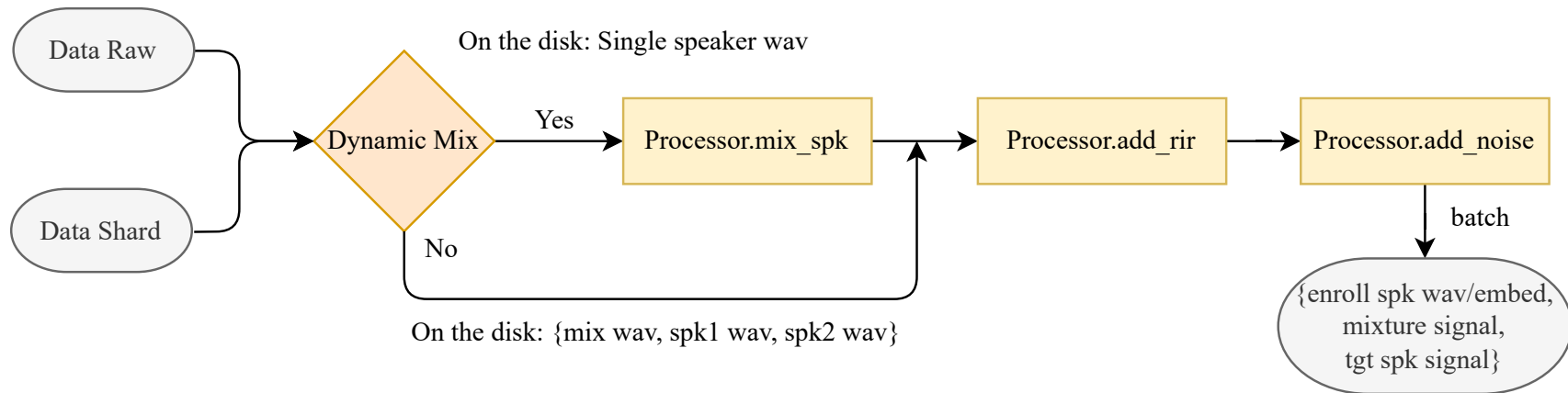
Streaming Personal VAD with Enrollment Conditioning^[61]



FiLM and speaker pre-nets turn one enrollment embedding into frame-level target/non-target decisions under on-device constraints.

Target-Speaker Speech Processing

WeSep: Reproducible Extraction Experiments^[62]



Why the data pipeline matters

Online mixture generation changes the speakers, overlap, reverberation, and noise seen during training. WeSep pairs this pipeline with speaker-conditioned extraction models.

^[62]Wang et al., "Wesep: A scalable and flexible toolkit towards generalizable target speaker extraction".

Task-Specific Representation Design

Speech Synthesis: Beyond Simple Embeddings

Challenge

Generate natural, expressive speech sounding like the target speaker

Key Requirements:

- Timbre accuracy
- Prosodic naturalness
- Emotional expressiveness
- Speaking style preservation

Evidence:

- Customized encoders > SV embeddings
- Prompt-based methods dominate zero-shot TTS

GAP

SV embeddings miss:

- Dynamics
- Prosodic details
- Emotional cues
- Style variations

Task-Specific Representation Design

Speech Synthesis: Embedding Comparison^[63]

Findings by Adriana STAN et al. (2023)

1. Embedding choice does not affect the learning process

- Networks adapt to speaker conditioning regardless of embedding choice
- Similar synthesis quality can be achieved

2. Speaker leakage is inevitable

- Core modules contain speaker information under standard training
- Simple conditioning cannot ensure perfect disentanglement

3. Inconsistency under zero-conditioning

- Core networks learn similar representations
- Speaker identity is unstable in zero-conditioning

^[63]Stan and O'Mahony, "An analysis on the effects of speaker embedding choice in non auto-regressive TTS".

Task-Specific Representation Design

Voice Conversion: Disentanglement Challenges

Fundamental Issue

Speech = Speaker + Content + Prosody + Accent + Emotion + ...

Current methods:

- Adversarial training
- Gradient reversal layer
- Multi-encoder architectures
- Self-supervised representations

Challenges:

- Perfect disentanglement is impossible
- Residual speaker information
- Content-speaker entanglement
- Prosody control complexity

Open Question

Can we ever achieve perfect disentanglement?

Task-Specific Representation Design

Target Speaker Extraction: Rise of Non-embedding Methods

Paradigm Shift

From pre-computed embeddings to adaptive, context-aware modeling

Traditional:

- Pretrained speaker embeddings
- Fixed representations
- Limited context awareness
- Performance bottlenecks

Recent:

- USEF-TSE (embedding-free)
- Attention mechanisms
- Multi-level representations
- SSL-based features

Key Insight

Direct acoustic matching can outperform abstract embeddings

Section 5

Multi-Modal Speaker Modeling

Multi-Modal Speaker Modeling

Complementary Audio and Visual Cues^[64]

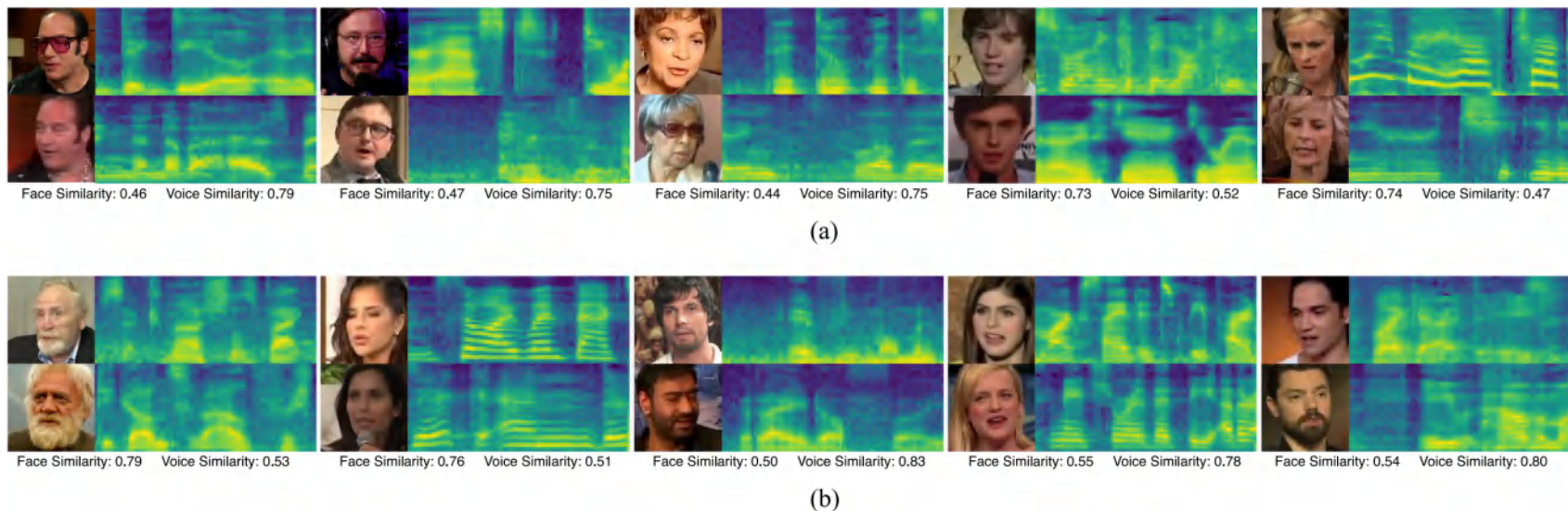


Figure 14: Speaker similarity from audio or visual information

- Upper part shows similarity to the same person
- Lower part shows similarity between different persons

^[64]Qian, Chen, and Wang, "Audio-visual deep neural network for robust person verification".

Multi-Modal Speaker Modeling

Audio-Visual Fusion Levels^[65]

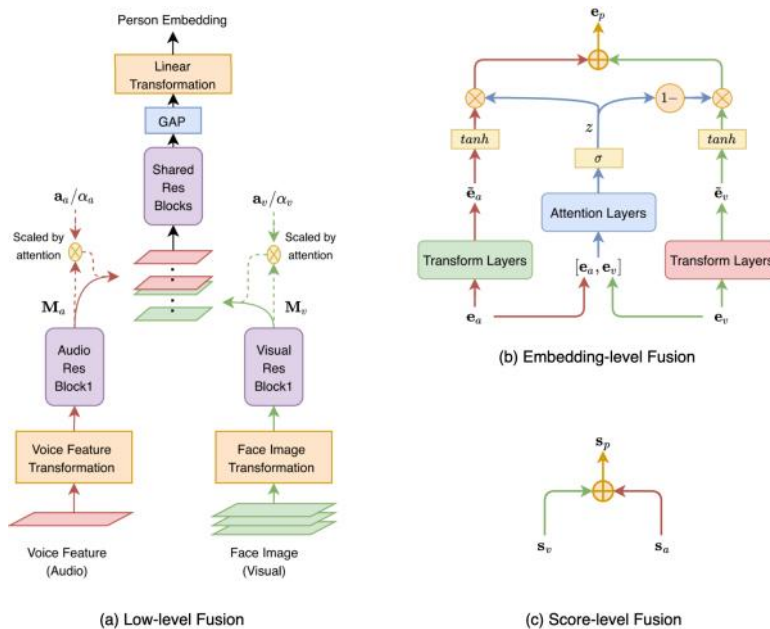


Figure 15: Different levels of audio-visual fusion

Multi-Modal Speaker Modeling

Audio-Visual to Single-Modality Distillation^[66]

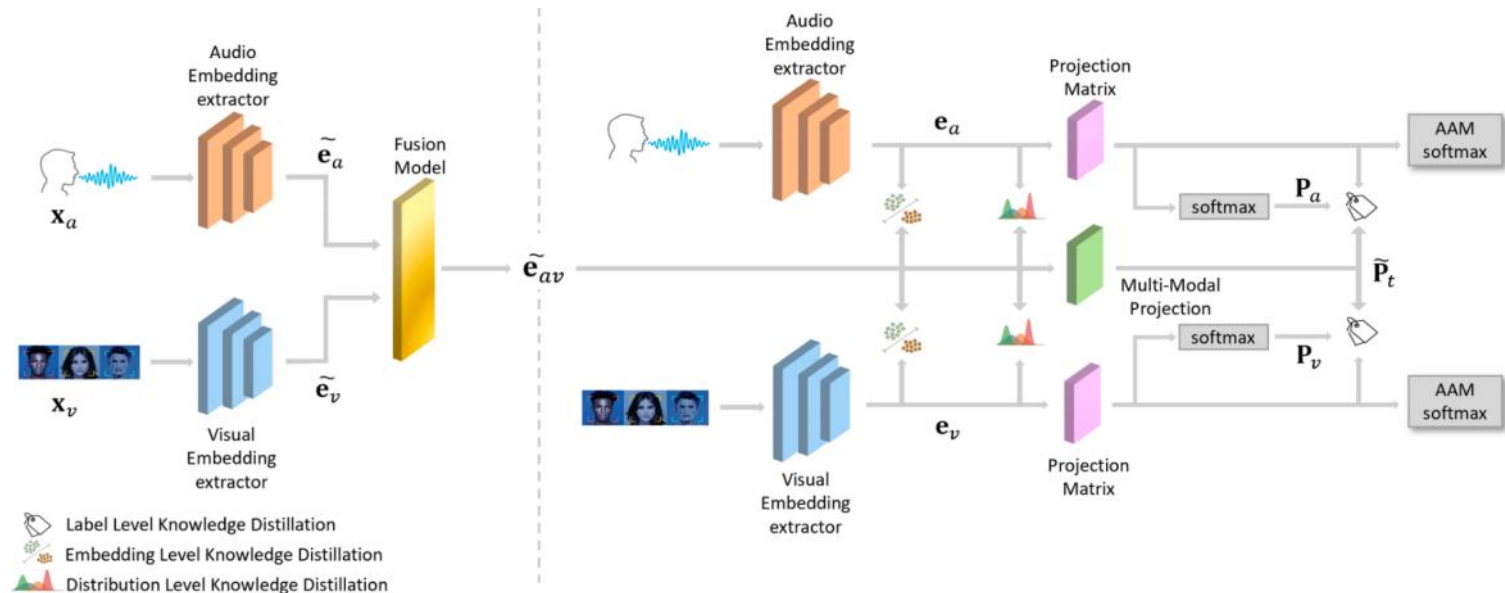


Figure 16: Audio-visual to single-modality distillation

^[66]Zhang, Chen, and Qian, "Knowledge Distillation from Multi-Modality to Single-Modality for Person Verification".

Multi-Modal Speaker Modeling

Visual-to-Audio Knowledge Distillation^[67]

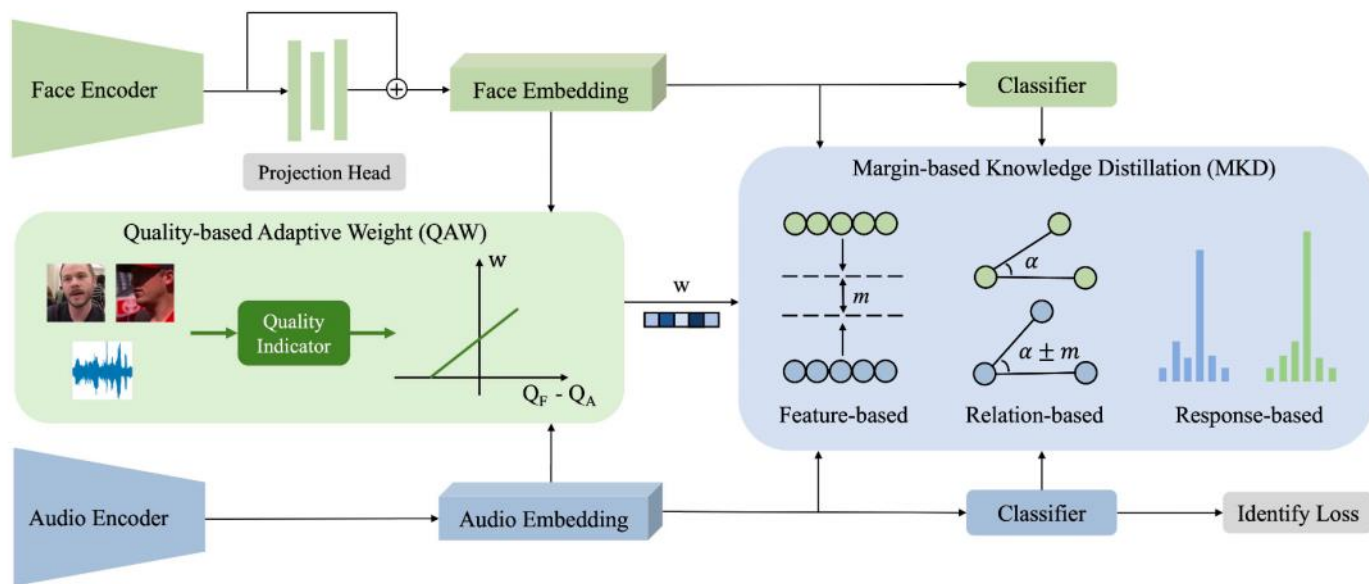


Figure 17: Visual-to-audio knowledge distillation

Section 6

Frontiers and Resources

Foundation Models for Speaker Tasks

Audio Language Models and Speaker Verification^[68]

What transfers

- General audio understanding and instruction following
- Speaker attributes such as age, gender, and accent
- Joint reasoning about identity and linguistic content

What still requires adaptation

- Fine-grained identity discrimination
- Calibration across devices and durations
- Hard-negative sampling for similar speakers

Cross-task implication

Foundation models broaden the interface, but speaker identity remains a specialized decision that benefits from task-aware training.

^[68]Ren et al., “Can Audio Large Language Models Verify Speaker Identity?”

Foundation Models for Speaker Tasks

Fine-Tuning Effects Across Conditions^[69]

Model	Gender	Lang.	Age	Device	Dialog	< 2s
Kimi zero-shot	70.20	68.40	63.40	52.67	55.00	53.70
Kimi fine-tuned	95.07	97.00	92.40	88.20	89.00	80.90
ECAPA-TDNN	99.33	99.27	94.13	94.67	93.00	78.80

- Fine-tuning closes most of the zero-shot gap across speaker and recording conditions.
- The conventional speaker encoder remains stronger on most matched conditions.
- The audio language model is competitive on short speech, suggesting complementary robustness rather than universal superiority.

^[69]Ren et al., “Can Audio Large Language Models Verify Speaker Identity?”

Foundation Models for Speaker Tasks

SpeakerLLM: Speaker Understanding and Verification Reasoning^[70]

Frozen ReDimNet-B3 and a hierarchical tokenizer condition Qwen2.5-1.5B-Instruct.

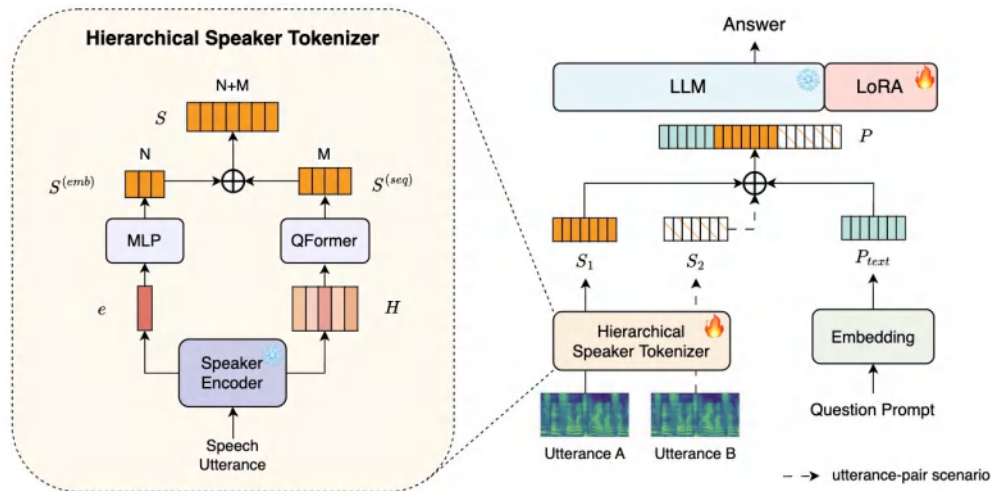


Figure 18: Hierarchical speaker tokens condition the language model

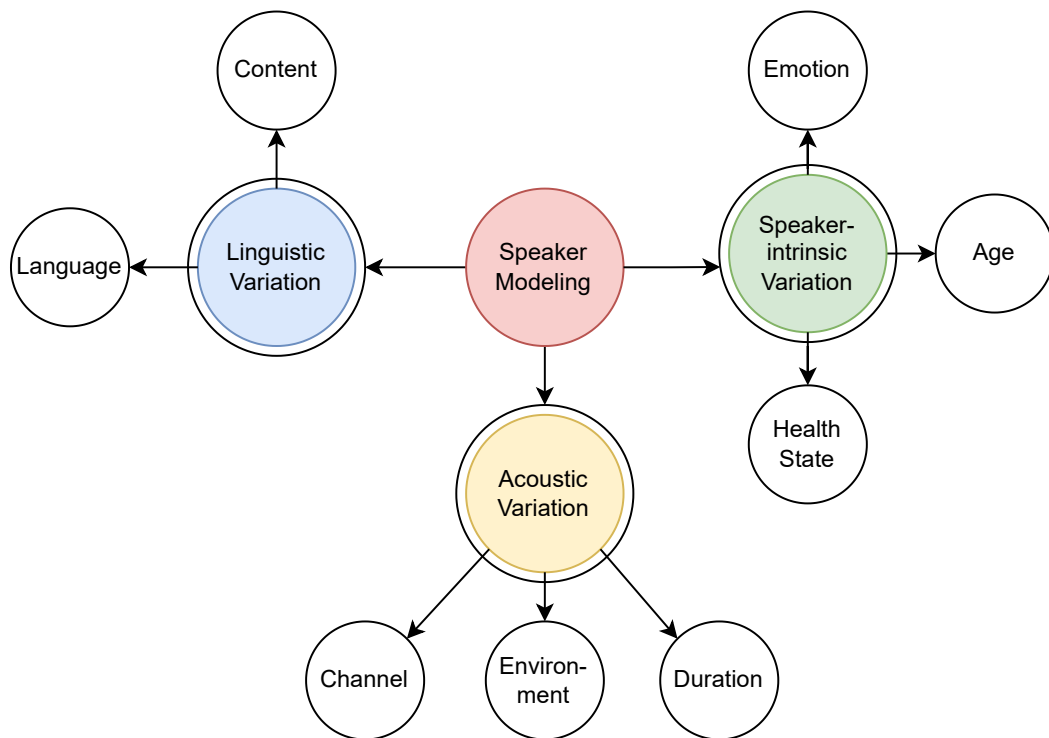
Takeaway: identity summaries and frame-level cues provide complementary evidence.

SpeakerLLM-Base: 96.1% generated-verdict accuracy on VoxCeleb1-O. This metric is not EER or a calibrated verification score.

^[70]Nam et al., SpeakerLLM: A Speaker-Specialized Audio-LLM for Speaker Understanding and Verification Reasoning.

Robust Speaker Representations

Linguistic, Acoustic, and Speaker-Intrinsic Variation



Evaluation and Tuning

Protocol, Model Selection, and Operating Decisions

1. **Protocol:** fix speakers, sessions, duration, language, channel, and trials.
2. **Model selection:** use development data for the checkpoint and scoring setup.
3. **Decision:** report DET/EER/minDCF, calibration, and the target operating point.

A common source of optimistic results

Selecting a checkpoint or threshold on the test condition measures condition-specific tuning rather than transfer.

Resources

Open-Source Code and Further Reading

WeSpeaker

Speaker embedding training, verification, and diarization

github.com/wenet-e2e/wespeaker

WeSep

Target-speaker extraction with reusable speaker models

github.com/wenet-e2e/wesep

Papers: WeSpeaker, Speech Communication 2024^[72]; WeSep, INTERSPEECH 2024^[73].

^[72]Wang, Chen, Han, et al., “Advancing Speaker Embedding Learning: WeSpeaker Toolkit for Research and Production”.

^[73]Wang et al., “Wesep: A scalable and flexible toolkit towards generalizable target speaker extraction”.

Resources

OpenSVBench: Cross-Scenario Speaker Verification

SPEAKER VERIFICATION BENCHMARK

OpenSVBench Leaderboard

Cross-scenario evaluation of speaker verification systems on a fixed public core. The global ranking uses source-balanced scenario-macro EER; lower is better.

18 ranked systems

11 scenarios

26 trial sets

16 source datasets

Submit a system

Browse scenarios

FIXED-CORE RANKING

Global ranking

Only reviewed, completed submissions with full current-core coverage are included.

Current leader: **wespeaker/w2vbert2**

RANK	SYSTEM	EER ↓ Scenario macro	MINDCF ↓ Scenario macro	STRONGER SCENARIOS	WEAKER SCENARIOS
#1	wespeaker/w2vbert2 vb2-w2vbert	9.9604	0.360767	Noise/reverb Speaking style	Genre shift Channel/device
#2	palabraai/redimnet2-b6 redimnet2	10.5277	0.419432	Overlap Aging	Genre shift In-the-wild
#3	wespeaker/samresnet100 vb2-sam100	10.5780	0.390239	Aging Noise/reverb	Accent/dialect Channel/device

opensvbench.xyz

89

Resources

SpeakerCard-1M: Evidence-Grounded Speaker Supervision^[74]

10.2k speakers 56.7k Speaker Cards 1.78M captions

Bilingual EN/ZH annotations built on VoxCeleb1/2 and CN-Celeb1/2.

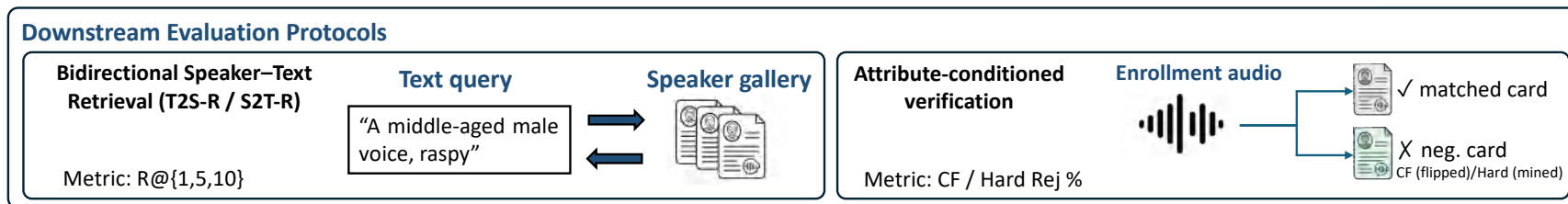


Figure 19: Cross-modal evaluation protocols (detail of the paper’s Fig. 1)

- **Tool-first, LLM-last:** probes extract evidence, then an LLM renders it as text.
- **Trait-state separation:** stable speaker traits differ from utterance states.

junyipeng00.github.io/SpeakerCard-1M-page

^[74]Peng et al., SpeakerCard-1M: An Evidence-Grounded Corpus for In-the-Wild Speaker Verification.

