Deep Speaker Representation Learning

Theory, Applications, and Practice

Shuai Wang

APSIPA ASC 2025

About the Speaker

Shuai Wang
School of Intelligence Science and Technology
Nanjing University

- Research: Speaker Modeling, Speech& Music Generation
- Open-source: WeSpeaker, WeSep, West, DiffRhythm, SongBloom ...
- Email: shuaiwang@nju.edu.cn
- Website: https://shuaiwang-nju.github.io/

Outline

南京大學 NANJING UNIVERSITY
智能科学与技术学院 School of Intelligence Science and Technology

Speaker Modeling

What is Speaker Modeling?

Definition

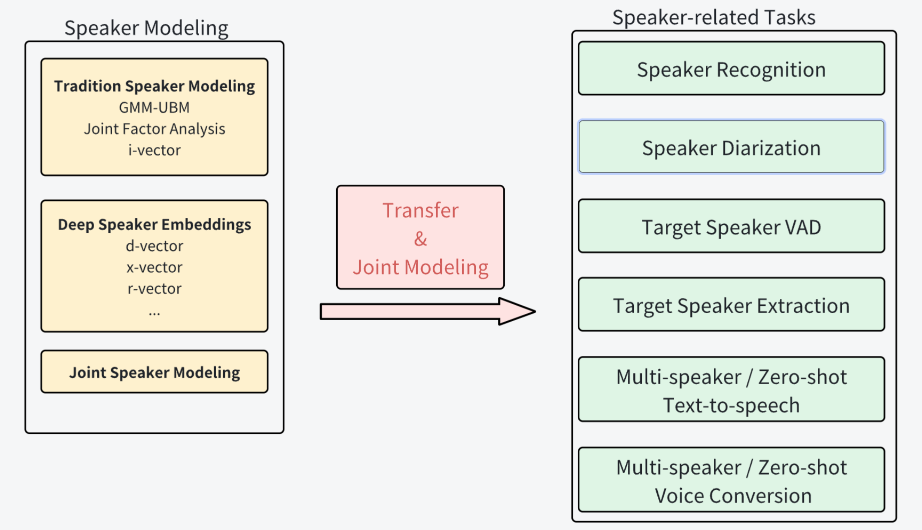
Speaker modeling aims to characterize and recognize an individual's unique traits by analyzing patterns embedded in speech signals.

Main Applications:

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

Real-world Impact:

- Biometric authentication
- Surveillance systems
- Personalized services
- Forensic analysis
- Privacy protection


Speaker Modeling
Tradition Speaker Modeling
GMM-UBM
Joint Factor Analysis
i-vector
Deep Speaker Embeddings
d-vector
x-vector
r-vector
...
Joint Speaker Modeling
Transfer
&
Joint Modeling
Speaker-related Tasks
Speaker Recognition
Speaker Diarization
Target Speaker VAD
Target Speaker Extraction
Multi-speaker / Zero-shot
Text-to-speech
Multi-speaker / Zero-shot
Voice Conversion