Trends in Speaker Modeling

What the Methods and Applications Tell Us

Axis	Earlier focus		Current direction
Encoder	Deep models, one accuracy score	→	Efficient models on a cost–accuracy frontier
Learning	Speaker labels from scratch	→	Speaker SSL and speech-model transfer
Adaptation	Full model retraining	→	Layer selection and parameter-efficient tuning
Use	Verification as the endpoint	→	Task-specific conditioning and generation
Robustness	Clean, matched evaluation	→	Cross-scenario tests for channel, language, duration, style, and noise
Evaluation	One EER on one corpus	→	Calibration, subgroup results, and reproducible protocols

Overall trend

Robustness remains a central goal. Speaker models must preserve identity evidence across realistic variation, and evaluation must show where each system remains reliable.

Chapter 2 of 3

Harnessing System Excellence with Frozen Backbone Speaker Models



Quan Wang Google DeepMind

About the lecturer



2014 - 2015
Machine Learning Scientist

Optical character recognition
"Alexa" wakeword detection



Google DeepMind

2024 - Today
Software Engineer Manager

Gemini Audio

2006 - 2014
Student

Compute vision
Medical image processing



2015 - 2024

Software Engineer | Tech Lead Manager

"Hey Google" hotword & voice match
Speaker diarization
Language recognition and multilingual ASR



What this talk is about

TUTORIAL TITLE

Speaker Modeling
beyond Recognition

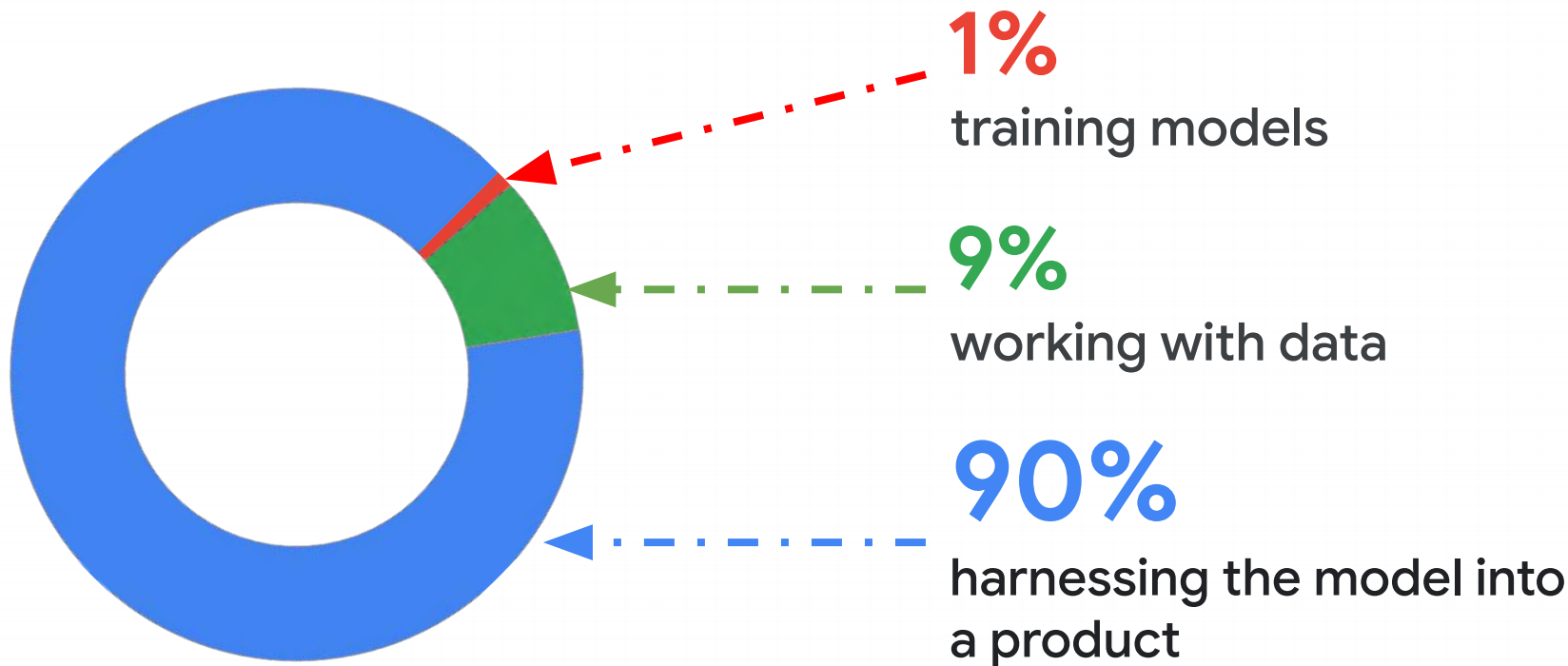
FOCUS OF THIS CHAPTER

Speaker Recognition
beyond Modeling

KEY TAKEAWAY

A production-ready speaker recognition or diarization system is much more than just a good backbone model.

How I spend my time at Google / DeepMind



Agenda

01

Intro to Harness Engineering

02

Harnessing Speaker Models

03

System Routing & Triage

04

Version Control, Privacy and Security

05

Enrollment Audio as Memory

06

Harnessing Constraints in Speaker Diarization

07

Multi System Post-Processing

08

Harnessing Longform Diarization

09

Closing Remarks

Section 1

Intro to Harness Engineering



What is harness engineering?

The concept of harness engineering went viral in early 2026, introducing a foundational paradigm for AI development.

$$\text{Agent} = \text{Model} + \text{Harness}$$

Backbone LLM

Upper Bound

Determines the theoretical maximum capability, core reasoning capacity, and peak potential of the system excellence.

The Harness

Lower Bound

Determines the practical baseline reliability, handling constraints, safety scaffolding, and minimum guaranteed performance.

The evolution of agent paradigm

Timeline	Paradigm	Focus
Nov 2022 ~ Jun 2025	Prompt Engineering	• Input to a single inference call • Guided reasoning, few-shot examples
Jul 2025 ~ Jan 2026	Context Engineering	• Dynamic pruning of conversation history • Manipulating the active memory limits • Managing RAG
Feb 2026 ~ Now	Harness Engineering	• End-to-end task completion • Tool use and sandbox execution • Validation and self-correction • Multi-model/agent orchestration

Section 2

Harnessing Speaker Models



Backbone speaker models

Speaker recognition / diarization system = Backbone model + Harness

In an AI Agent, the Backbone is a LLM

Core Capability:

- Natural language understanding & generation
- Serves as the core engine for planning, reasoning, and action

Speaker Recognition / Diarization

Core Capability:

- Voice characteristic discrimination
- Ability to precisely distinguish and separate different speakers from audio

Speaker systems

Backbone Models

Core speech representation & diarization engines:

Speaker Embedding Models

- i-vectors
- ECAPA-TDNN
- x-vectors
- Google d-vectors

End-to-End Neural Diarization (EEND)

Multimodal Large Language Model (MLLM)

The Harness

The orchestration & execution framework:

Orchestration

- When to call each model
- Multi-model orchestration

Inputs & Outputs

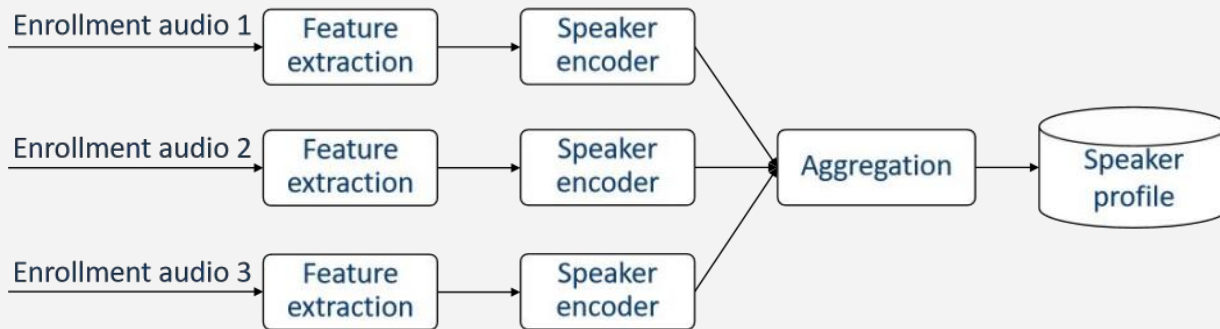
- Inputs to each model
- Processing model output

Peripheral Models

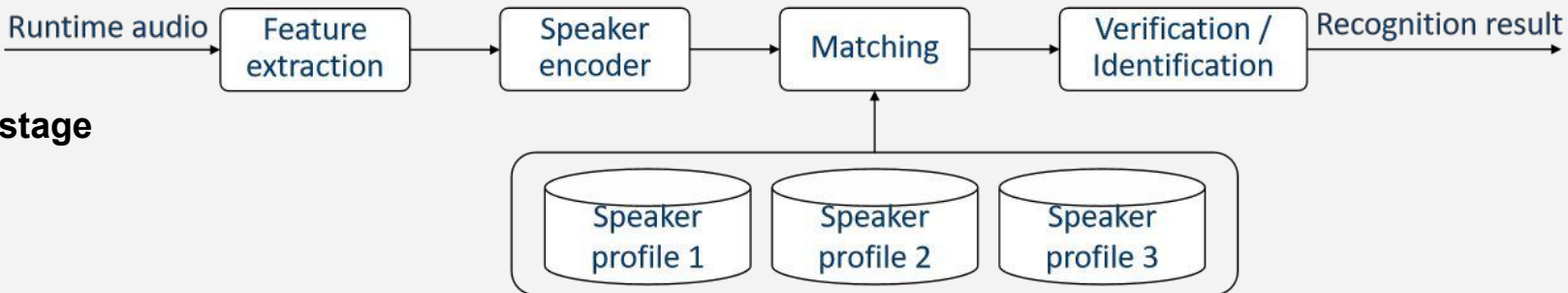
- Effective integration of supporting tools

Simplified speaker recognition workflow

Enrollment stage



Runtime stage

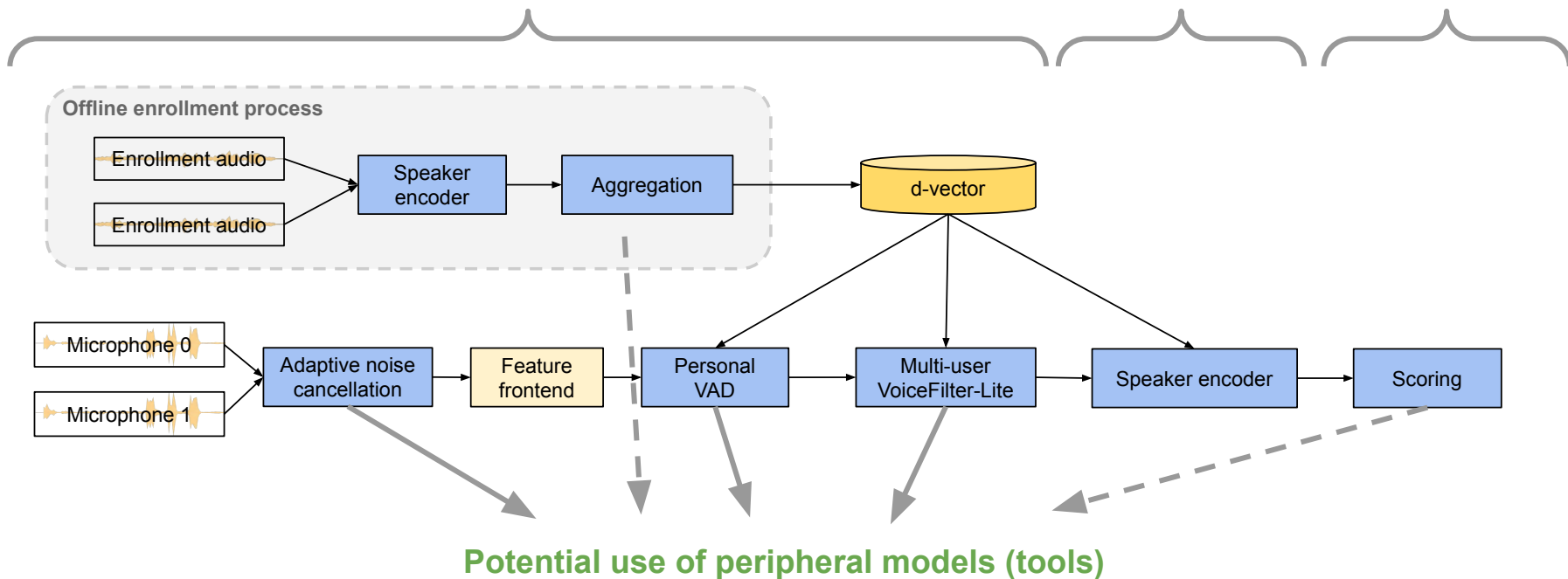


Production speaker recognition workflow

Harnessing the inputs

Backbone
model

Harnessing
the outputs

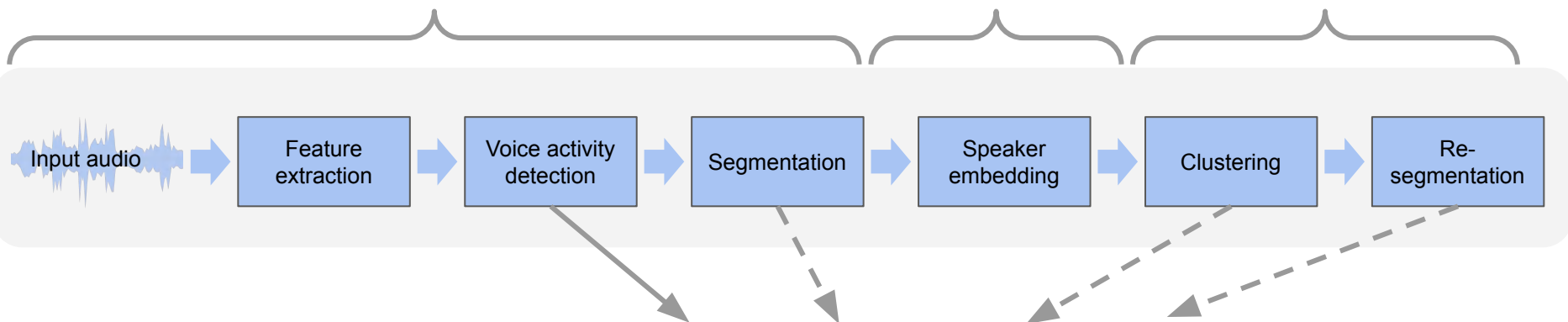


Conventional diarization workflow

Harnessing the inputs

Backbone
model

Harnessing the outputs



Potential use of peripheral models (tools)

Harnessing: AI agent vs speaker recognition

Purpose	AI Agent	Speaker recognition
Cost saving	Routing models with different sizes [ref]	TD + TI model triage [ref]
Consistency & robustness	Self-consistency decoding [ref]	Test time augmentation
Cache compression	H2O [ref] , Scissorhands [ref]	Attentive scoring [ref]
Privacy & security	Privacy-preserving fine-tuning [ref]	Speaker embedding encryption [ref]
Version control	Context branching [ref]	Dual-version serving [ref]

Harnessing: AI agent vs speaker diarization

Purpose	AI Agent	Speaker diarization
Constraining output	Guided generation [ref]	Constrain by speaker turn [ref] or semantics [ref]
Error correction	Reflexion [ref]	DiarizationLM [ref] , etc.
Multi-system fusion	LLM-Blender [ref]	DOVER [ref] , DOVER-Lap [ref]
Longform streaming	Full-duplex	Dynamic compression [ref] , WEEND [ref]
Multi-session	Dynamic memory	EEND-VC [ref] , Speaker prompt cache [ref]

Section 3

System Routing & Triage



LLM Routing

Routing queries to the LLM with **appropriate size** can significantly reduce **latency and cost**.

APPROACH 01

Pre-generation

Intent classification: The router evaluates the query's complexity or intent before sending it to any LLM, matching it directly to the right-sized model.

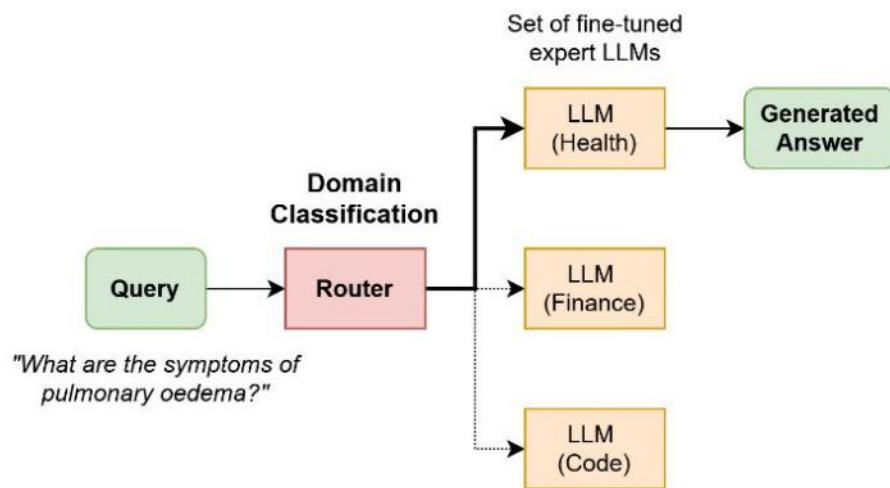
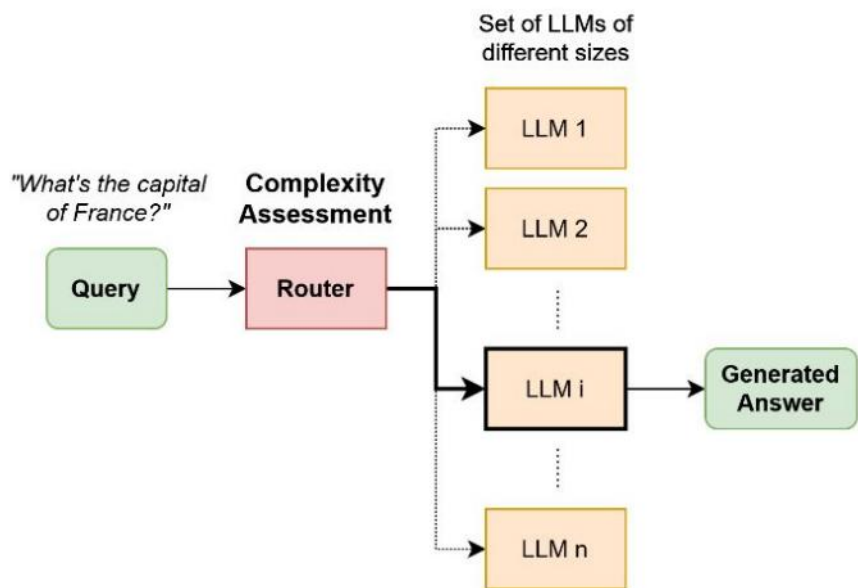
APPROACH 02

Post-generation

LLM cascade: Queries are sequentially passed through increasingly larger models. Output quality is evaluated post-generation to decide if a larger model is needed.

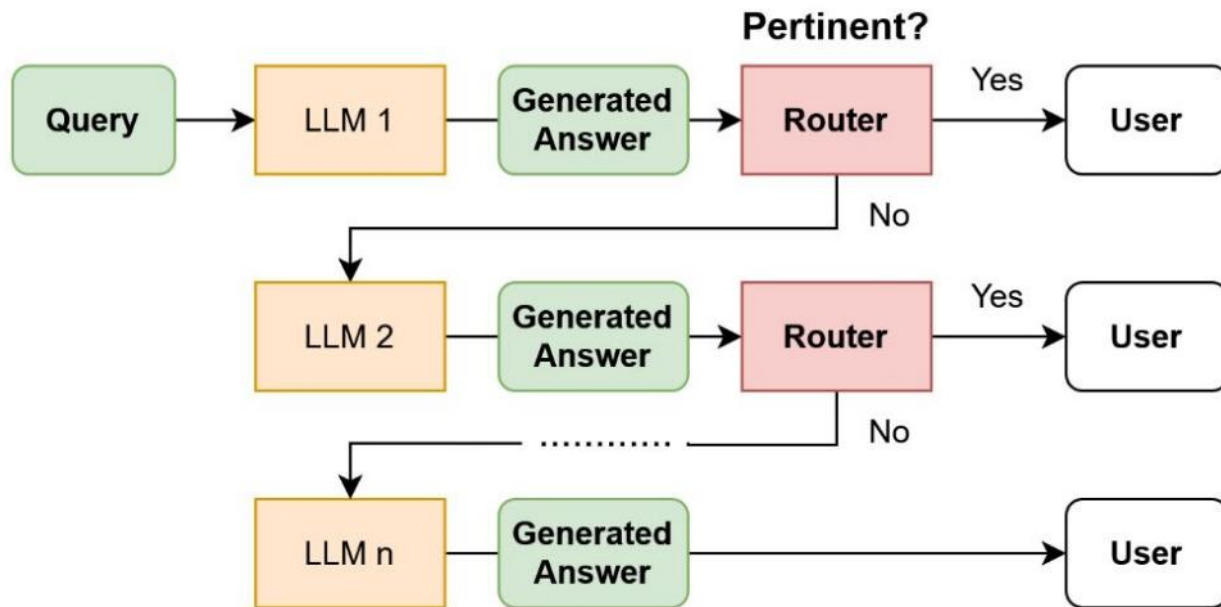
LLM routing: Pre-generation

- Cost and latency of router:
 - LLM intent classifier > embedding similarity > regular expression / keyword matching



LLM routing: Post-generation

- The router is basically a scoring function



When to run a speaker recognition model

- **Topic 1: Racing against ASR**

- Streaming speaker recognition results to fetch user profile

- **Topic 2: Cost saving**

- Triaging text-dependent (TD) and text-independent (TI) models based on confidence

- **Topic 3: Handling missing signal**

- Fusion of embeddings from TD or TI models

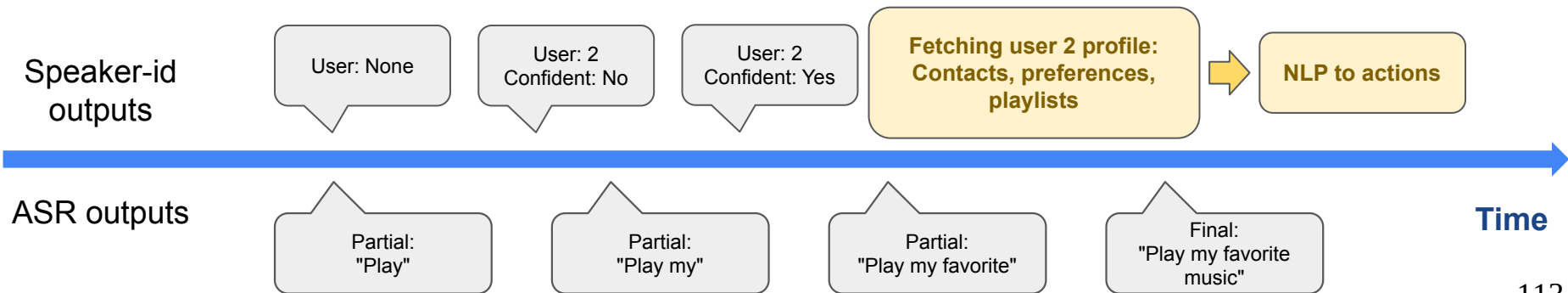
Racing against ASR

- **Standalone speaker-id systems:**

- No need for streaming.
- A simple **Unary API**:
 - `result = f([user_1.enroll, ..., user_n.enroll], test.wav)`

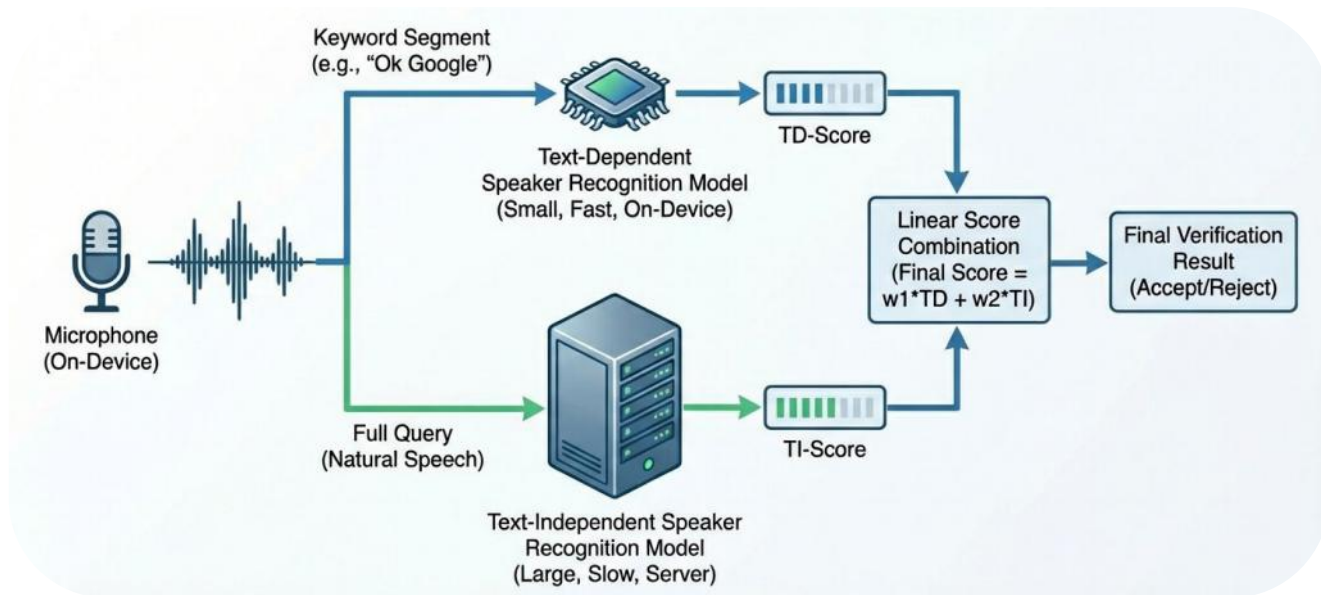
- **Speaker-id as part of voice AI:**

- Need to produce streaming results **per frame**
- Once we have a **confident** result, we use it to fetch user profile for ASR biasing and NLP



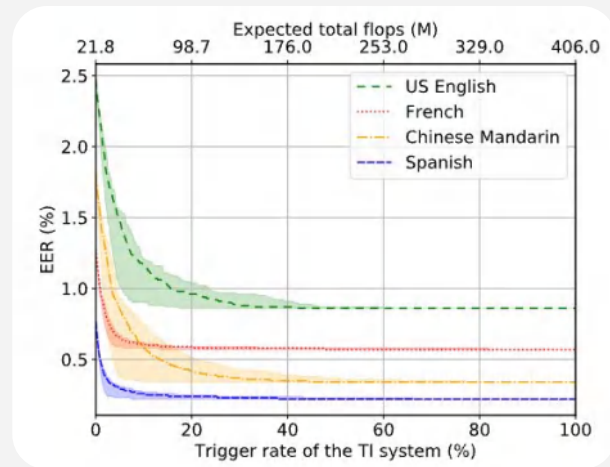
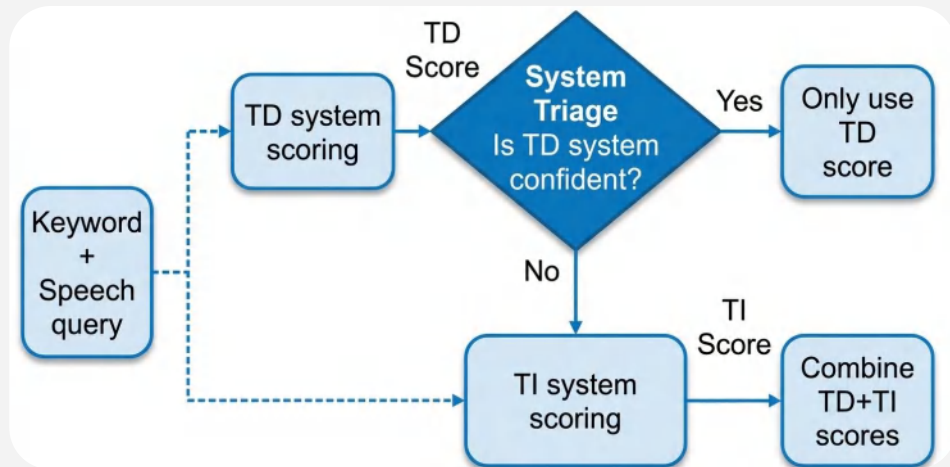
Cost saving

- Speaker verification score is usually a linear combination of two models:
 - **Text-dependent (TD)**: Runs on keyword part; small size; fast; on-device
 - **Text-independent (TI)**: Runs on full query; large size; slower; usually on a server



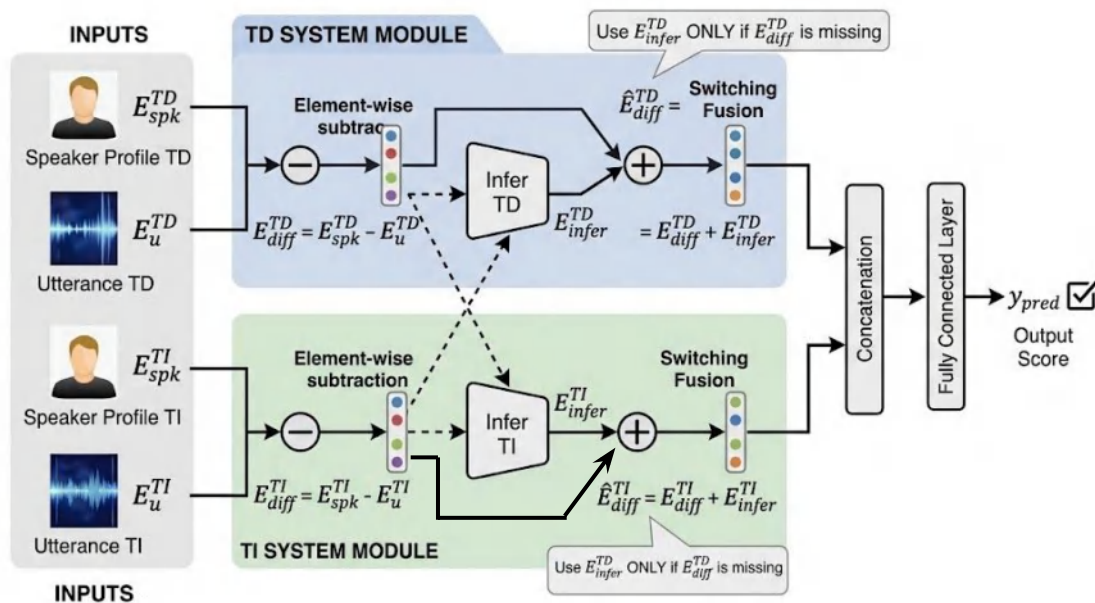
Cost saving

- When we have million+ QPS:
 - Server-side TI becomes **bottleneck** for compute and latency
 - Solution: **Only triage to TI when TD has low confidence** (less than 40% of queries!)



Handling missing signal

- **FOEnet**: Fusion of TD and TI into one embedding
- Either can be **absent**: Backup embedding inferred from the other system



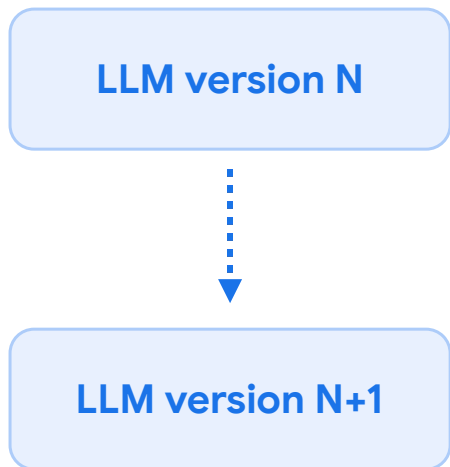
Section 4

Version Control, Privacy and Security



Version control of LLM

- LLM version update is often not just a simple checkpoint replacement

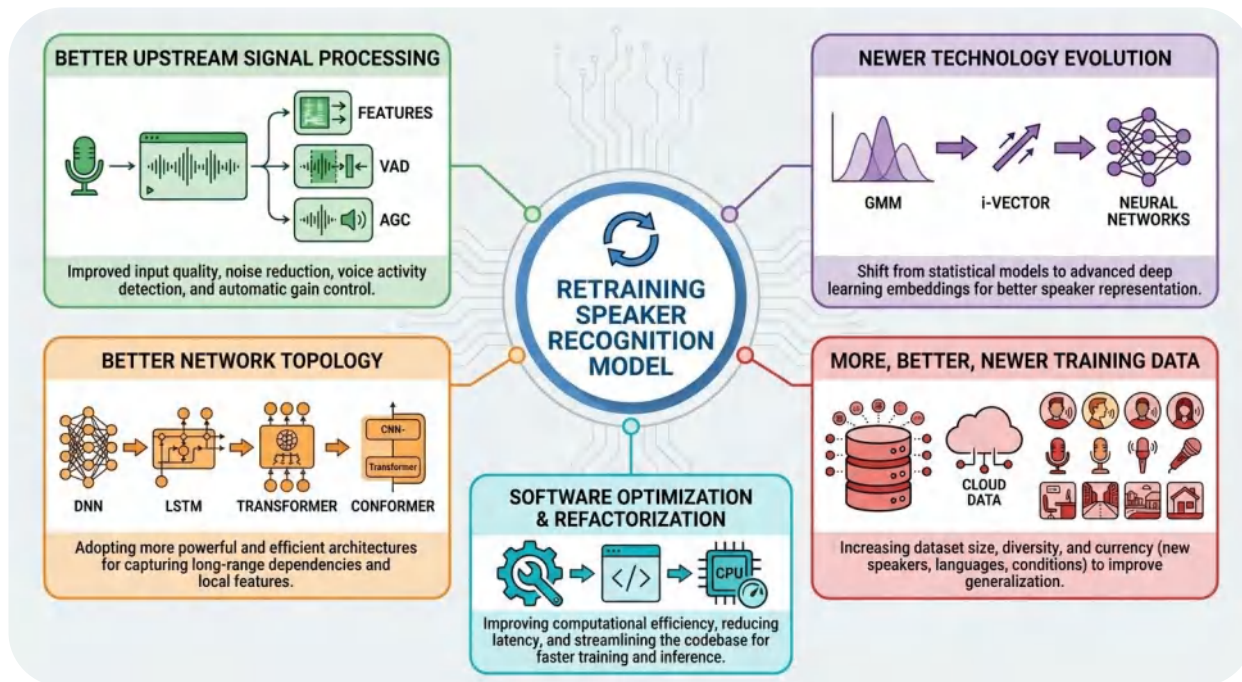


Updates needed:

- **Text tokenizer**
- **Image/audio tokenizer**
- **Special tokens** (delimiter for user/model turns)
- **System prompt**
- **Instructions** via automatic prompt optimization (APO)
- **Function calling syntax/schema**
- **Context length limit**

Version control of speaker recognition

- We often need to **retrain** the speaker recognition model:

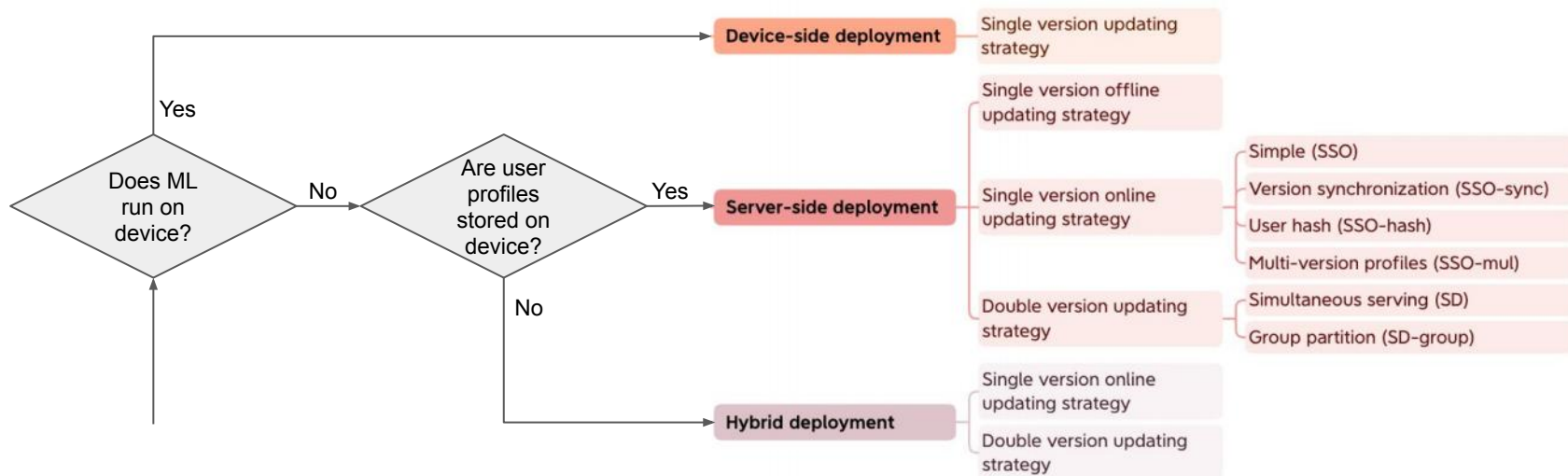


Version control of speaker recognition

- Updating the model may cause **version mismatch**



- Different version control strategies for different types of deployment



Simulator tool: SpeakerVerSim

Overview

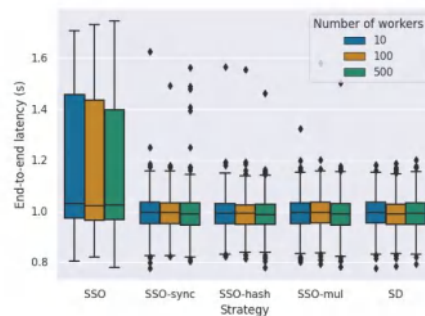
A dedicated simulator for evaluating and analyzing different speaker recognition version control strategies.

- Simple configuration via YAML files
- Includes pre-built common strategy templates

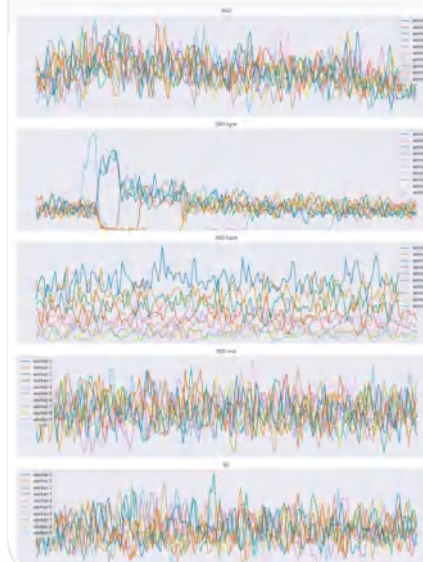
GET STARTED

github.com/wq2012/SpeakerVerSim

Latency Analysis



Load Simulation



Security of LLMs

Securing a Large Language Model focuses on protecting the critical **assets**:



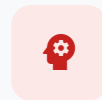
Training Data

Proprietary datasets and private information used during the pre-training and fine-tuning phases.



System Prompts

Proprietary instructions and system-level guardrails designed to guide and steer the model's behavior.



Model Weights & Architecture

The core neural network parameters, embedding configurations, and proprietary architectural designs.

Security of LLM

Critical Assets	Potential Attacks	Protection Methodology
Training Data	• Data Extraction: Recovering verbatim training examples (e.g., emails, IDs) via targeted querying. • Membership Inference (MIA): Deducing whether specific private data was used during fine-tuning.	• Differential Privacy (DP/LDP): Injecting mathematical noise during pre-training or locally before cloud tuning. • Data Deduplication: Removing duplicate sequences from datasets to reduce memorization. • Utility-Preserving Masking: Balancing privacy and performance via token reconstruction.
System Prompts	• Prompt Stealing: Reverse-engineering proprietary prompts by training models on the LLM's input-output behavior. • Prompt Leakage / Injection: Tricking the model to bypass guardrails and output its hidden instructions.	• Perturbation / Filtering: Modifying generated answers or adding slight noise to disrupt extraction alignment. • Context Obfuscation: Masking prompt intents or sensitive entities within the context window.
Model Weights & Architecture	• Model Stealing / Extraction: Systematically querying APIs (often exploiting logits) to train a cheaper "surrogate" mimic model. • Parameter Inference: Probing the model to extract hidden dimensions or embedding details.	• API Restrictions: Limiting query rates and restricting access to output probability distributions (logits). • Secure Inference (SMPC): Using cryptography to process data without exposing model weights. • Watermarking: Embedding traceable linguistic signals into generated text.

Sha, Zeyang, and Yang Zhang. "Prompt stealing attacks against large language models." *arXiv preprint arXiv:2402.12959* (2024).

Carlini, Nicholas, et al. "Extracting training data from large language models." *30th USENIX security symposium (USENIX Security 21)*. 2021.

Li, Yansong, et al. "Privacy-Preserving Parameter-Efficient Fine-Tuning for Large Language Model Services." *IEEE Transactions on Audio, Speech and Language Processing* (2025).

Security of Speaker Recognition

What are the critical **assets** of a speaker recognition system?



The Model Weights

- Expensive and compute-intensive to train new models from scratch.
- High engineering overhead and long time cycles required for deployment.



Original Speaker Embeddings

- Critical user **biometric** data that serves as the unique digital voiceprint.
- Leakage directly leads to **identity theft**, highly realistic voice cloning, and fraud.



The Consequence of Leakage

Once speaker embeddings are leaked, a new model needs to be retrained and re-deployed, which is very costly.

Security of Speaker Recognition

01 / Prevention

Encrypt speaker embedding via a separate **privacy protection module** to prevent leakage.

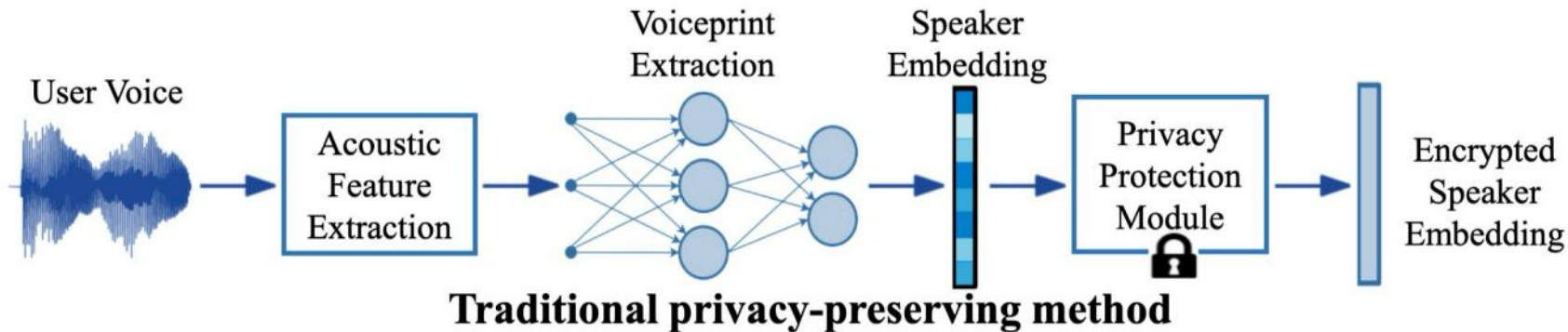
e.g., Random Matrix, Hash Mapping, Homomorphic Encryption

02 / Mitigation

If encrypted embeddings are leaked, **only update the protection module**. No need to retrain speaker encoder.

03 / Drawbacks

Traditional privacy-preserving frameworks often suffer from **compromised quality** and **information loss**.



Security of Speaker Recognition

01 / Encryption key

Additional Key Input: A 64-dim random vector is concatenated to the 256-dim RawNet3 output, followed by a FC layer.

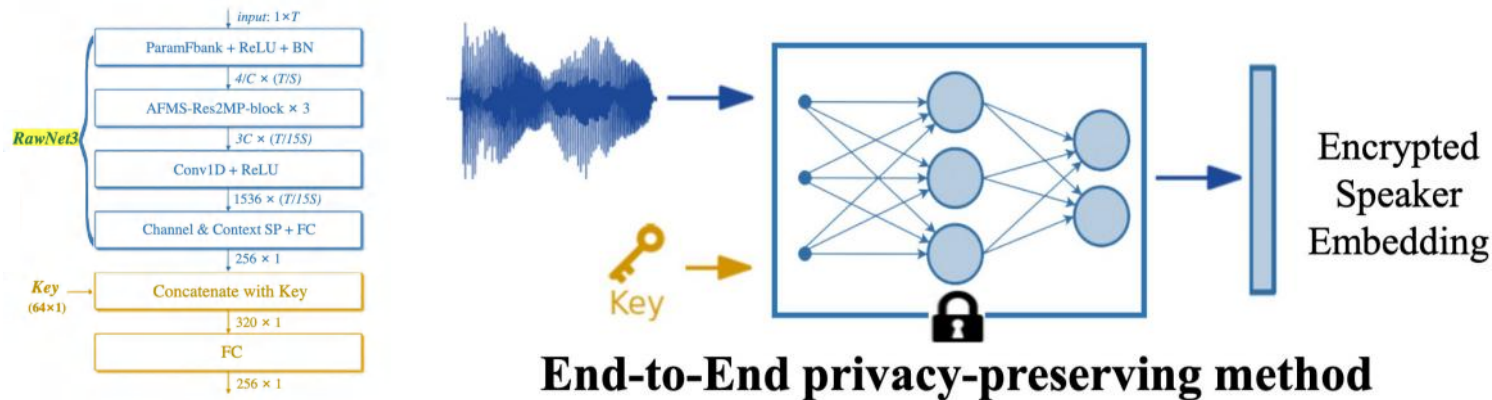
02 / Mitigation

Instant Mitigation: If encrypted embeddings are leaked, you can secure the system by just updating the key.

03 / Advantage

No Retraining: Eliminates cost of retraining the core model.

Minimal information loss: Encryption is part of E2E training.



Section 5

Enrollment Audio as Memory



Speaker enrollment: Static

For speaker verification, how do we collect reference audio for enrollment?

Typical approach: **Initialization setup** (one-time calibration)

Challenges of Static Enrollment



Different devices

Acoustic signatures vary significantly across microphones and hardware.



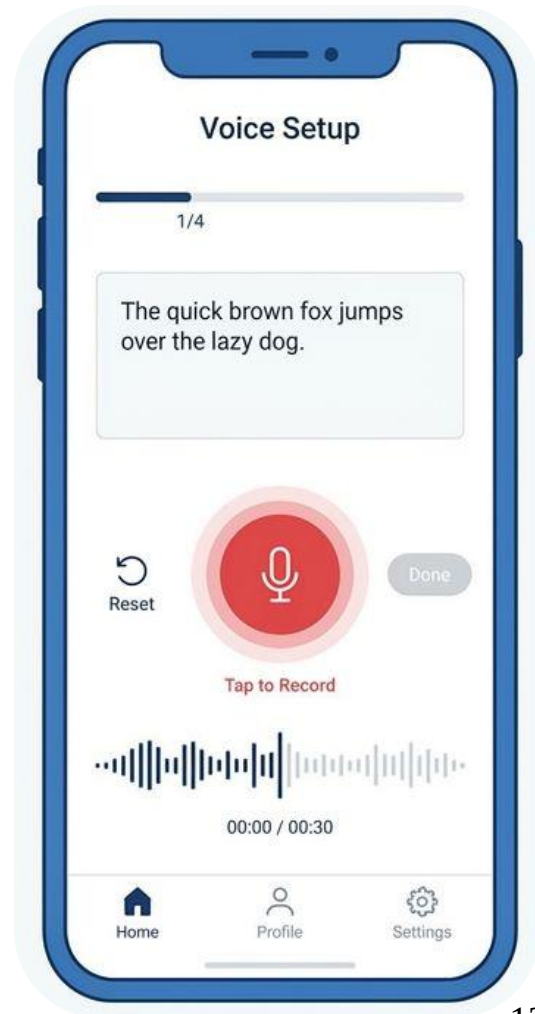
Different acoustic conditions

Background noise, reverb, and environmental changes distort voice patterns.