Speaker verification: voice as a password

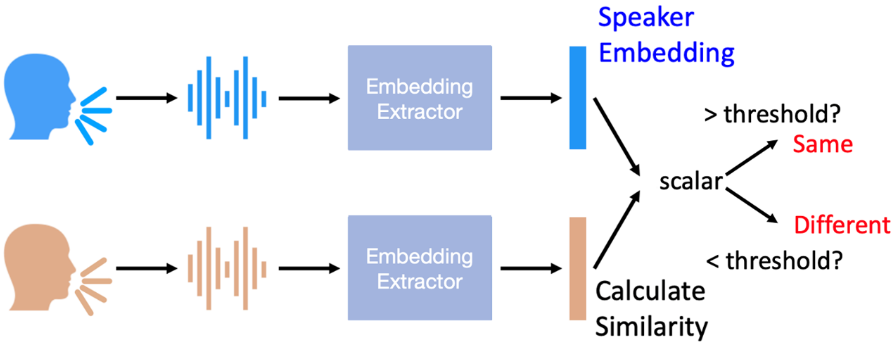
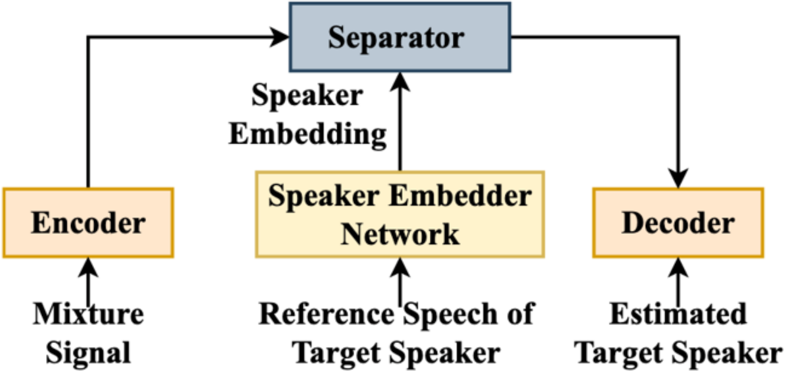
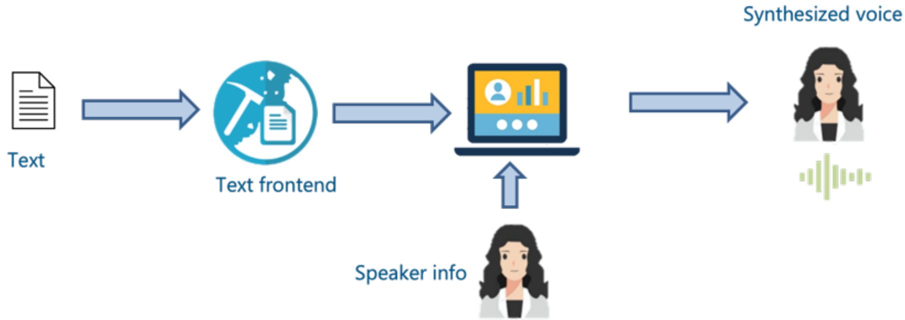


Figure adapted from Hung-Yi Lee's DLHLP20 slides[1]

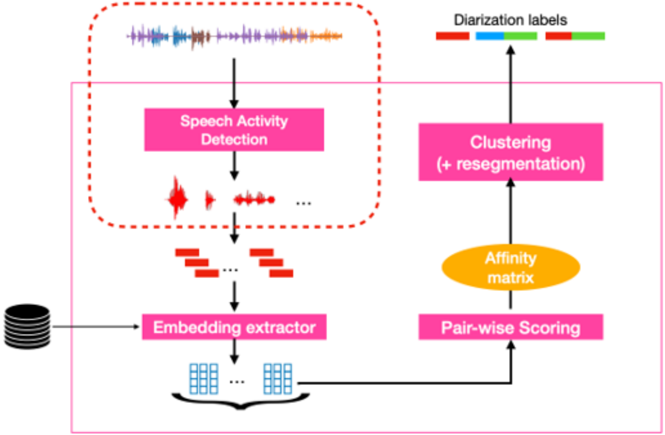
Target speech extraction: listen to the target person



Speaker modeling for the target speaker in speech synthesis.



Speaker diarization: who spoke when?



4

Historical Evolution: Paradigm Shifts

1. **From VQ to GMM** (1980s–1990s)
   - Vector quantization $\rightarrow$ 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

Era 1: From VQ to GMM (1980s–1990s)

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.

Key Innovations

- Universal Background Model (UBM): speaker-independent acoustic space shared by all speakers.
- MAP adaptation: derive speaker models from UBM using limited enrollment data (with relevance factor control).
- Likelihood-ratio scoring: $\log p(\mathbf{X} \mid \text{spk}) - \log p(\mathbf{X} \mid \text{UBM})$ for verification.

Practical Impact

- Robust under scarce enrollment data; improved cross-channel generalization.
- Established reproducible, scalable baselines for large-scale evaluations (telephone, microphone speech).

Speaker Modeling

GMM-UBM & GMM Supervector

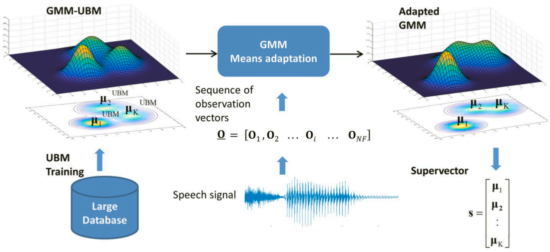


Figure: GMM-UBM[a] for speaker modeling

[a]Zheng, Zhang, and Xu, "Text-independent speaker identification using gmm-ubm and frame level likelihood normalization".

Gaussian Mixture Model (GMM)[a]

- $p(\mathbf{x}) = \sum_1^K c_k \, \mathcal{N}\left(\mathbf{x} \mid {}_k, {}_k\right)$ s.t. $\sum_1^K c_k = 1$
- Any distribution can be approximated by a weighted linear combination of several Gaussians.
- When modeling speaker acoustics with GMMs, the number of Gaussians can be viewed as types of produced sounds.