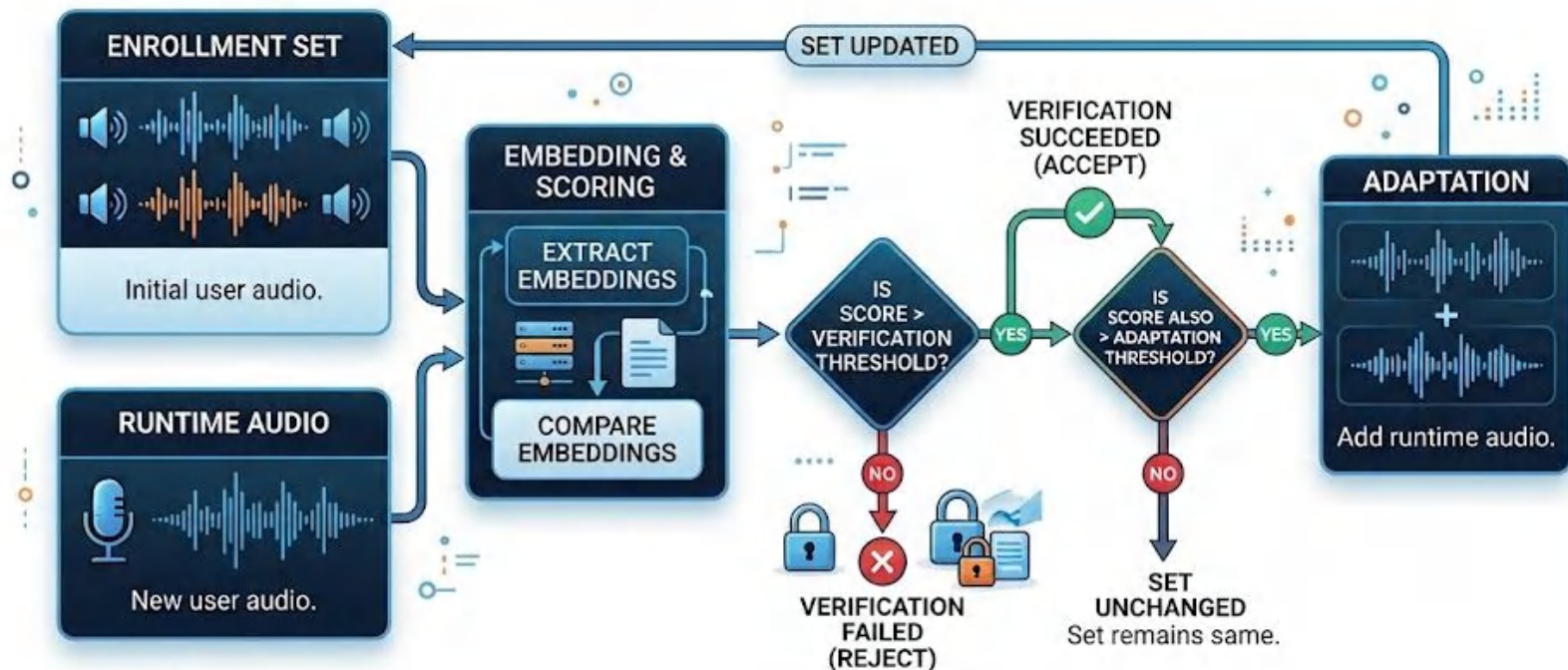
User voice drift over time

Natural voice alterations due to aging, fatigue, or illness.



Speaker enrollment: Adaptive

- Better approach: **Dynamic speaker adaptation**



Growing enrollment set

- When user keeps using product, enrollment set may grow huge
- This is similar to LLM dialogue history
- We need a way to **compress the memory**

LLM memory compression	Analogy in speaker recognition
Sliding window: Only keep last K conversations/turns.	Only keep last K adaptations? <i>Bad - We will lose diversity of enrollment coverage.</i>
Summarization: Use LLM to summarize dialogue history into concise text.	Average all embeddings? <i>Bad - We will lose lots of information.</i>
KV cache compression: Only preserve important KV pairs (e.g. H2O, Scissorhands).	Attentive scoring! <i>Good - We extract critical information from diverse conditions.</i>

Averaged scoring

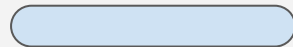
We may have **many enrollment utterances**:

- Recordings under different **acoustic environments**
- Continuous **adaptation** to mitigate voice drift over time

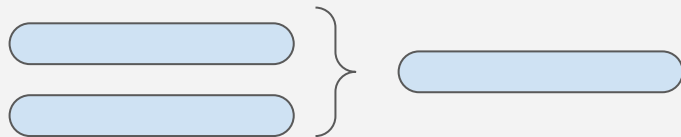
How do we **aggregate** enrollment embeddings:

- Common approach: Simply **average** all the embeddings
- But we are **losing information** by averaging

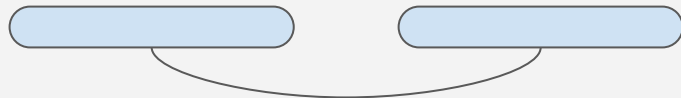
Conventional speaker embedding is **fixed-dim vector**:



Embeddings of multiple enrollments are **averaged**:



Two embeddings are compared via **cosine similarity**:



Attentive scoring

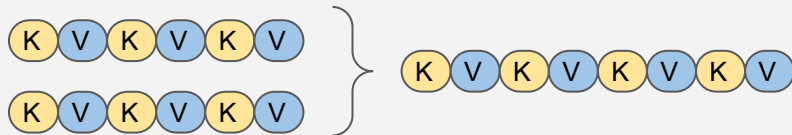
How to **preserve information** from multiple enrollment?

- Represent speaker embeddings as a **group of key-value pairs** (query is optional)
- Embeddings can be **concatenated**, to preserve information
- Optionally **prune** some key-values pairs based on similarity, to avoid embedding being too long
- **Attentive score** of two embeddings: use key scores as attention weights for value scores

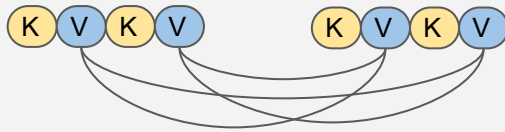
Proposed embedding is a **group of key-value pairs**:



Embeddings of multiple enrollments are **concatenated** first, then optionally **pruned**:



Two embeddings are compared via **attentive scoring**:



Attentive scoring: Pruning

System Parameters

- X** **Pairs per Utterance** Number of key-value pairs of each utterance
- Y** **Enrollment Utterances** Total number of enrollment utterances
- Z** **Maximum Threshold** Max number of key-value pairs at all time

The Goal

Prune the accumulated key-value pairs to maintain a constant computational footprint.

$$X \cdot Y \rightarrow Z$$

Compress total pairs from $X \cdot Y$ down to target size Z

Attentive scoring: Pruning

Algorithm 1: Self-similarity based pruning

- Start with $X \cdot Y$ initial key-value pairs
- Merge **most-similar** key-value pairs iteratively, via Agglomerative Hierarchical Clustering (AHC)
- Until we have only Z final key-value pairs

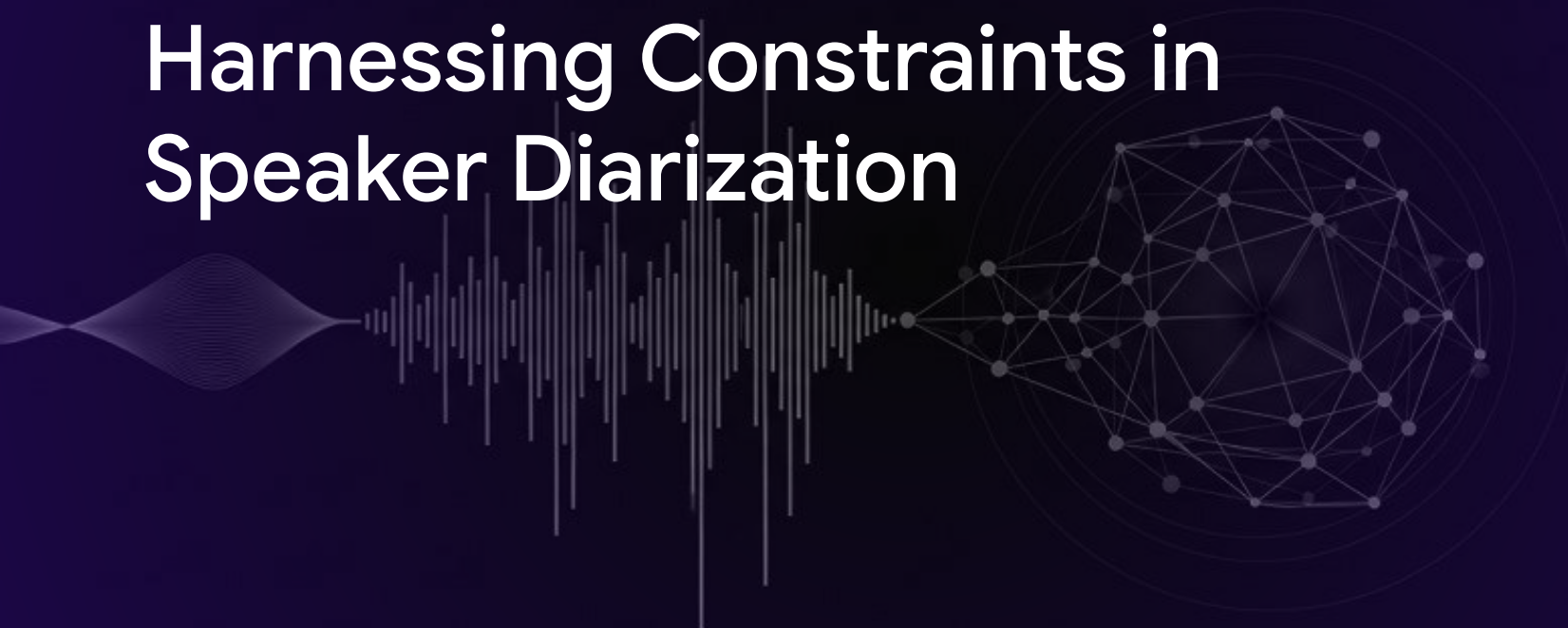
Attentive scoring: Pruning

Algorithm 2: Reference based pruning

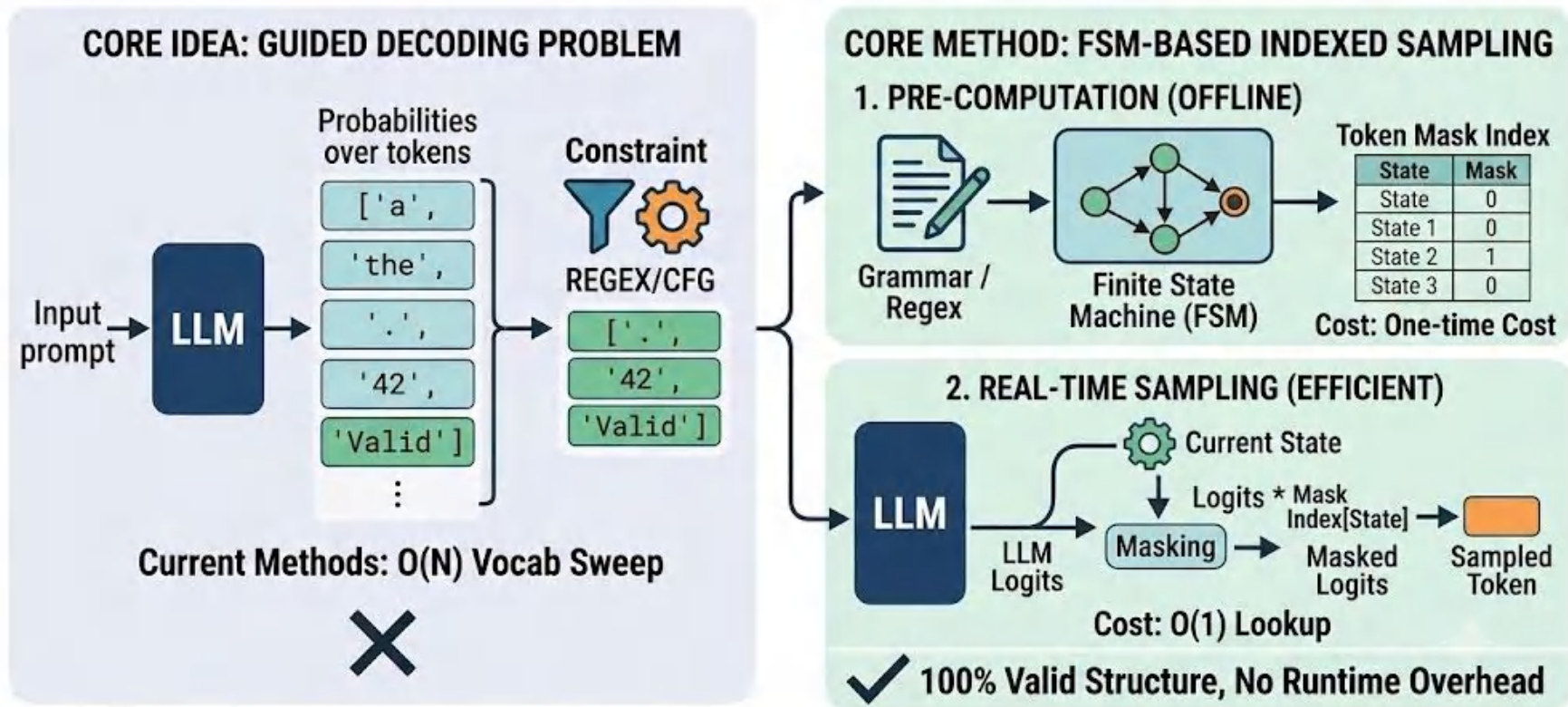
- Given a **cohort** of C reference embeddings
- Compute the **C attentive scores** between the current embedding and the reference embeddings
- Find the Z final key-value pairs to **minimize the changes** to the C attentive scores after pruning

Section 6

Harnessing Constraints in Speaker Diarization



Constraining LLM output: Guided generation



Constraints for speaker diarization



Lexical

Speaker boundary should not fall in the middle of a word or phrase.



Speaker Turn

Segments across speaker turn should belong to different speakers.



Temporal-Spatial

Different meeting rooms should never have same speakers.

Park, Tae Jin, et al. "Speaker diarization with lexical information." arXiv preprint arXiv:2004.06756 (2020).

Xia, Wei, et al. "Turn-to-diarize: Online speaker diarization constrained by transformer transducer speaker turn detection." ICASSP 2022.

Why speaker turn?

- Annotating speaker labels is **too expensive**
- From “**speaker labels**” to “**speaker turns**”
 - Annotating speaker labels requires frequently **checking previous speakers** from long term history
 - Annotating speaker turns:
 - Only focuses on local **short term** context (**no looking back**)
 - **Minimal incremental efforts** on top of regular ASR annotation

spkA: good morning
spkB: morning how are you
spkA: good what about you



good morning <st>
morning how are you <st>
good what about you <st>

Speaker labels

Speaker turns

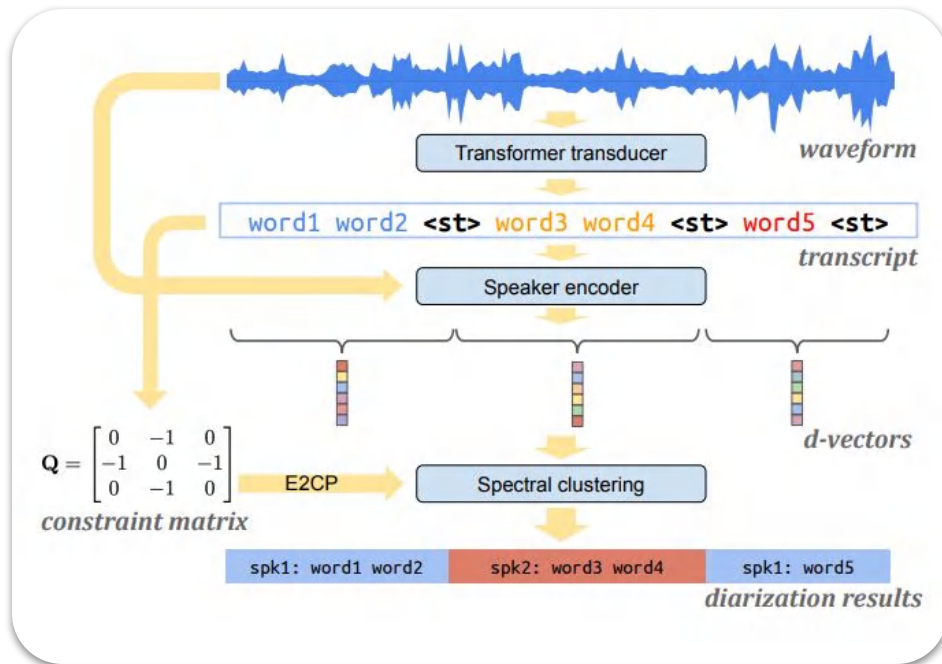
Speaker turn as constraints to clustering

Speaker turn modeling:

- **Just an augmented ASR model** - Add one more token to the ASR output vocabulary `<st>`

Constraining clustering:

- **Speaker turn for segmentation** - No speaker segments cross turns
- **Soft constraints** - Build constraint matrix, and apply to affinity matrix used by spectral clustering

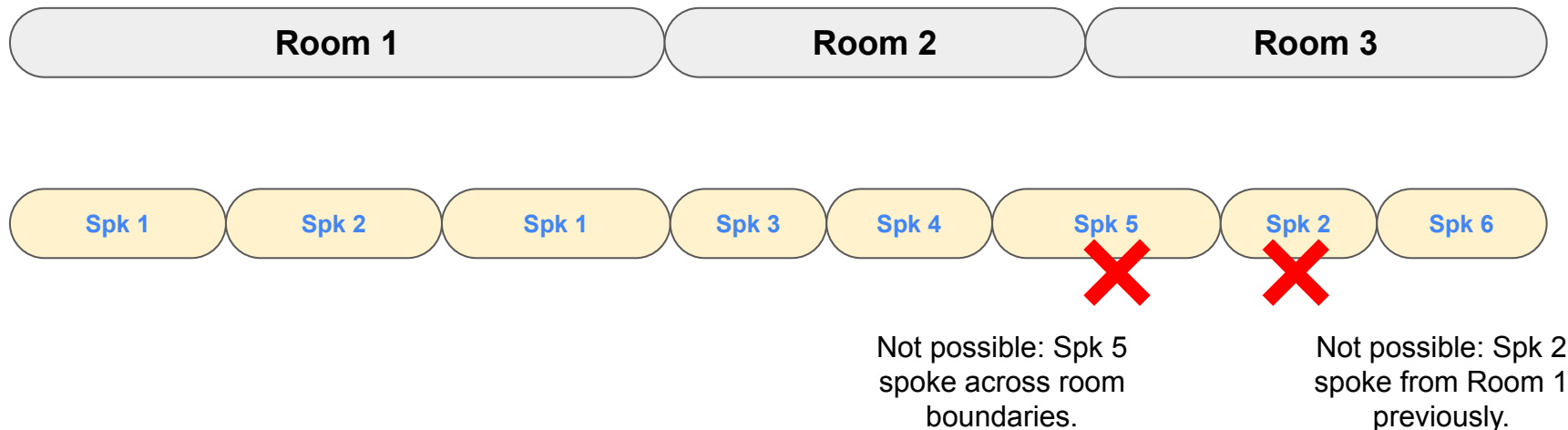


Xia, Wei, et al. "Turn-to-diarize: Online speaker diarization constrained by transformer transducer speaker turn detection." ICASSP 2022.

Temporal-spatial constraints

Special constraints for **virtual meeting diarization**:

- For each room, we know which room is active
- The same speaker should never appear in different rooms



Temporal-spatial constraints

Clustering based diarization

- Similar to speaker turn constraints, apply constraint to affinity matrix
- **-1**: Cannot-Link (CL)
- **+1**: Must-Link (ML)
- **0**: Otherwise

Room 1	Room 2	Room 3
0	-1	-1
-1	0	-1
-1	-1	0

Temporal-spatial constraints

Multimodal LLM (MLLM) based diarization

- Input: Text instruction + Audio
- Output: Rich transcription (Time + ASR + Speaker)

Prompted constraints (soft)

- Training the MLLM with prompts constraints, e.g.:

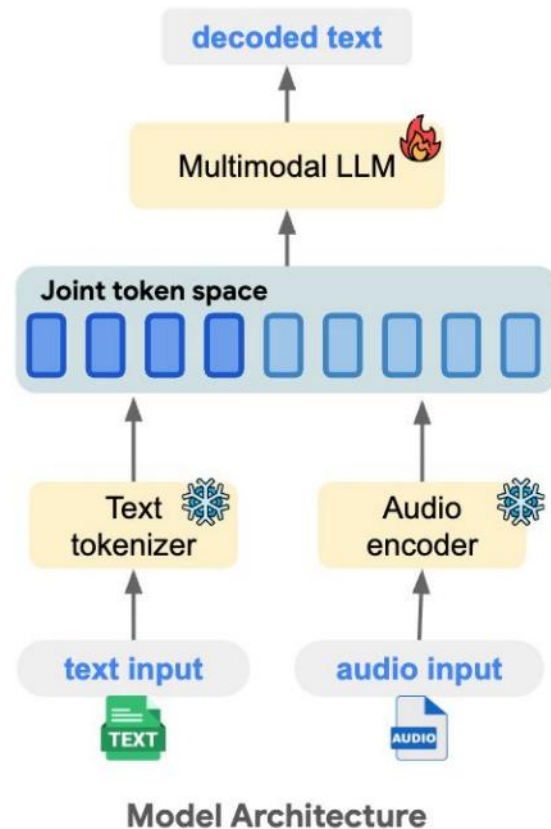
Recognize the following speech turn by turn. Each turn should use the following format:

[Start_time-End_time] Speech transcript <spk:N>

Same speaker should never appear across rooms:

Room1: 00:00:00.00-00:00:52.45, 00:02:52.83-00:03:01.21

Room2: 00:00:54.02-00:02:51.19, 00:03:03.06-00:03:07.12



Temporal-spatial constraints

Multimodal LLM (MLLM) based diarization

- Input: Text instruction + Audio
- Output: Structured ASR + Timing + Speaker labels

Guided generation (hard)

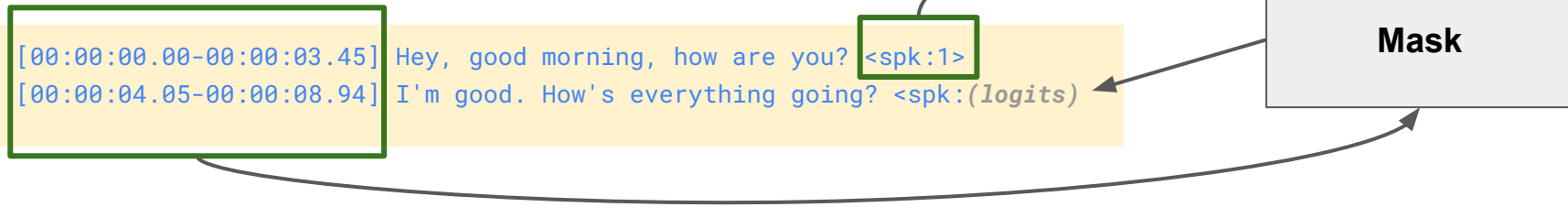
```
[00:00:00.00-00:00:03.45] Hey, good morning, how are you? <spk:1>  
[00:00:04.05-00:00:08.94] I'm good. How's everything going? <spk:(logits)>
```

Temporal-spatial constraints

Multimodal LLM (MLLM) based diarization

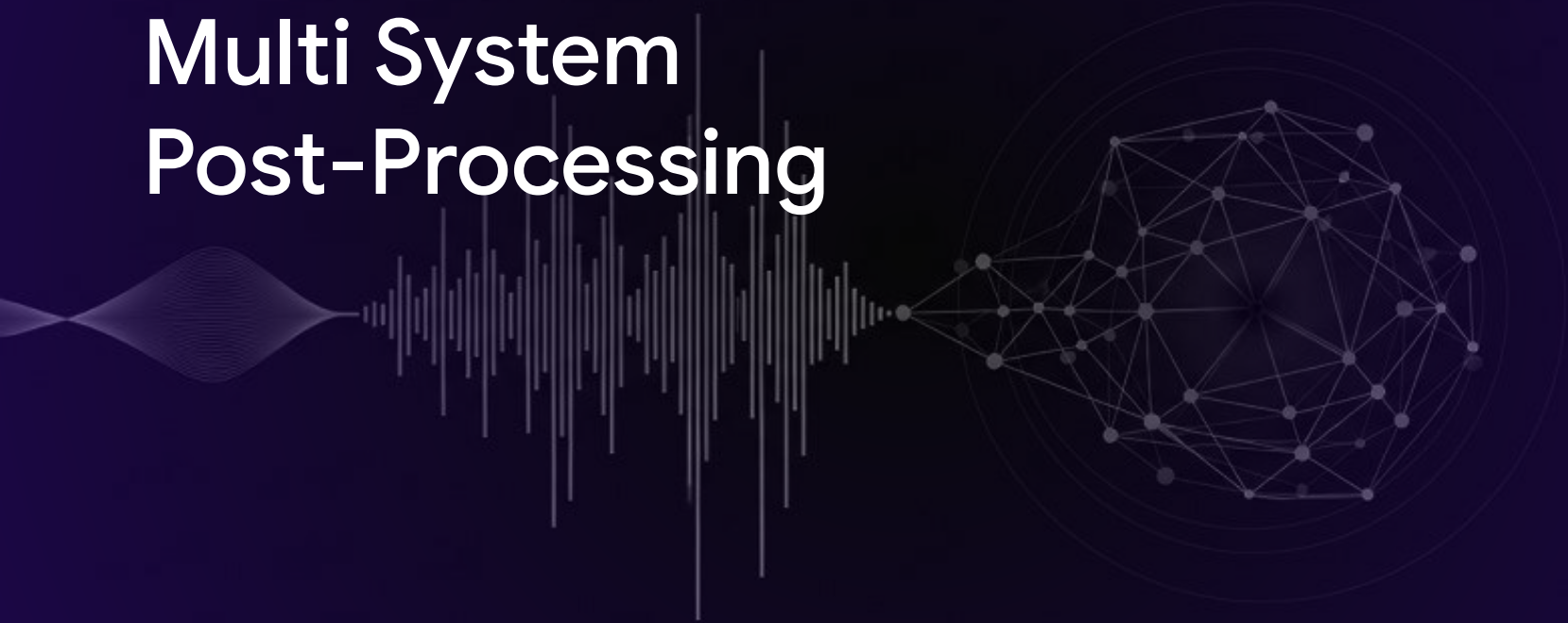
- Input: Text instruction + Audio
- Output: Structured ASR + Timing + Speaker labels

Guided generation (hard)



Section 7

Multi System Post-Processing



Error correction

- When we combine ASR and diarization systems, we see lots of **few-words-off errors**
- This is because ASR outputs and diarization outputs are combined via an **orchestration module** (a.k.a. “reconciliation”)
- The **timing information** from the two systems are not consistent

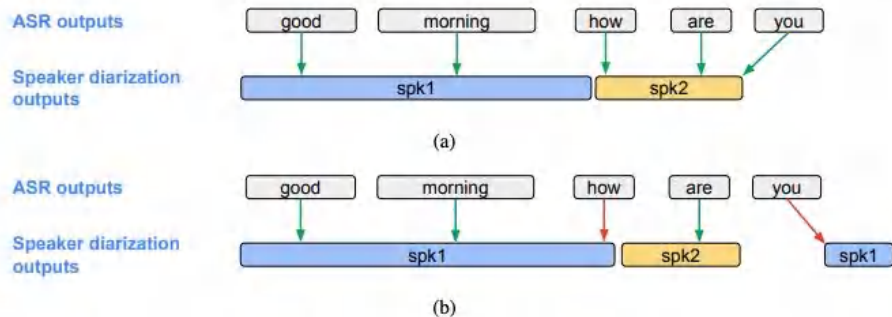
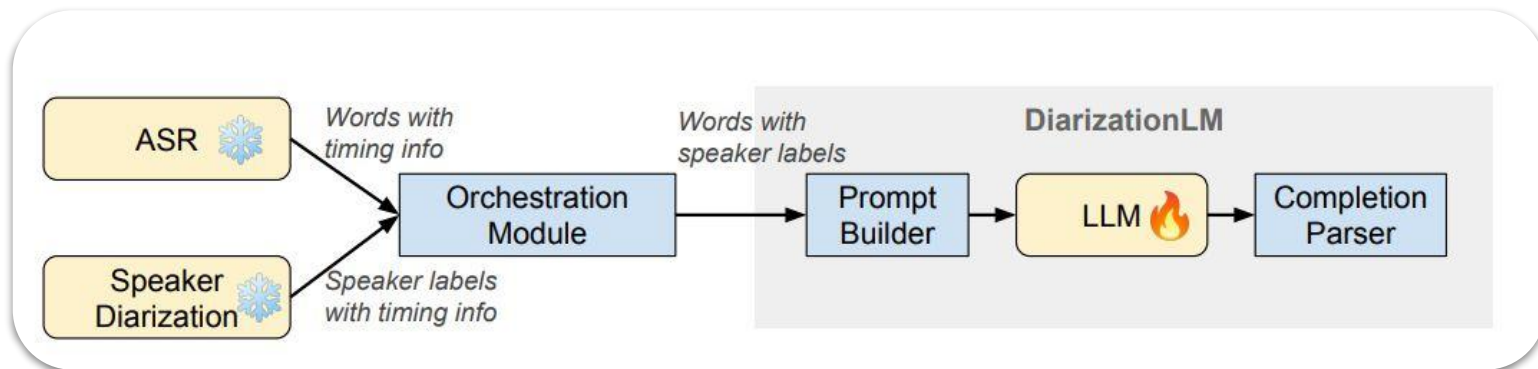


Figure 1: The orchestration module associates each word from the ASR transcript with a speaker label from the speaker diarization outputs. (a) In this example, all words are associated with the correct speaker labels (green arrows). The words “good”, “morning”, and “are” and “you” are associated with the only speaker label that overlap with them. The word “how” overlaps with both spk1 and spk2, but has bigger overlaps with spk2, thus is associated with spk2. The word “you” does not overlap with any speaker, but is closest to spk2, thus is associated with spk2. (b) In this example, two words are associated with wrong speaker labels (red arrows) due to inconsistent timing information from the two systems. The word “how” is mistakenly associated with spk1, since spk1 has more overlap with this word than spk2. The word “you” is mistakenly associated with spk1, since spk1 is closer to this word than spk2.

DiarizationLM: The high-level framework

- Take any existing ASR and speaker diarization systems
- Assume the outputs are **hyp**, while the ground truth is **ref**
- Finetune LLM to predict **ref** given **hyp**

It works with any off-the-shelf ASR and diarization systems.



Represent ASR + diarization results in text

- After orchestration, we get a word sequence and a speaker sequence
- We need a pure text representation so we can prompt LLM

Example text representation of the prompt

```
Word sequence:      ["good", "morning", "how", "are", "you"]  
Speaker sequence:   [1, 1, 2, 2, 2]  
Text representation: "<spk:1> good morning <spk:2> how are you"
```

Different word sequences in hyp and ref

- The ref and hyp sequences will have:
 - **Different words** - LLM for ASR error correction is interesting! But not our focus in DiarizationLM
 - **Different speakers** - We want to focus on diarization errors
- Trying to correct both ASR and diarization errors at the same time will confuse the LLM - how do we avoid this?
- Our solution: **Transcript-Preserving Speaker Transfer (TPST)**

Transcript-Preserving Speaker Transfer (TPST)

- Assume we have two sets of diarized transcript:
 - Source
 - Target
- TPST will transfer the speaker labels from the source sequences to the target sequences, such that:
 - The transferred speaker label sequence has a 1-to-1 association with the target word sequence.
 - The transferred speaker labels are more consistent with the source speaker labels.

Example inputs and output of the TPST algorithm

Source words:	hello good morning hi how are you pretty good
Source speakers:	1 1 1 2 2 2 2 1 1
Target words:	hello morning hi hey are you be good
Target speakers:	1 2 2 2 1 1 2 1
Transferred speakers:	1 1 2 2 2 2 1 1

 **LLM is trained for this correction**

When does DiarizationLM help the most?

- Broken sentences
- Disfluency
- Interruptions

fe_03_11210 (Δ WDER =6.35%)	... <spk:1> the vikings mine's the eagles i'm from new jersey oh all right i have my jersey on now i watch the game tonight yeah well i i may i may just watch <spk:2> part of it tonight too then but uh it's a case as i say if i had to pay for it i probably wouldn't watch it <spk:1> i wouldn't either uhhuh <spk:2> unless <spk:1> it was an eagles game ...	... <spk:1> the vikings mine's the eagles i'm from new jersey <spk:2> oh all right <spk:1> i have my jersey on now i watch the game tonight yeah <spk:2> well i i may i may just watch part of it tonight too then but uh it's a case as i say if i had to pay for it i probably wouldn't watch it <spk:1> i wouldn't either <spk:2> uhhuh <spk:1> unless it was an eagles game ...
fe_03_11159 (Δ WDER =4.05%)	... <spk:2> yeah <spk:1> anniversary that's horrible <spk:2> yeah <spk:1> yeah it's not good <spk:2> i <spk:1> i hear i hear you there that's not a good thing you <spk:2> know i mean of course you know that's a day that will go down instantly nobody will ever remember it ...	... <spk:1> yeah anniversary that's horrible yeah yeah it's not good i i hear i hear you there that's not a good thing <spk:2> you know i mean of course you know that's a day that will go down instantly nobody will ever remember it ...

Utterance	Before DiarizationLM	After DiarizationLM
fe_03_07146 (Δ WDER =8.80%)	... <spk:3> it's it's <spk:1> more of summer always like you know we never experience a bit cold over here <spk:4> usually it'll <spk:1> be warm or like very hot in summer yeah and <spk:3> extremely hot yeah with high humidity my humidity is pretty <spk:1> much high because i stay close to the sea coast over here <spk:3> yeah <spk:1> so <spk:3> that makes you live houston is it like houston where you live yeah i i i live <spk:1> in houston	... <spk:1> it's it's more of summer always like you know we never experience a bit cold over here usually it'll be warm or like very hot in summer <spk:2> yeah and extremely hot yeah with high humidity my <spk:1> humidity is pretty much high because i stay close to the sea coast over here <spk:2> yeah so that makes you live houston is it like houston where you live <spk:1> yeah i i i live in houston ...
fe_03_06816 (Δ WDER =6.61%)	... <spk:3> uhuh <spk:2> did you see the the woman golfer that was on this the one <spk:1> monica yeah yeah <spk:2> what's her name monica stone yeah mhm she she <spk:1> blew out she fell out of that tournament but i didn't think she'd do it she she's girls can't compete against guys ...	... <spk:2> uhuh did you see the the woman golfer that was on this the one <spk:1> monica yeah yeah <spk:2> what's her name monica stone <spk:1> yeah <spk:2> mhm <spk:1> she she blew out she fell out of that tournament but i didn't think she'd do it she she's girls can't compete against guys ...

Try it yourself!

- **Code:**

<https://github.com/google/speaker-id/tree/master/DiarizationLM>

- **Pretrained model on Fisher:**

<https://huggingface.co/google/DiarizationLM-8b-Fisher-v2>

- **Web demo:**

<https://huggingface.co/spaces/diarizers-community/DiarizationLM-GGUF>

DiarizationLM GGUF inference on CPU

A demo of the DiarizationLM model finetuned from Llama 3. In this demo, we run a 4-bit quantized GGUF model on CPU.

To learn more about DiarizationLM, check our paper: <https://arxiv.org/abs/2401.03506>

prompt

<speaker:1> This demo looks really <speaker:2> good! Thanks, I am glad to hear that.

Clear

Submit

output

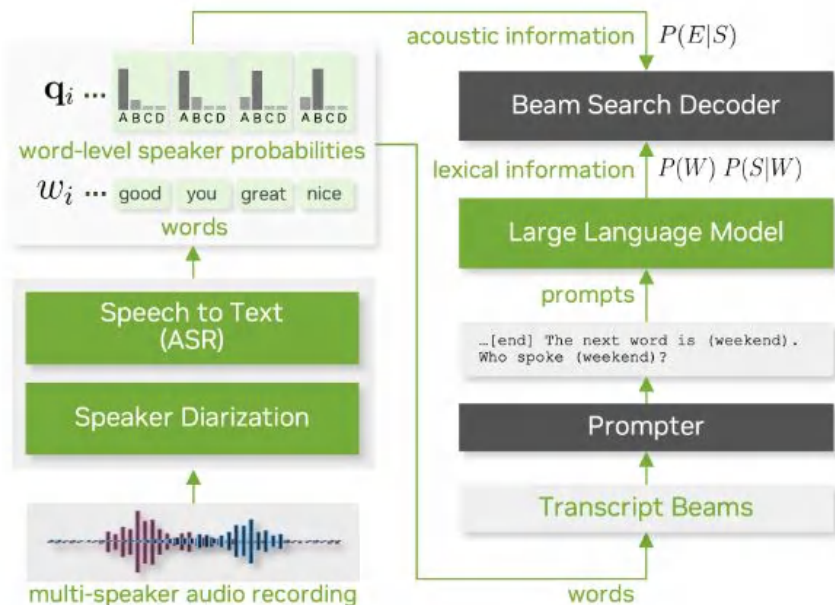
<speaker:1> This demo looks really good!
<speaker:2> Thanks, I am glad to hear that.

Examples

<speaker:1> Hello, my name is Tom. May I speak to Laura <speaker:2> please? Hello, this is Laura. <speaker:1> Hi Laura, how are you? This is <speaker:2> Tom. Hi Tom, I haven't seen you for a <speaker:1> while.

<speaker:1> This demo looks really <speaker:2> good! Thanks, I am glad to hear that.

Error correction: Word-by-word prompting



[Speaker0]:how are you doing these days
[Speaker1]:things are going very well
[Speaker0]:well tell me more
[Speaker1]:there is a project that i'm
[end]
Question: The next word is (working).Who spoke (working)?
Answer:[Speaker1]

Error correction: Speaker & word embeddings

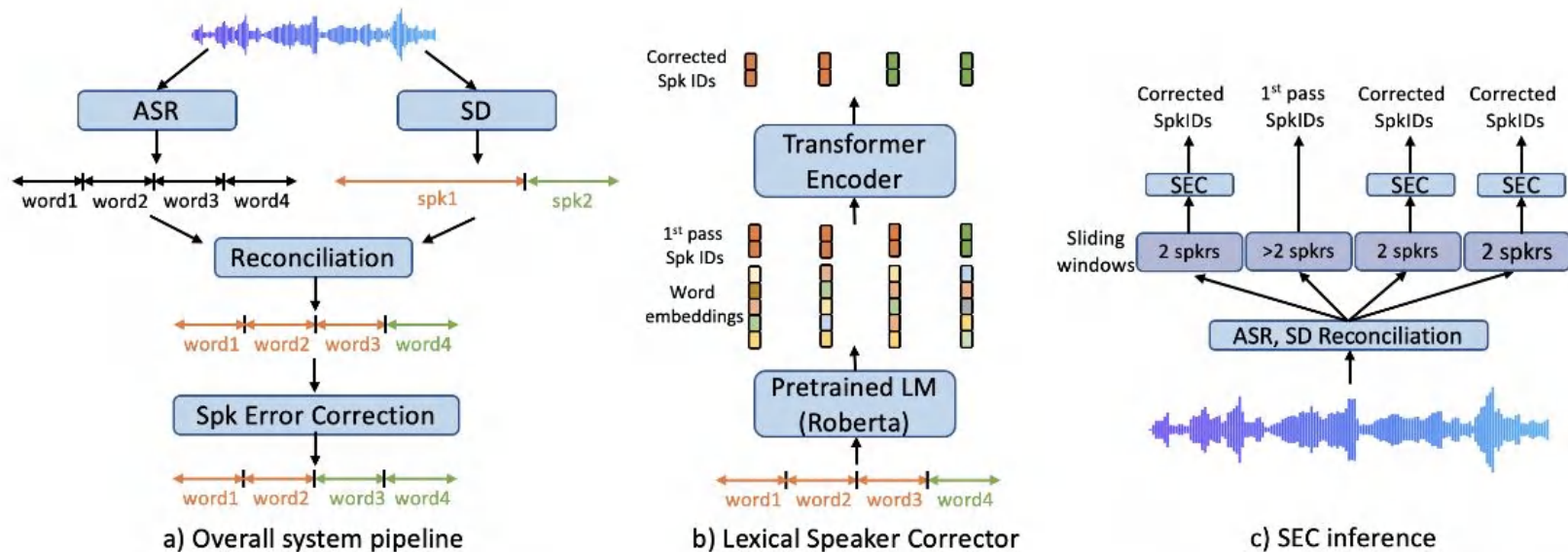


Figure 1: (a) Speaker Error Correction as a 2nd-pass post-processing step to the traditional SD-ASR system, (b) Lexical SEC: Word embeddings from the LM, Speaker IDs from SD are passed to the Transformer Encoder to get the corrected Speaker IDs, (c) SEC inference performed on sliding windows with 2 hypothesis speakers.

Model ensemble

- When we have multiple LLMs, how do we ensemble their outputs?
- Example approach: **Mixture-of-Agents**

Table 1: Aggregate-and-Synthesize Prompt to integrate responses from other models.

You have been provided with a set of responses from various open-source models to the latest user query. Your task is to synthesize these responses into a single, high-quality response. It is crucial to critically evaluate the information provided in these responses, recognizing that some of it may be biased or incorrect. Your response should not simply replicate the given answers but should offer a refined, accurate, and comprehensive reply to the instruction. Ensure your response is well-structured, coherent, and adheres to the highest standards of accuracy and reliability.

Responses from models:

1. [Model Response from $A_{i,1}$]
 2. [Model Response from $A_{i,2}$]
 - ...
 - n . [Model Response from $A_{i,n}$]
-

Model Ensemble

ENSEMBLE FRAMEWORK

DOVER Method

Diarization Output Voting Error Reduction (inspired by **ROVER**) combines outputs from multiple systems.

01. Iterative Label Mapping

Align and map speaker labels iteratively across different diarization systems.

02. Voting & Consensus

Perform majority voting on the aligned steps to output the final consensus labeling.

1 - Original labels



2 - Map System B to System A labels



3 - Map System C labels to System A+B labels



4 - Voting



Section 8

Harnessing Longform Diarization



Longform diarization



Short video — 1~2 min



Phone calls — 1~10 min



Podcast — 10~30 min



Meeting recordings — 30~120 min



Live event streaming — 2~10+ hours

For a 10-hour recording, assuming:

1.44M tokens

at 25ms per token

18K segments

at 2 seconds per segment

6K speaker turns

at 6 seconds per speaker turn

Cost of longform clustering

Clustering Algorithms

Offline Clustering

Runs on all embeddings, providing superior quality.

Online Clustering

Updates incrementally on the newest embedding, but yields worse quality.

Common Choice

Rerun offline clustering on all embeddings upon receiving a new embedding.

Longform Complexity

Longform clustering can be extremely expensive, scaling rapidly as data grows:

$$O(N^2)$$

Agglomerative Hierarchical Clustering (AHC)

$$O(N^\omega) \times N_{\text{autotune_steps}}$$

$$2.37 \leq \omega \leq 3$$

Spectral Clustering

Multi-stage clustering

Spectral Clustering

Main Clusterer — Offers the best performance in the pipeline

- **N** can't be too small: requires a minimum number of embeddings to be highly effective
- **N** can't be too big: computationally too expensive to execute

AHC

Pre-Clusterer — Cheaper alternative used for dynamic compression

- Caches the centroids of AHC to dynamically compress input embeddings
- Compresses newly observed $\mathbf{U}_2$ embeddings into $\mathbf{U}_1$ centroids
- Computational cost is strictly upper-bounded by $\mathcal{O}(\mathbf{U}_2^2)$

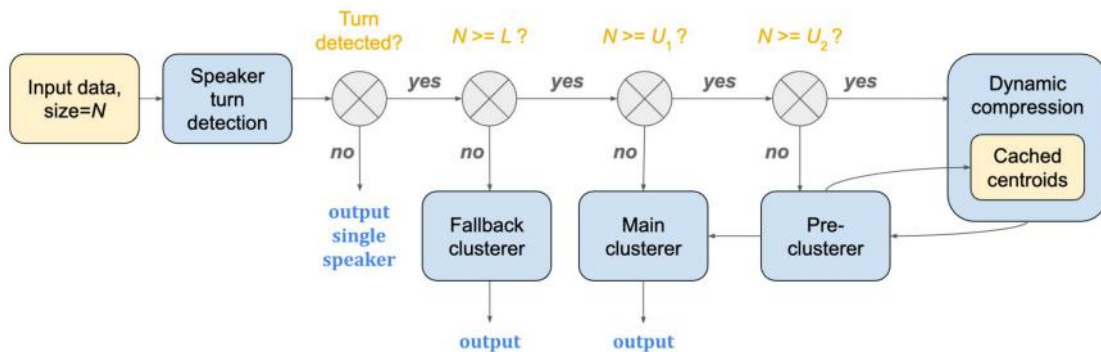
Multi-stage clustering

Cost & Quality Balance

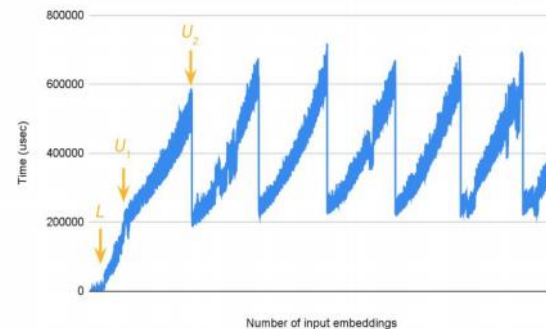
Achieving high-quality speaker diarization is computationally heavy. Cost and quality present an inevitable trade-off in streaming environments.

On-Device Optimized

Multi-stage clustering dynamically configures the pipeline for any cost upper bound, enabling seamless real-time execution even on mobile devices.



Clustering run time cost on Pixel 4 ($L=30$, $U_1=100$, $U_2=300$)

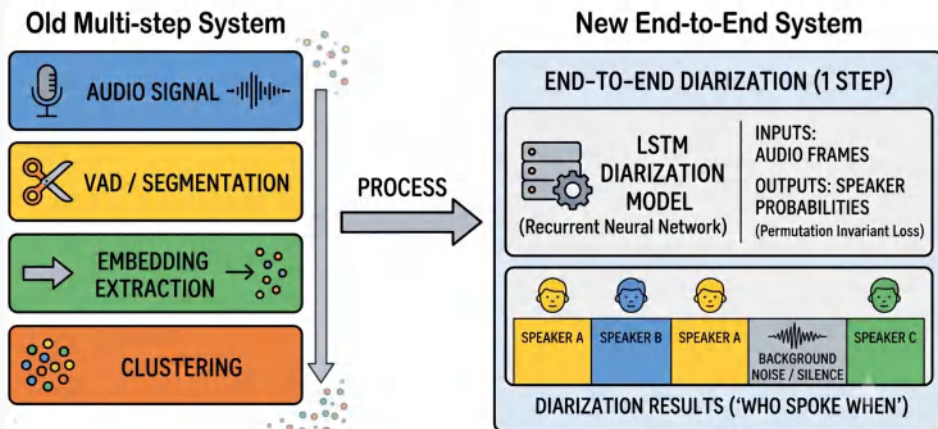


EEND as backbone models

Backbone Evolution

Shift to EEND Models

There had been a major trend to shift from **speaker embedding** models to **EEND** models.

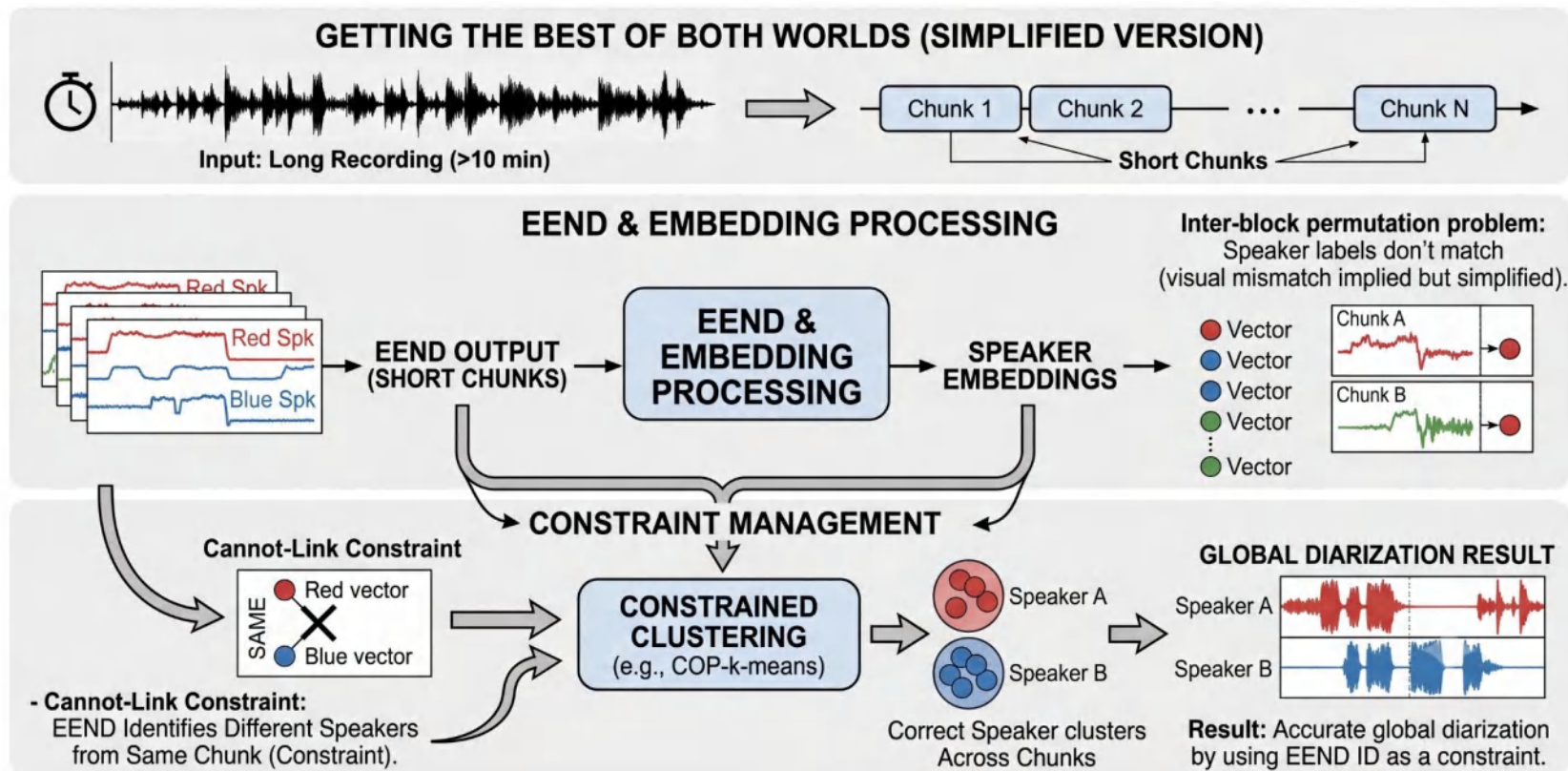


Key Benefits

EEND models provide key advantages over traditional embedding-based systems:

- Native Overlap Handling
- Elimination of Cascading Errors
- Direct Optimization of Diarization Errors
- Better Global Context

Longform solution: EEND-VC



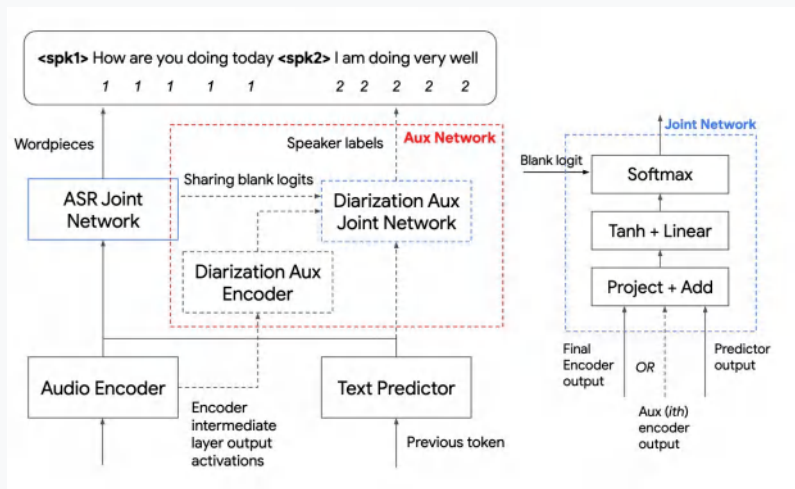
Word-Level E2E Neural Diarization (WEEND)

Approach Overview

Another approach to longform diarization:

- Build it into RNN-T
- Predict speaker label per word
- Leverage auxiliary encoder and joint network

WEEND Architecture



Multimodal LLM based longform diarization

Overview

- Training multimodal LLM with speaker diarization

Challenge for Longform

- **Length limitations:** Limit for each inference run due to cost and infra limitations.
- **Information flow:** Difficulty passing information across segments when longform audio is segmented.