Universal Background Model (UBM)

- Enrollment speech is often limited (a few seconds), making training a GMM difficult with such data.
- Train a UBM on large data and adapt it to specific speakers.

GMM Supervector

- Concatenate the mean vector of each Gaussian to represent a speaker.

[a]Reynolds et al., "Gaussian mixture models."

GMM-based speaker modeling – test scoring

- In GMM-UBM systems, a likelihood ratio is typically used for scoring a test utterance. Given an utterance Y, two hypotheses are:

$$H_0 : Y \text{ comes from target speaker } S$$
$$H_1 : Y \text{ does not come from target speaker } S$$

- The score Λ is determined by the log-likelihood ratio:

$$\Lambda = \frac{1}{T} \log \frac{p\left(Y \mid H_0\right)}{p\left(Y \mid H_1\right)} = \begin{cases} \geq \theta & \text{accept } H_0 \\ < \theta & \text{accept } H_1 \end{cases}$$

where T is the total number of frames of Y and θ is a preset threshold.

- Concretely, $p\left(Y \mid H_0\right)$ is the probability density of the features of Y on speaker S's GMM, and $p\left(Y \mid H_1\right)$ is that on the impostor model. In GMM-UBM systems, the UBM serves as the impostor model.

Era 3: From Supervector to i-vector (2010s)

From High- to Low-dimensional Space

- Supervector SVM: concatenate adapted GMM means into a very high-dimensional vector; powerful but channel-sensitive.
- Total-variability (T) model: joint factor analysis; utterance-level latent variable (i-vector) summarizing speaker and channel.
- i-vector extraction: posterior inference in $\mathcal{N}(\mathbf{m} + \mathbf{T}\mathbf{w}, \Sigma)$ to produce compact $\mathbf{w} \in \mathbb{R}^d$.

Back-ends and Compensation

- Length normalization, LDA/WCCN for inter-/intra-class variance control.
- PLDA or cosine scoring for calibrated verification.

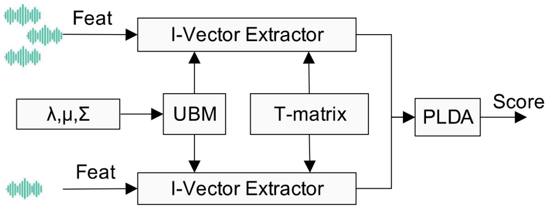


Figure: Block diagram of an i-vector based speaker recognition system

Drawbacks of GMM Supervector

- Supervectors are extremely high-dimensional (often tens of thousands), making computation challenging.

Factorize the supervector into a low-dimensional i-vector[a]:

$$M(s) = m + T\,w(s)$$

- $M(s)$: GMM supervector of speaker s
- m: speaker-independent supervector
- T: total-variability matrix capturing all variability sources (speaker- and channel-related)
- $w(s)$: i-vector of speaker s

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

i-vector Extraction

1. **UBM training**: train the universal background model with large data
2. **T-matrix training**: learn the total-variability matrix $\mathbf{T}$
3. **Posterior inference**: compute $\mathbf{w} = E[\mathbf{w}|\mathbf{X}]$
4. **Length normalization**: $\mathbf{w}_{norm} = \frac{\mathbf{w}}{||\mathbf{w}||}$

Back-end Compensation

- **LDA**: linear discriminant analysis
  - Maximize between-class variance
  - Minimize within-class variance

Scoring

- **PLDA**: probabilistic linear discriminant analysis
  - Accounts for within-speaker variability
  - Provides a probabilistic interpretation

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).

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

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

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.

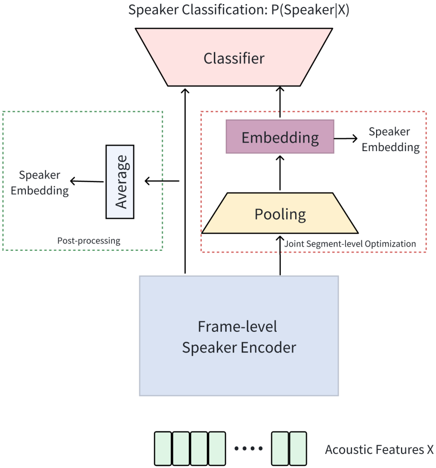
Frame-level (d-vector)

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

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

Segment-level (x-vector)