Inference Workflow

Prompt

"Generate speech transcript and speaker labels for the following audio."



Multimodal LLM



Output

<speaker:1> Good morning.
<speaker:2> How are you?

Multimodal LLM based longform diarization

APPROACH 1

Clustering

Similar to EEND-VC, use per-segment diarization output as global constraints for clustering.

APPROACH 2

Sequential Prompting

Append past diarization output (text only) as prompt for next segment.

Prompt

"Below is speech transcript and speaker labels for the conversation before the audio:
<speaker:1> Good morning.
<speaker:2> How are you?

Generate speech transcript and speaker labels for the following audio."

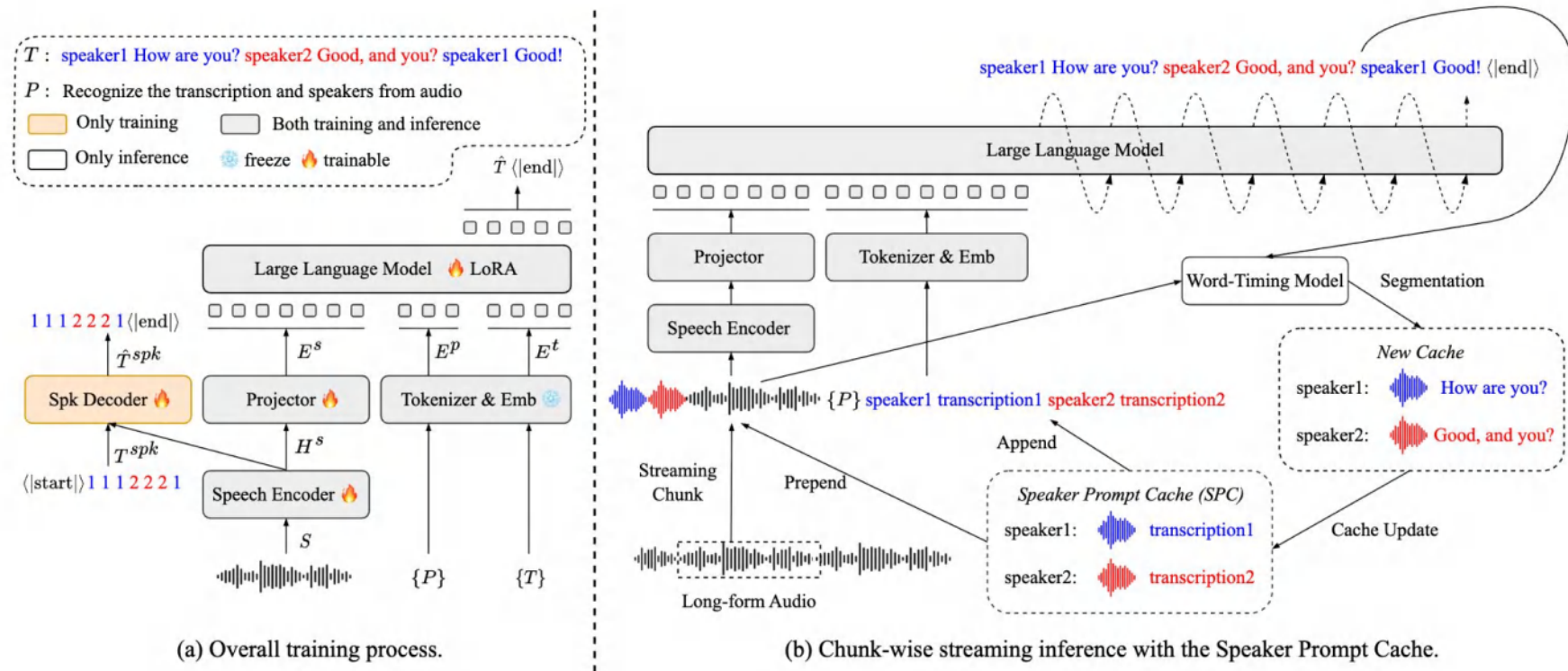
APPROACH 3

Audio Prompting

Extract **audio snippets for each speaker** based on diarization outputs.

Prompt next chunk with both past diarization output and audio snippets for all detected speakers.

Multimodal LLM based longform diarization



Diarization with Gemini 3.5 Transcribe



Unary API: Single request, single response

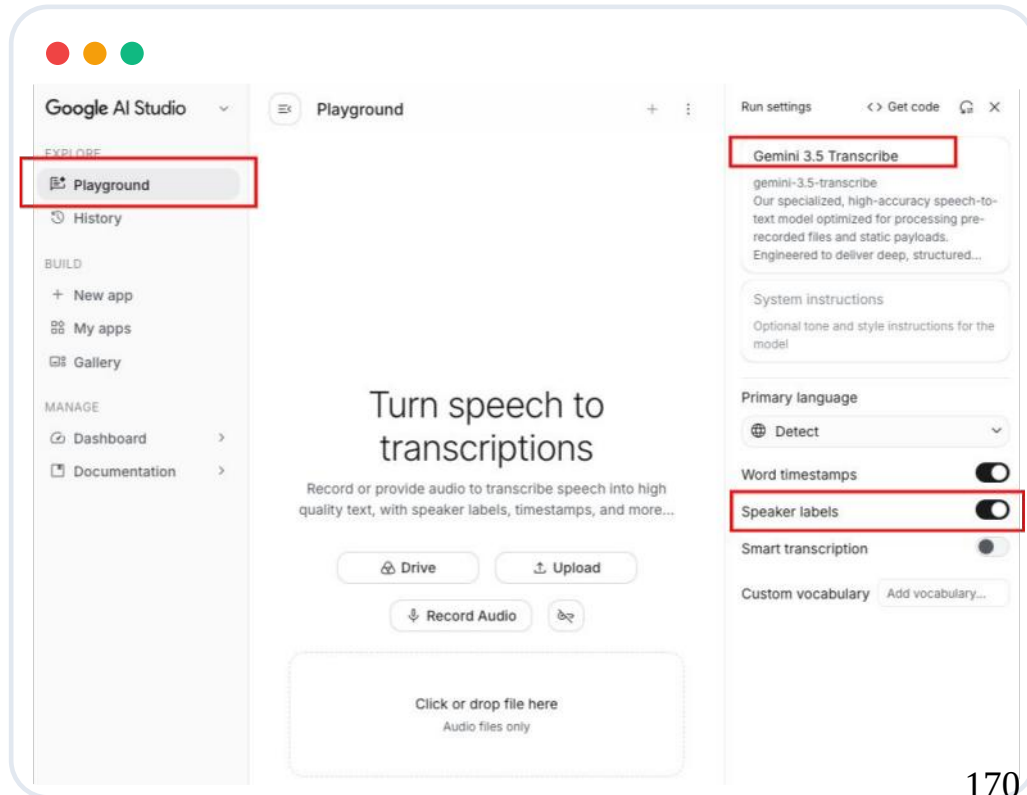
```
interaction = client.interactions.create(  
    model="gemini-3.5-transcribe",  
    input=[  
        {  
            "type": "audio",  
            "uri": audio_file.uri,  
            "mime_type": audio_file.mime_type,  
        }  
    ],  
    generation_config={  
        "transcription_config": {  
            "mode": {  
                "type": "verbatim",  
                "diarization_mode": "speaker",  
            },  
        },  
    },  
)  
print(interaction.output_text)
```

Diarization with Gemini 3.5 Transcribe

Google AI Studio Quick Access

https://aistudio.google.com/prompts/new_chat?model=gemini-3.5-transcribe

- 1 Select the Model**
In the right-hand panel, set the model to **Gemini 3.5 Transcribe**.
- 2 Enable Speaker Labels**
Toggle on **Speaker labels** to automatically detect and separate speakers in the transcript.
- 3 Upload & Run**
Upload your audio file or record directly to start transcribing with diarization.



Section 9

Closing Remarks



| Harnessing matters!

01

Evolution

Backbone speaker models will **keep evolving**: better embeddings, better EEND, and more powerful multimodal LLM.

02

Necessity

But harnessing will **never be gone** — it will only become more and more important.

03

Shift in Focus

From boosting the performance of weak backbone models, to **maximizing system and product excellence** by fully activating their potentials.

Thanks for your attention!

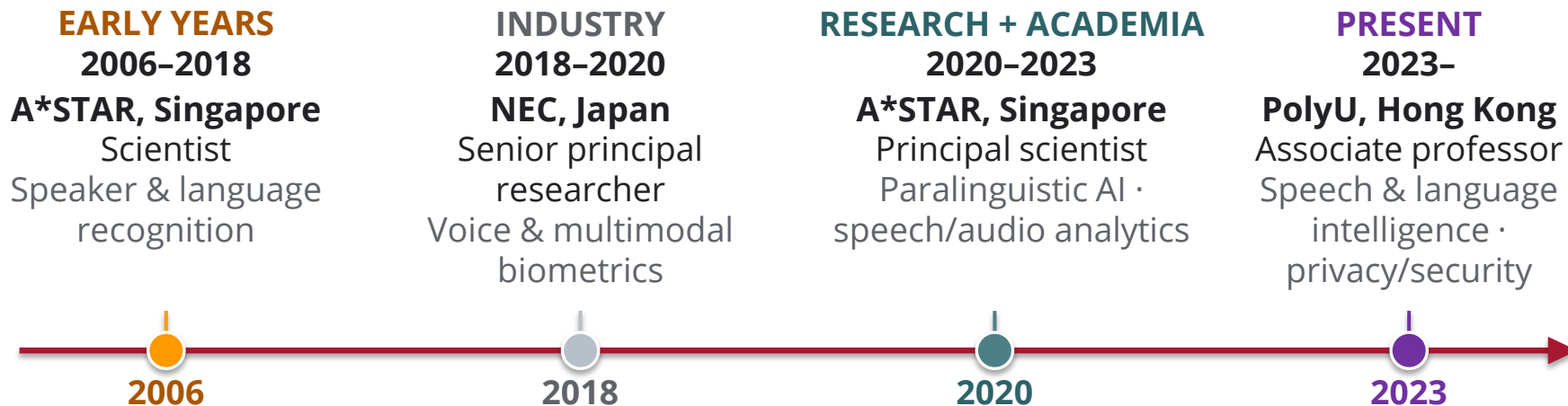
Chapter 3 of 3

Responsible Deployment Privacy, Security, and Anonymization



Kong Aik Lee The Hong Kong Polytechnic University

Self Introduction



- ISCA Distinguished Lecturer 2026 – 2027
- Chair, ISCA Speaker and Language Characterization (SpLC) SIG, 2025 – 2027
- Editor, Elsevier Computer Speech and Language (CSL), 2016 – Present
- Senior AE, IEEE SPL, 2024 – Present, Associate editor, IEEE T-ASL, 2017 – 2021
- Elected member, IEEE Speech and Language Technical Committee (SLTC), 2019 – 2021, 2021 – 2024

What this chapter is about

TUTORIAL TITLE

Speaker Modeling
beyond Recognition

FOCUS OF CHAPTER 3

Responsible
Deployment

KEY TAKEAWAY

Voice technology must protect identity while remaining useful, secure, and accountable.

Agenda

01

Voice Privacy,
Security, and Safety

02

Voice Spoofing and
Countermeasures

03

Speaker Privacy
Preservation

Section 1

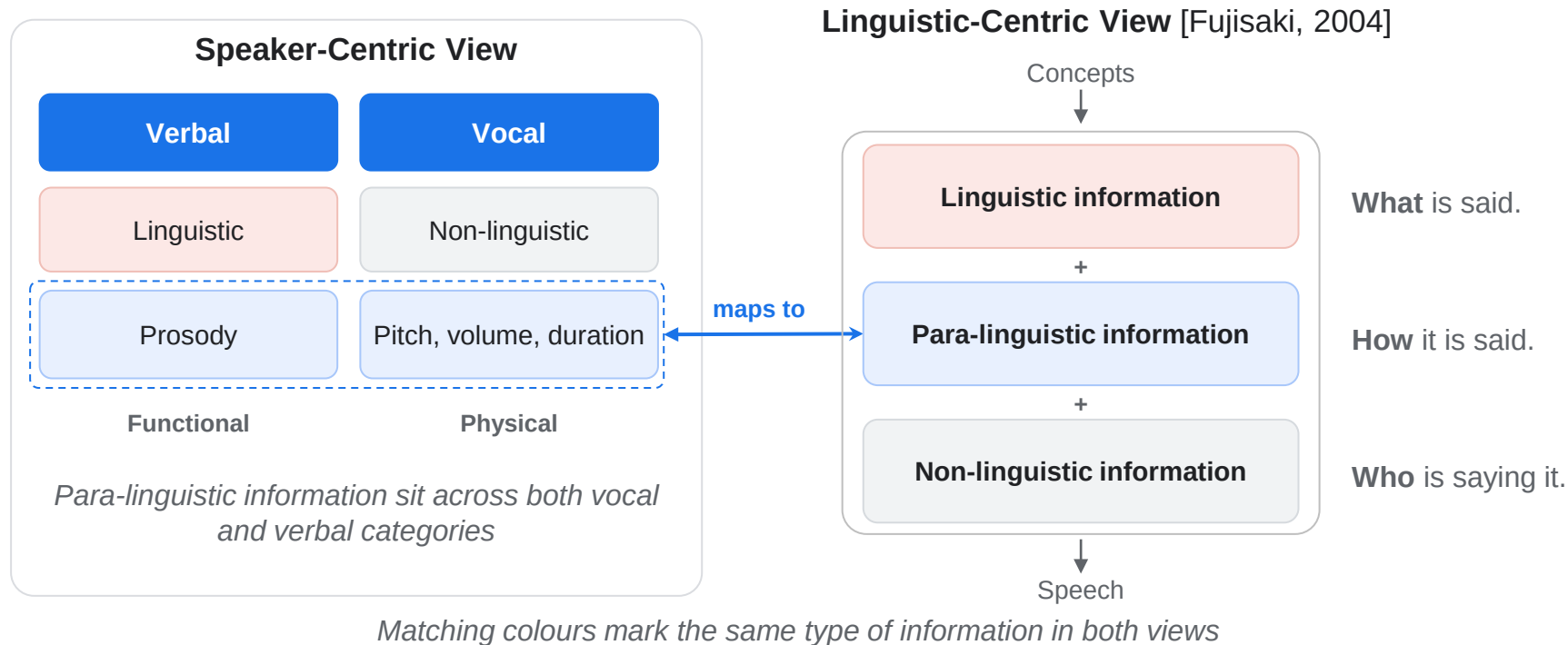
Voice Privacy, Security, and Safety

The background features a dark blue gradient. On the left, there is a white waveform graphic. To the right of the waveform is a circular network diagram with nodes and connecting lines. The text 'Section 1' is in a light green color, and the main title 'Voice Privacy, Security, and Safety' is in white.

Privacy, Personal Data, and Voice

- Privacy in general
 - **Privacy** is the **right to be let alone**, or freedom from interference or intrusion.
- Data privacy
 - **Data privacy** is the right to have control over how your **personal data** or information is collected and used (e.g., contact tracing).
- Personal data
 - **Personal data** denotes any data related to an **identifiable natural person**.
- Voice data contain a lot of personal information
 - **Vocal** (physical): age, gender, health, emotional state, ethnic origin, speaker identity.
 - **Verbal** (linguistic): political orientation, religious belief.
 - Background noise could also reveal personal information (e.g., whereabouts).

Linguistic vs Speaker Centric Viewpoints



H. Fujisaki, "Information, prosody, and modelling – with emphasis on tonal features of speech," in *Proc. Speech Prosody*, 2004.

Personal Data Taxonomy

Definition

REF: https://www.openriskmanual.org/wiki/Personal_Data_Taxonomy

A **Personal Data Taxonomy** is a classification of **Personal Data** according to various characteristics.

• Data internal to the Data Subject

- Preferences
- Knowledge
- Beliefs

• External Data

- Behavioral
- Demographics
- Physical
- Sexual

• Personally Identifying Information

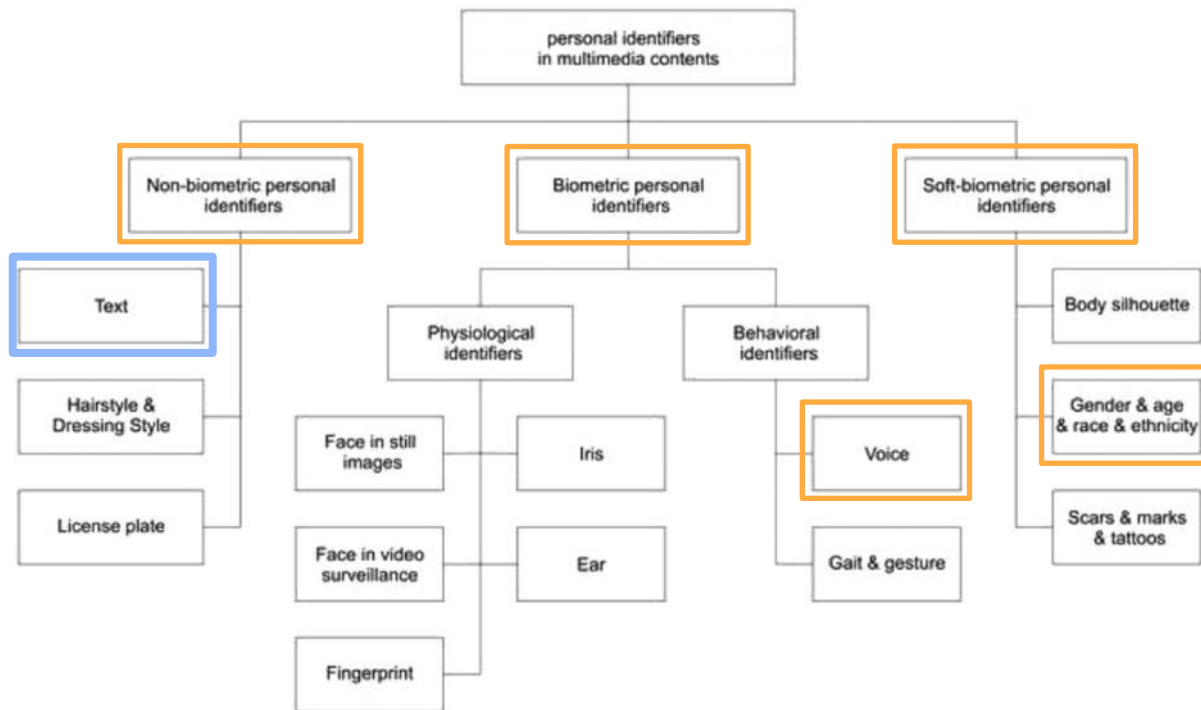
- Social Environment
 - Family
 - Friends
 - Professional
 - Public Life
 - Communications

- Financial Data
 - Transactional
 - Ownership
 - Financial Accounts
- Digital Tracking Data
 - Location
 - Device Telemetry
 - Contact

PII

Any information that can be used to distinguish one person from another.

Voice in the Context of PII



S. Ribaric, A. Ariyaeenia, and N. Pavesic, "De-identification for privacy protection in multimedia content: A survey," *Signal Processing: Image Communication*, vol. 47, 2016, pp. 131-151.

Privacy in Digital Edge

Evolution of Privacy Concept

TRADITIONAL

The right to be let alone



DIGITAL AGE

The right to control personal data

For voice: full control over the collection, storage and destruction of voice biometrics.

GDPR Personal Data Taxonomy

General Data

e.g., name, address – follows the principles of fairness and transparency.

Special Category Data (Art. 9)

Explicitly includes biometric data used for identification, requires strict protection and explicit consent.

AI Ethics – Responsible Deployment

- **Ethics** is the set of moral principles that help us discern between right and wrong.
- **AI ethics** is the set of principles used to guide how AI is designed, deployed and governed. It is concerned with how to maximize the benefits of AI while minimizing its risks, preventing harm, and ensuring the technology aligns with human values.
- Five pillars of AI ethics: fairness, explainability, **data privacy**, **accountability**, and **safety & Security**.
- **Responsible deployment** prevents the misappropriation of speech technologies (ASV, TTS/VC)
 - Deepfake detection [\[security\]](#)
 - Anonymization [\[data privacy\]](#)
 - Watermark of synthetic speech [\[accountability, safety\]](#)
 - Timbre-preserving voice anonymization (asynchronous voice anonymization) [\[data privacy\]](#)

Section 2

Voice Spoofing and Countermeasures

Voice Spoofing: Definition

Presentation attacks

TARGET

Target is an ASV system

SPEAKER SIMILARITY

Speaker similarity matters (targeted)

REFERENCE

Presentation Attack (ISO/IEC 30107)

Speech deepfakes

TARGET

Target is a human listener

SPEAKER SIMILARITY

In some cases, speaker similarity does not matter (untargeted)

DEFINITION

Deepfakes are synthetic media that have been digitally manipulated to replace one person's likeness convincingly with that of another.

Voice Spoofing (Human Target)

A scammer pretends to be someone we trust – a government agency, police, a business we recognize, or a family member (speaker similarity matters in this case).

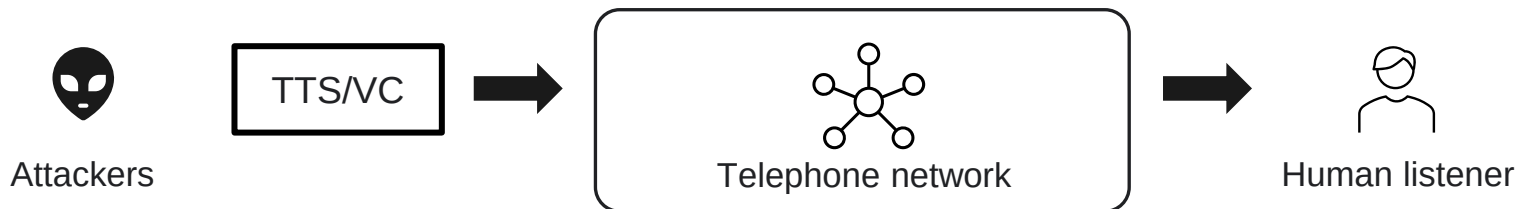
Untargeted spoofing

Claiming “I am an authentic human being”

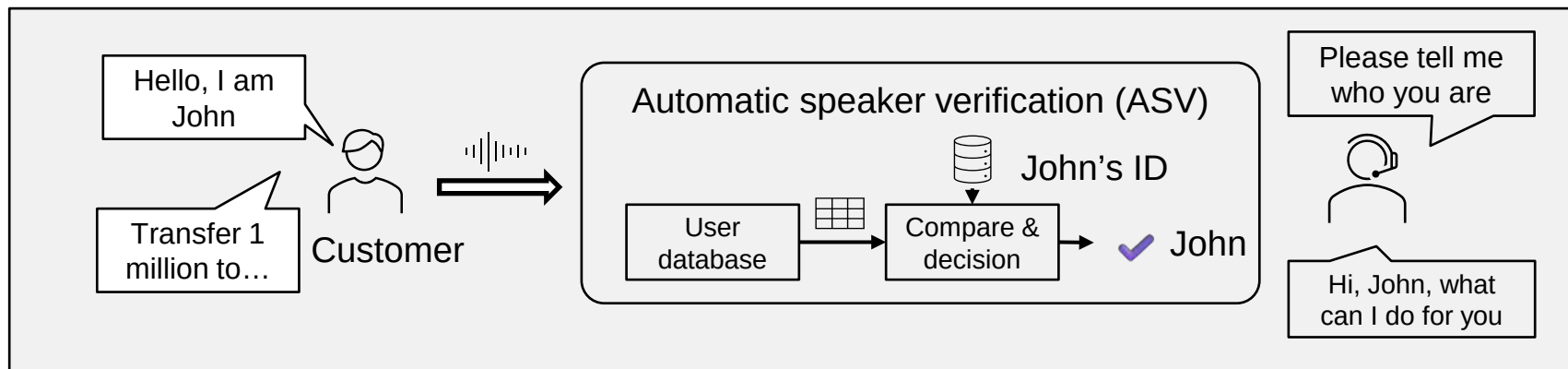
Targeted spoofing

Claiming “I am Person X”

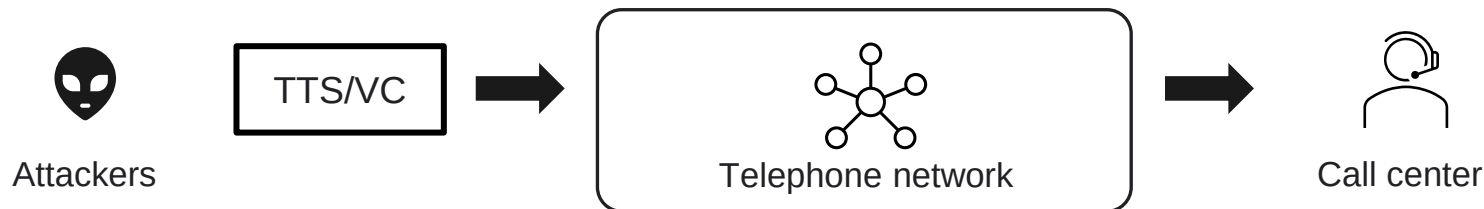
ATTACK FLOW



Voice Spoofing (ASV Target)

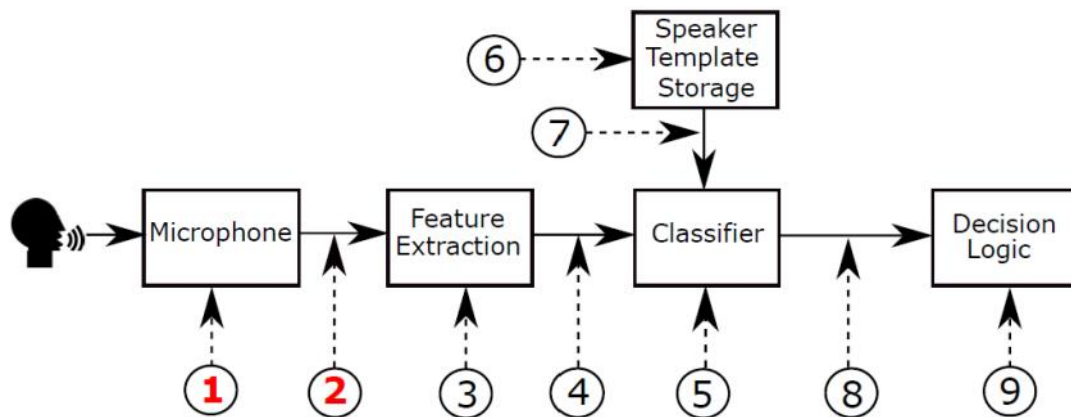


ATTACK FLOW



Presentation Attack Points

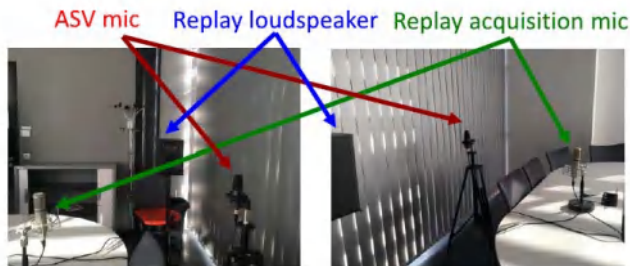
- ISO/IEC 30107 identifies the possible locations of presentation attack points in a typical ASV system [Md Sahidullah et al, 2023].
- The most vulnerable places in an ASV system are marked by 1 and 2, corresponding to **physical access** (PA) and **logical access** (LA).



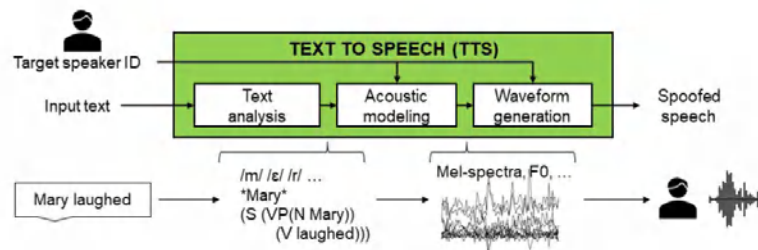
Physical and Logical Access Scenarios

- **Physical access (PA)** – spoofing attacks (e.g., replay of a bona fide speech) are presented to the microphone (Point 1) of the ASV system placed in a physical space. Room acoustic counts.
- **Logical access (LA)** – a remote user seeks access to a service protected by an ASV system, by injecting a synthesized or converted speech to the feature extractor (Point 2).

Physical Access (PA)



Logical Access (LA)



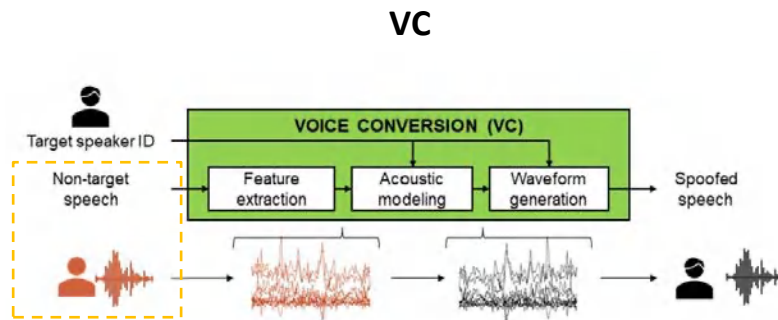
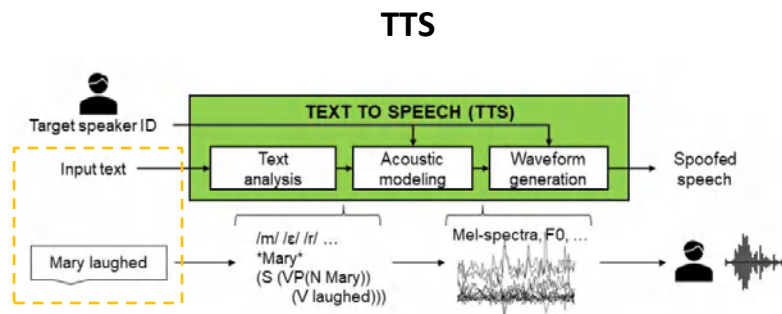
A. Nautsch *et al.*, "ASVspoo 2019: Spoofing Countermeasures for the Detection of Synthesized, Converted and Replayed Speech," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 252–265, Apr. 2021.

Presentation Attack Types (1/2)

- Four well-known ASV spoofing techniques are (1) **voice mimicry**, (2) **replay**, (3) **text-to-speech synthesis (TTS)**, and (4) **voice conversion (VC)**.
- **Voice mimicry attacks**
 - An impersonator intentionally modifies his or her speech to sound like the target speaker.
 - Likely to copy lexical, prosodic and idiosyncratic behaviour of target speakers.
- **Replay attacks**
 - The use of pre-recorded speech from a target speaker, which is then replayed through some playback device to feed the system microphone.
 - Require no specific expertise nor sophisticated equipment. Relatively low-tech attack within the grasp of any potential attacker even without specialised knowledge in speech processing.

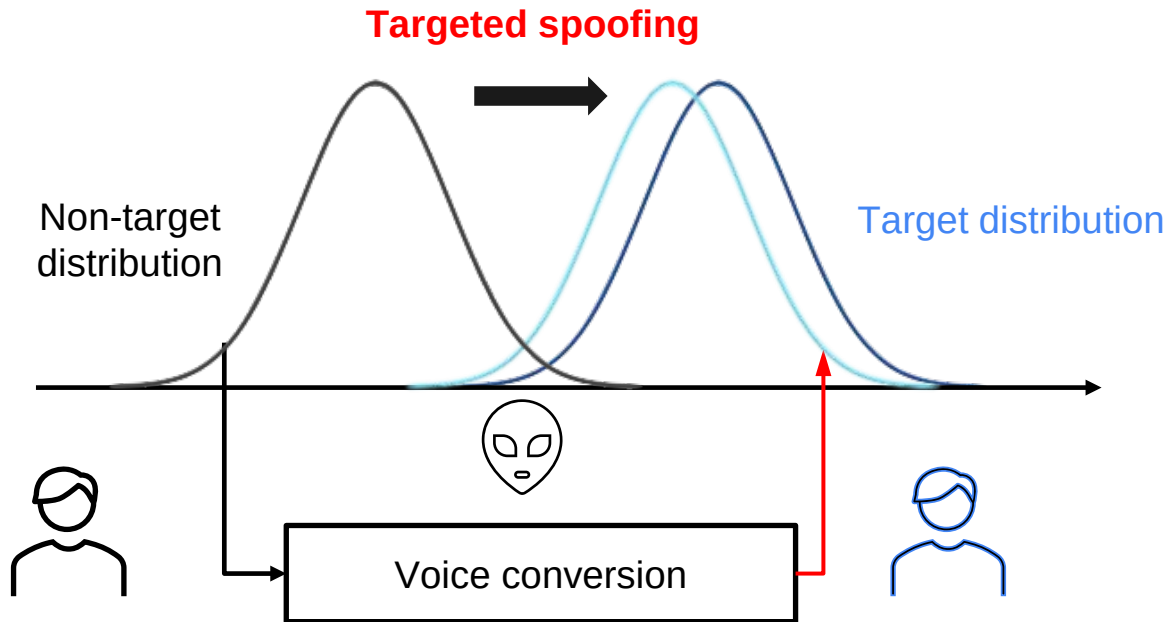
Presentation Attack Types (2/2)

- **Text-to-speech synthesis (TTS)** – generates intelligible, natural sounding artificial speech for any arbitrary text in target speaker's voice.
- **Voice conversion (VC)** – converts a natural speech (uttered by an attacker or other) towards a target speaker's voice.



Spoofing Attacks on ASV

- Presentation attacks shift imposter scores toward the target distribution, increasing the false acceptance rate (FAR).



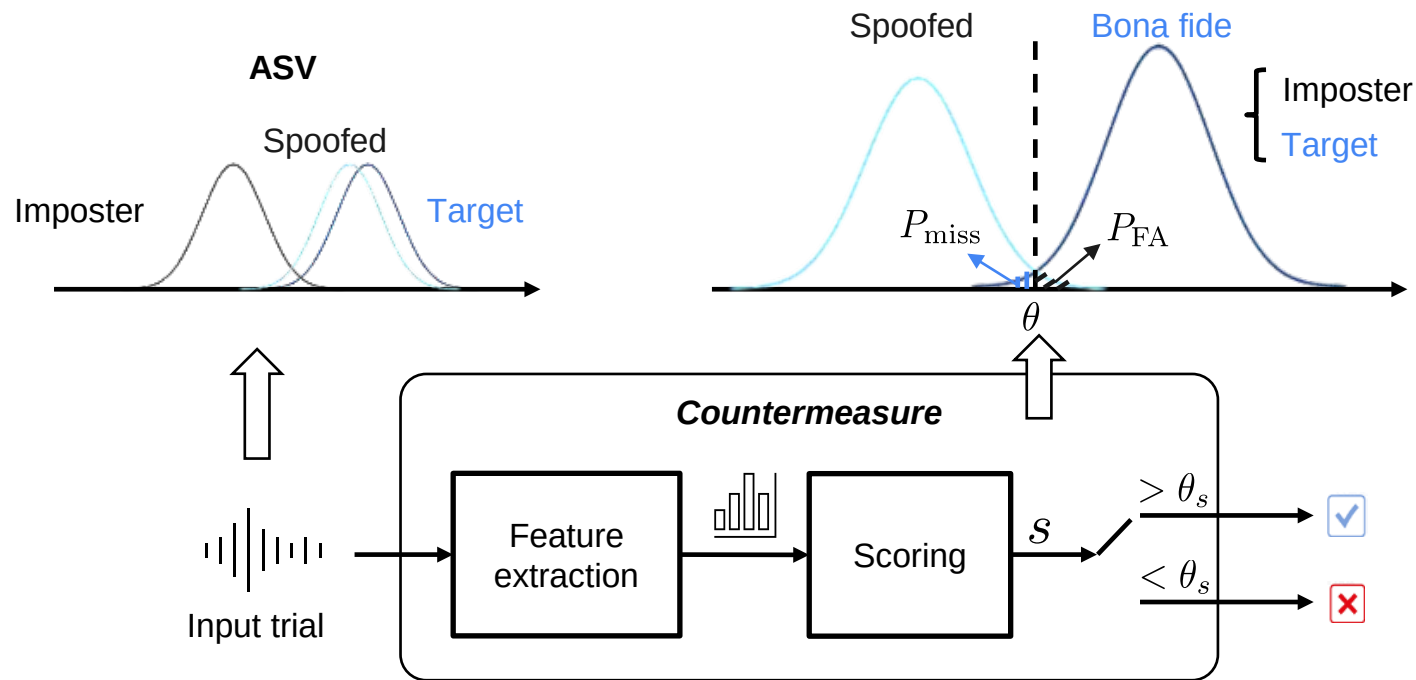
Voice Anti-Spoofing – Bonafide or Spoofed

- Voice anti-spoofing (speech anti-spoofing, or spoofing countermeasure) encompasses
 - Presentation attack detection (ISO/IEC 30107-1 standard)
 - Speech deepfake detection
- A spoofing countermeasure is formally defined as a binary classification problem:

Given an input waveform $\mathbf{o}_{1:T} = (o_1, o_2, \dots, o_T) \in \mathbb{R}^T$ with T sampling points, a PAD model is a function f with a parameter set $\boldsymbol{\theta}$ that conducts

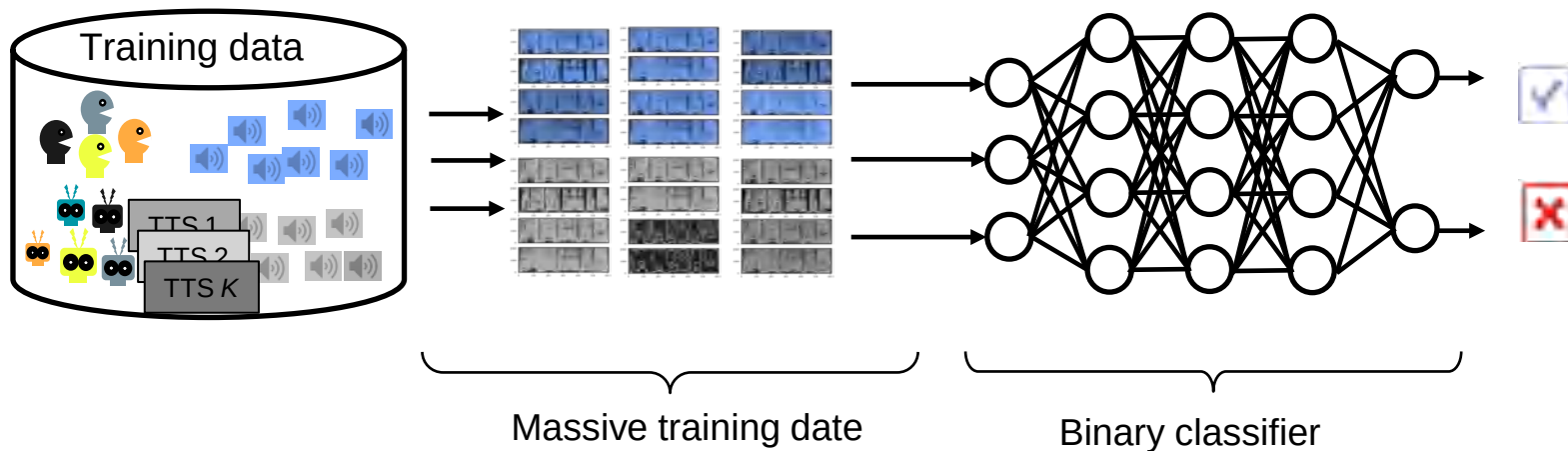
$$f_{\boldsymbol{\theta}} : \mathbb{R}^T \rightarrow \{\text{Bona fide, spoofed}\}$$

ASV with Spoofing Countermeasure

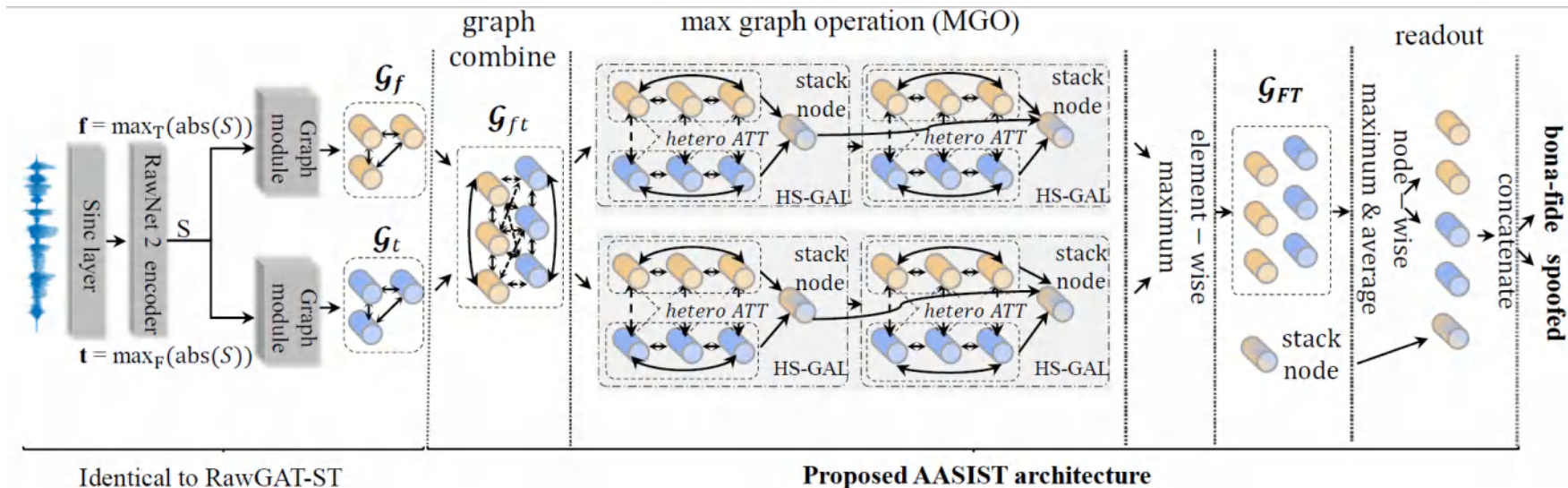


Methodology for Voice Anti-Spoofing

- Artifact detection
 - Massive amount of data + machine learning
 - End-to-end binary classification (bonafide versus spoofed)



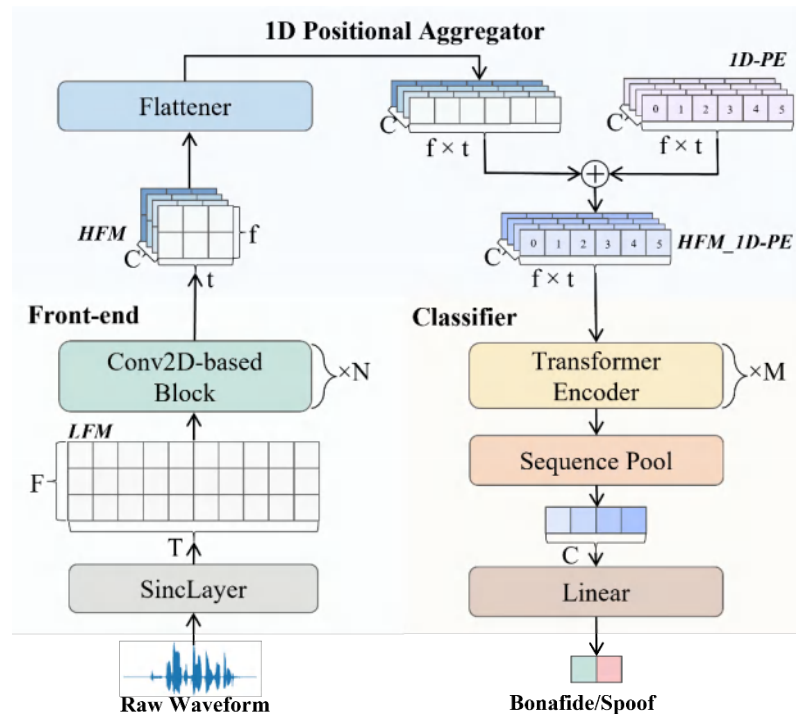
Method 1: Graph Attention with AASIST



J.-w. Jung, H.-S. Heo, H. Tak, H. Shim, J. S. Chung, *et al.*, "AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *Proc. IEEE ICASSP*, 2022, pp. 6367–6371.

Method 2: RawFormer

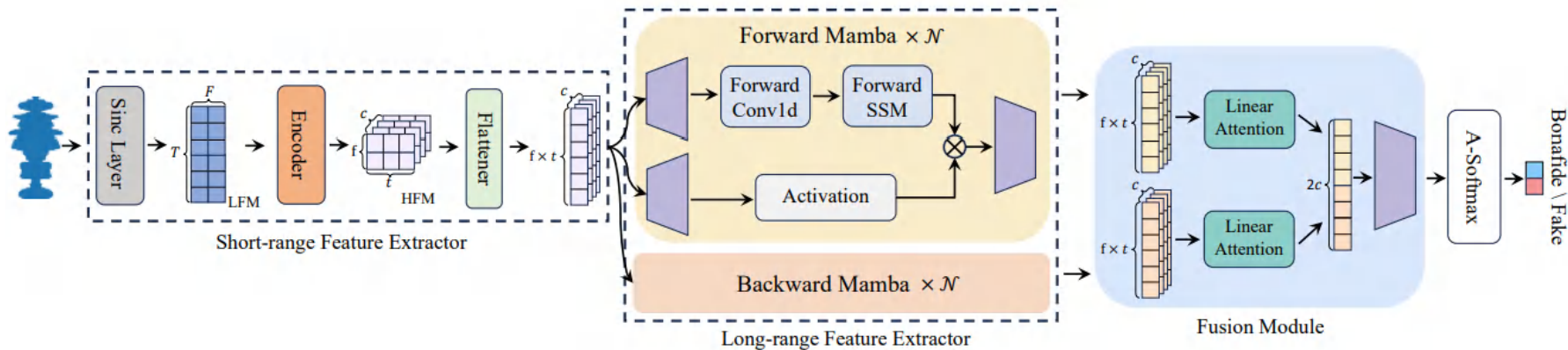
- Rawformer with 1D PE
 - A RawNet2 front-end (SyncLayer followed by 2D convolution blocks) to learn high-level feature maps (HFM)
 - HFM is flattened along the time then frequency
 - 1D positional encoding (PE) to preserve the spectro-temporal positional information
 - Transformer to capture local-global dependency more efficiently



X. Liu, M. Liu, L. Wang, K. A. Lee, H. Zhang and J. Dang, "Leveraging Positional-Related Local-Global Dependency for Synthetic Speech Detection," in *Proc. IEEE ICASSP*, 2023, pp. 1-5.

Method 3: RawMamba

- Similar to the RawFormer in the front-end. Replace the Transformer with MAMBA.

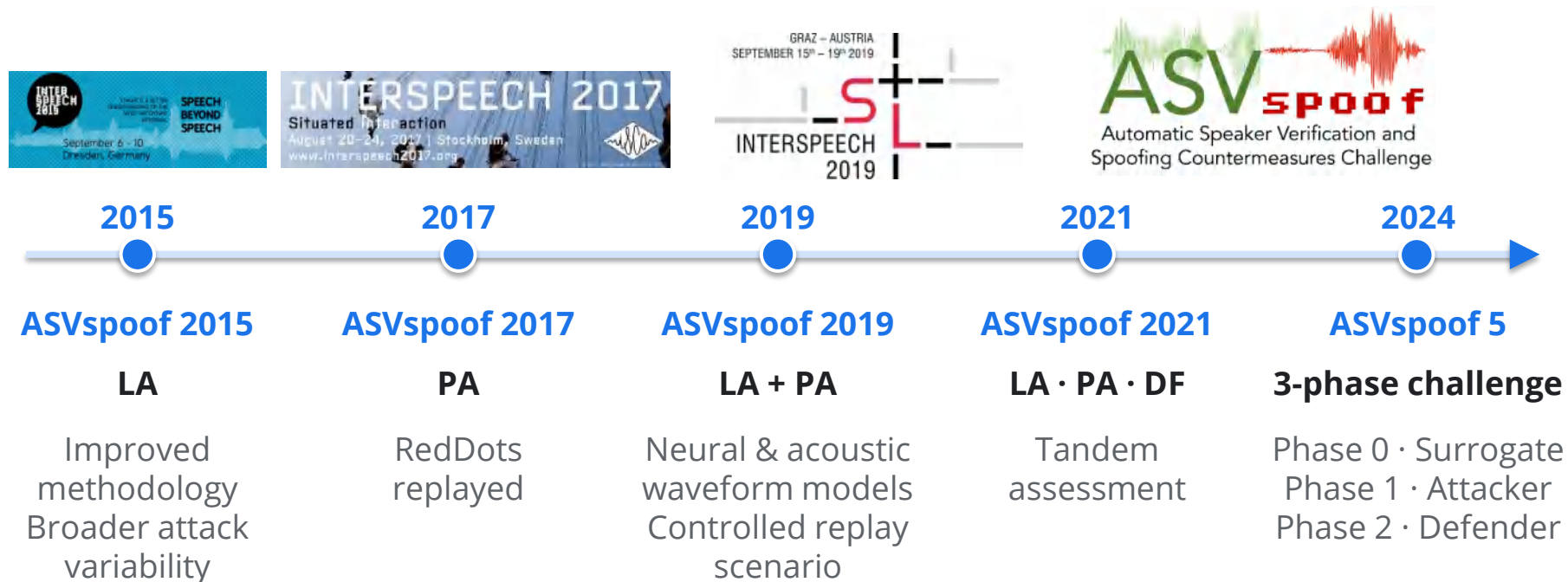


Y. Chen, J. Yi, J. Xue, C. Wang, X. Zhang, S. Dong, S. Zeng, J. Tao, Z. Lv, and C. Fan, "RawBMamba: End-to-End Bidirectional State Space Model for Audio Deepfake Detection," in *Proc. Interspeech*, 2024, pp. 2720–2724.

Model Comparison

Method	AASIST	RawFormer	RawMamba
Front-end extractor	Sinc layer + RawNet2	Sinc layer + RawNet2	Sinc layer + RawNet2
Feature processing	Separate temporal and spectral graph modules	Flattens high-level feature maps across time and frequency	RawFormer-style high-level feature flattening
Core classifier	Heterogeneous graph attention	Transformer encoder	Bidirectional MAMBA
Key innovation	Joint spectro-temporal artifact modeling via graph attention	Captures local-global positional dependencies with Transformers	Efficient long-range sequence modeling without a Transformer

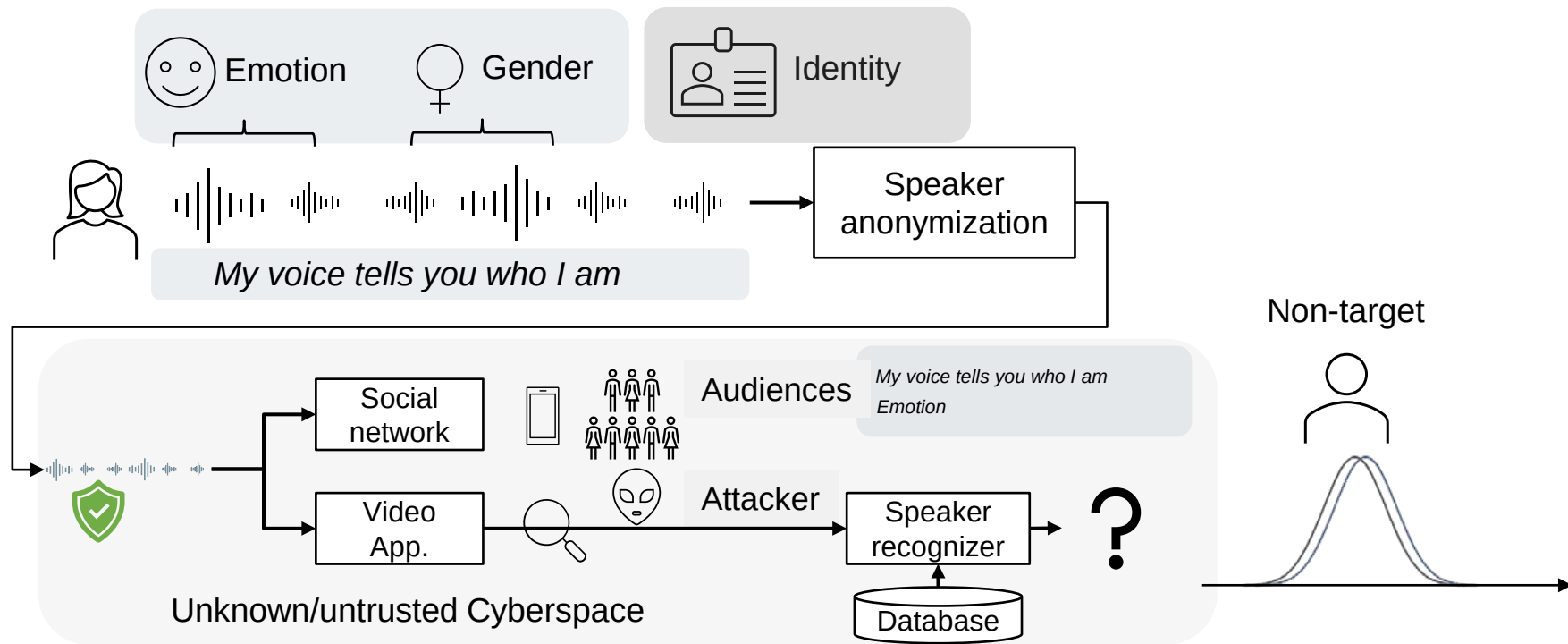
ASVspoof Challenges [www.asvspoof.org]



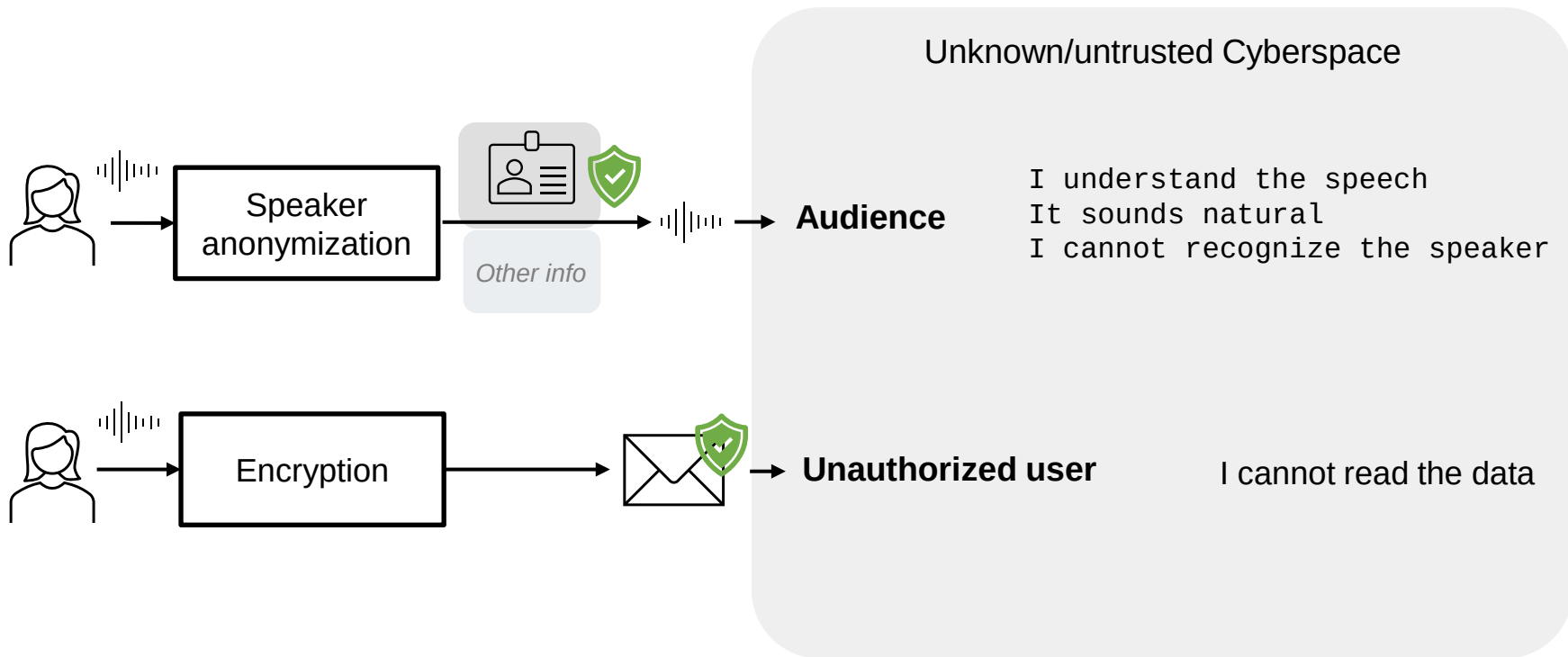
Section 3

Speaker Privacy Preservation

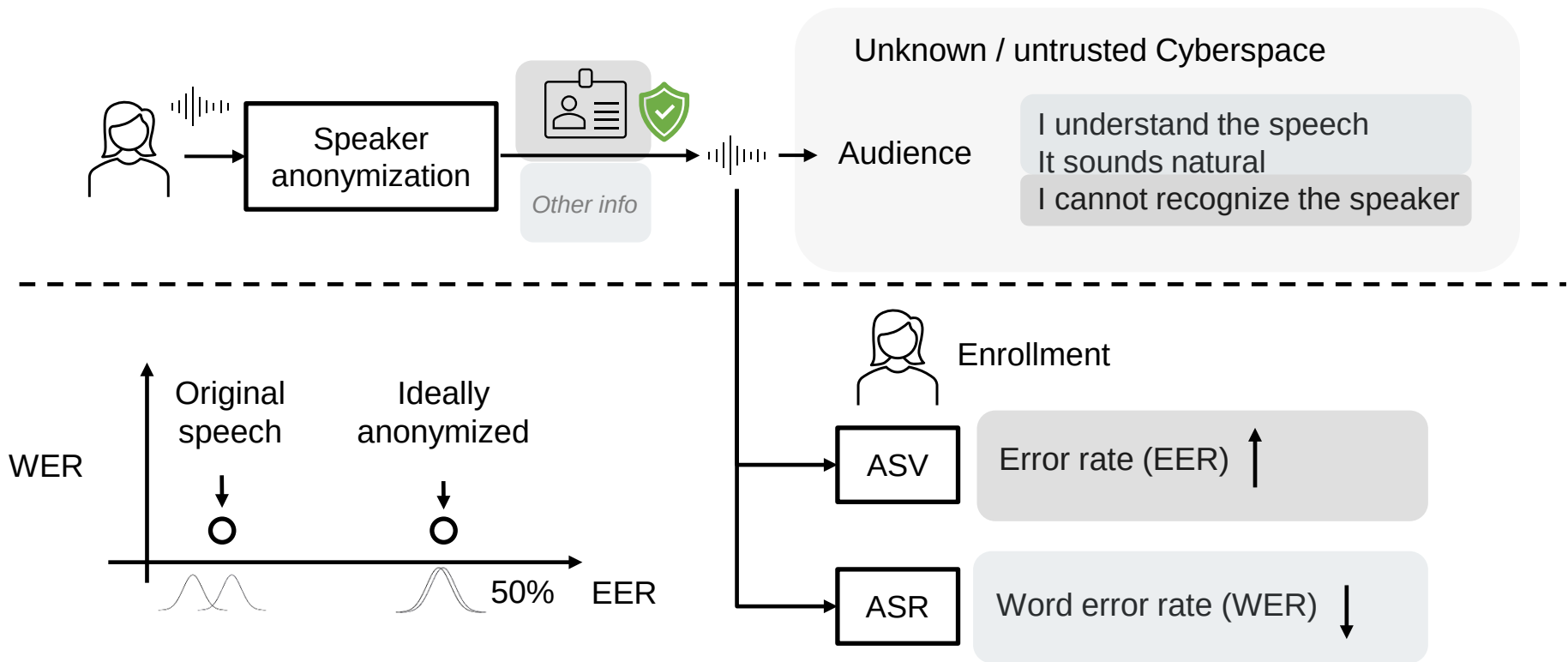
Speaker Anonymization – Mitigates Inference Risk



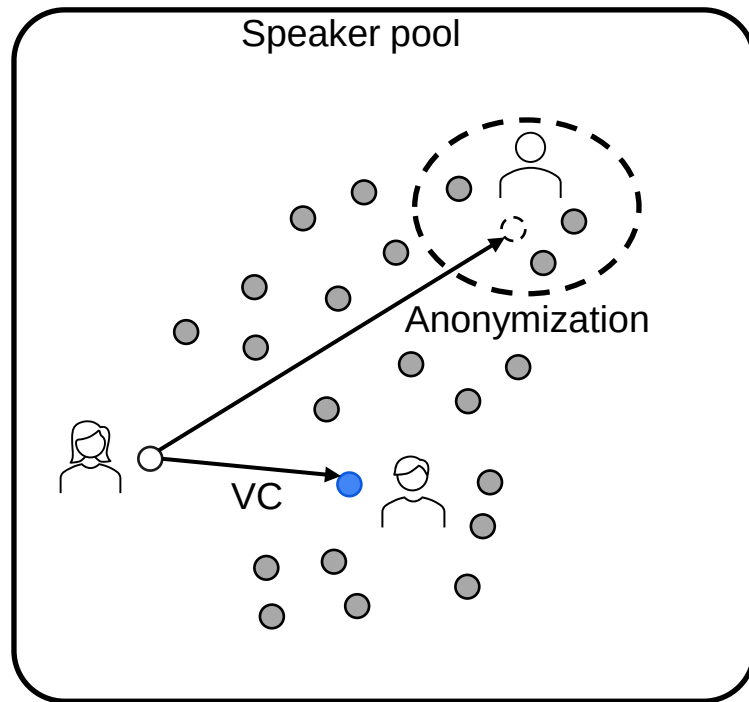
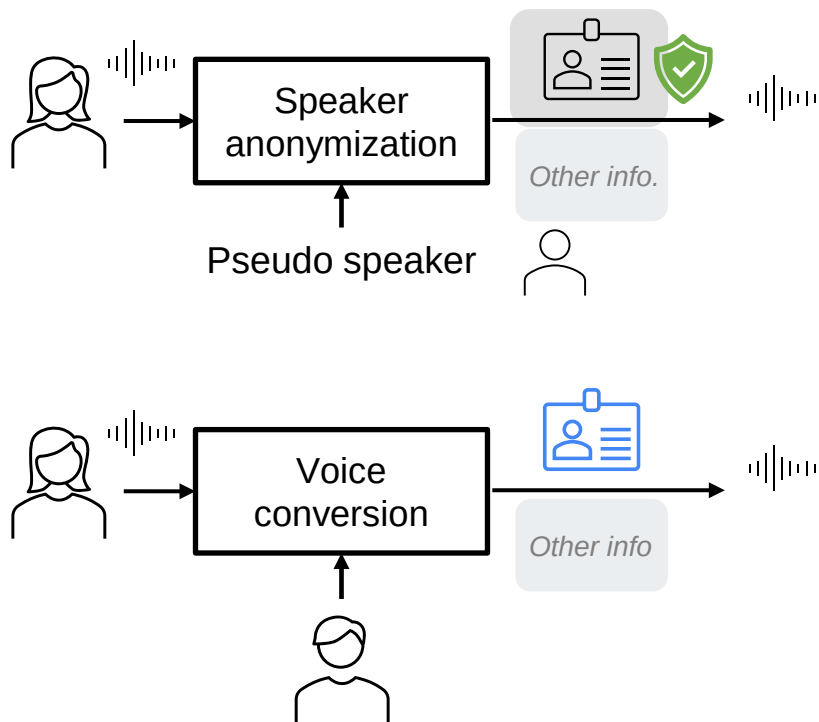
Speaker Anonymization (De-identification, Voice Sanitization)



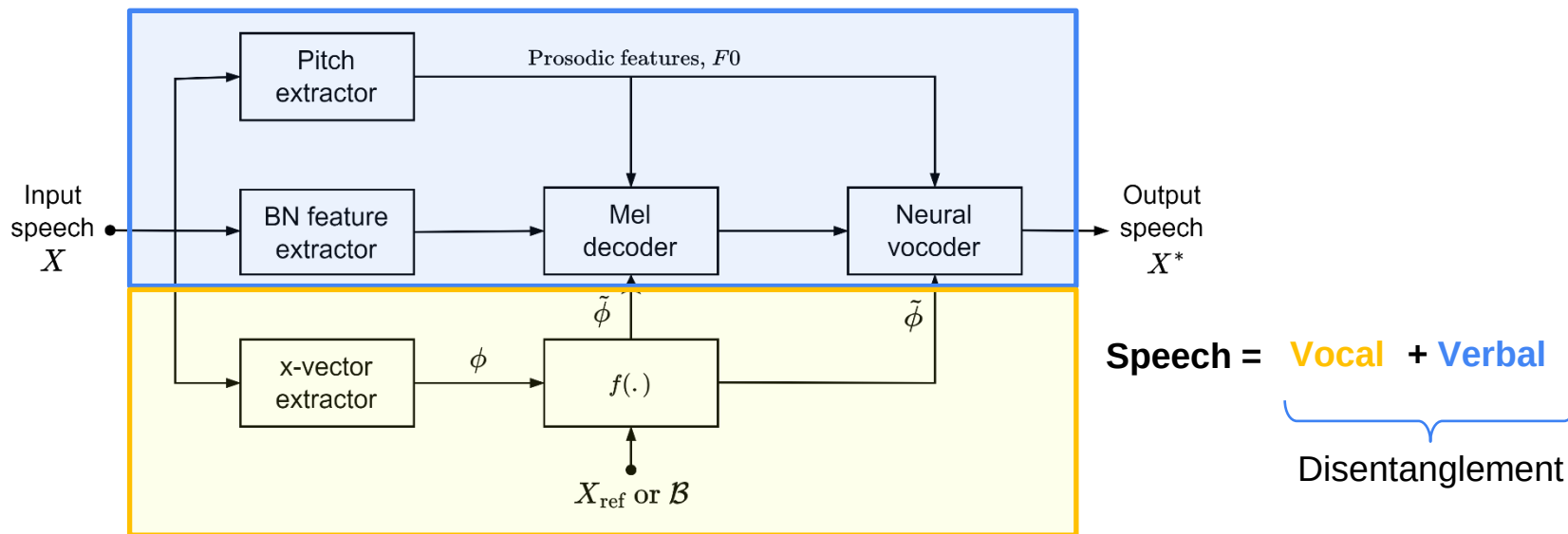
Speaker Anonymization – Metrics



Speaker Anonymization Methods



Speaker Anonymization Methods



N. Tomashenko, X. Wang, X. Miao, H. Nourtel, P. Champion, M. Todisco, E. Vincent, N. Evans, J. Yamagishi, and J.-F. Bonastre, "The VoicePrivacy 2022 Challenge evaluation plan," *arXiv preprint arXiv:2203.12468*, 2022.

Defining Speaker Anonymization

- Let X be a speech signal and X_v be the voice attributes in X that leads to the inference of the speaker identity Y_{ID} :

$$X_v \rightarrow Y_{ID}$$

- Speaker anonymization encompasses a mapping between the **input** X and the **anonymized speech** ($X \setminus X_v$)

$$f: X \mapsto X \setminus X_v$$


Voice attributes

- A speech signal cannot exist without a speaker's voice as a carrier of spoken words. In practice, the mapping is realized as

$$f': X \mapsto (X \setminus X_v) \cup X_{pseu}$$


Pseudo speaker

Pseudo Speakers

- **Pseudo-speakers** are artificial speakers created by algorithms that are not linked to any real person, thereby avoiding infringing the privacy of real individuals.
- Another important consideration is whether to use the same pseudo-speaker across different anonymization instances.
 - **Any-to-one** (a2o) mapping (one common pseudo-speaker for all utterances)

$$f': X_i \mapsto (X_i \setminus X_v^{(i)}) \cup X_{\text{pseu}}$$

- **Any-to-any** (a2a) mapping (distinct pseudo-speaker to each utterance)

$$f': X_i \mapsto (X_i \setminus X_v^{(i)}) \cup X_{\text{pseu}}^{(i)}$$

Pinhole Effect in Speaker Anonymization

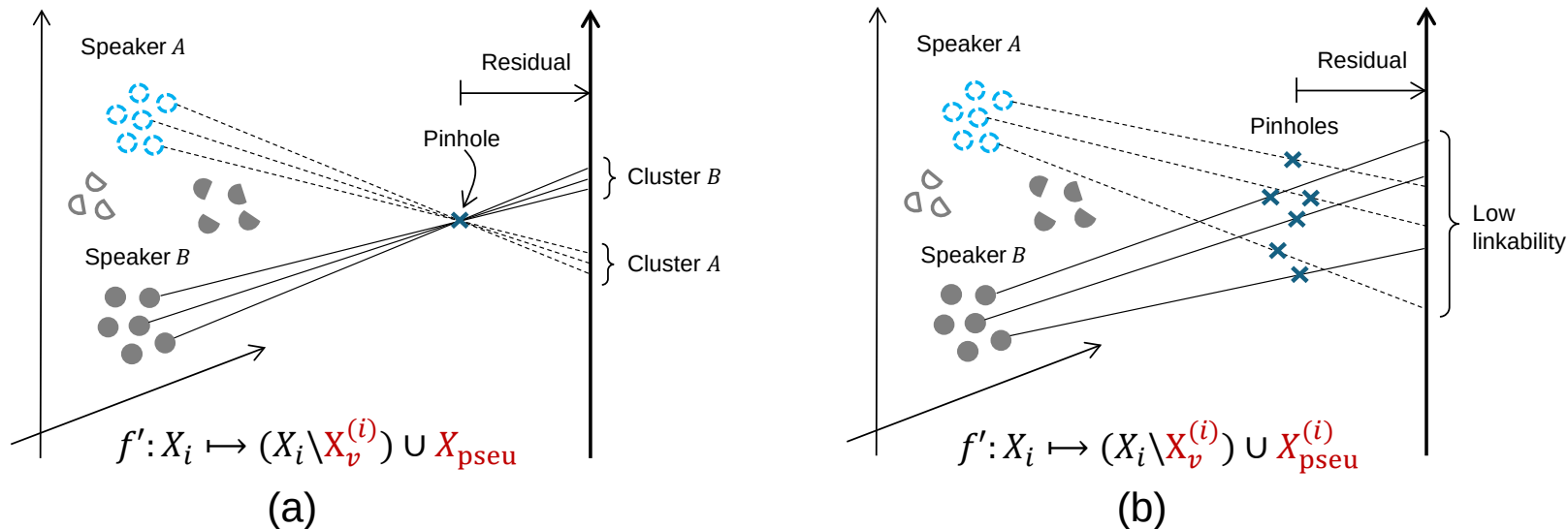
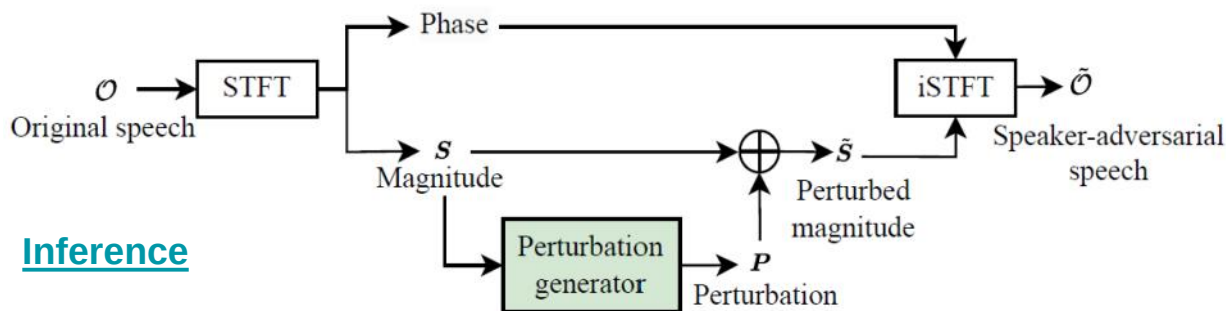


Figure: Pinhole effect in speaker anonymization: (a) a common pseudo-speaker is used across all anonymization instances, resulting in any-to-one (a2o) mappings, and (b) distinct pseudo-speakers are used for each anonymized utterance, resulting in any-to-any (a2a) mappings.

Adversarial Approach to Speaker Anonymization

- An **adversarial perturbation** is generated and added to the input speech to conceal speaker attributes, thereby anonymizing the speech.
- The perturbation is performed in the frequency domain, through short-time Fourier transform (STFT) and inverse STFT (iSTFT).



Closing Remarks

- 01 Speaker models support more than recognition.
- 02 System choices shape real-world performance.
- 03 Privacy and security guide responsible deployment.



Shuai Wang



Quan Wang



Kong Aik Lee



Slides

Thank you. Questions and discussion.