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



Speaker Encoder Architectures

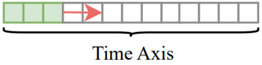
1D vs. 2D Convolution

1D Convolution

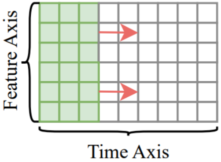
- Applied along time
- Lower computational cost
- Potentially large receptive field
- Simple architecture
- Limited frequency modeling

2D Convolution

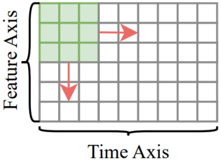
- Applied along time and frequency
- Better time-frequency modeling
- Higher computational cost
- Better performance potential



a) 1D Convolution for raw wav input



b) 1D Convolution for spectrogram input



c) 2D Convolution for spectrogram input

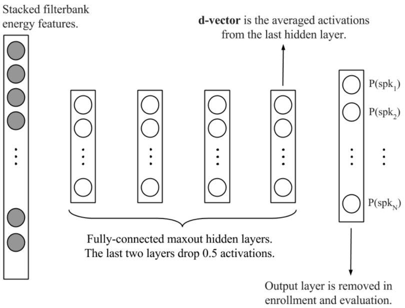


Figure: Architecture of d-vector

Labels for d-vector[a]:

- Early attempt to apply DNNs to speaker information modeling
- Demonstrated good complementarity to i-vector
- Frame-level embeddings averaged to utterance vector; trained with CE on speaker IDs
- Typical encoders: TDNN/LSTM/CNN with temporal mean pooling (no attentive statistics)
- Pros: simple training, suitable for short utterances, low latency
- Limitations: phonetic content leakage; weaker than x-vector for segment aggregation

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

Speaker Encoder Architectures

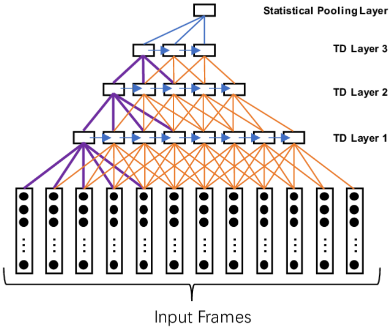


Figure: TDNN architecture used by x-vector

Labels for x-vector[a]:

- First deep speaker embedding surpassing traditional methods on standard datasets (NIST SRE)
- First to introduce segment-level optimization
- Strong variant: ECAPA-TDNN[b]

Temporal pooling:

$$= \frac{1}{T}\sum_{t=1}^{T}\mathbf{h}_t \quad (\mathbf{h}_t \in \mathbb{R}^D)$$

Statistics pooling:

$$= \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\mathbf{h}_t - \ )^{\odot 2}}$$

$$\mathbf{v} = [\ ;\ ] \in \mathbb{R}^{2D}$$

---

[a] Snyder et al., "X-vectors: Robust dnn embeddings for speaker recognition".

[b] Desplanques, Thienpondt, and Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification".

Labeled data:

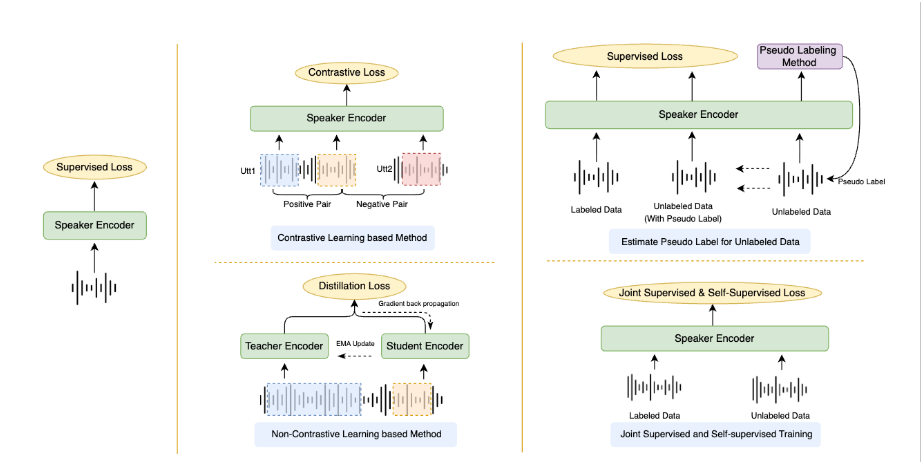
- Expensive to annotate
- Automatically collected speaker-labeled data such as VoxCeleb has privacy concerns
  - The VoxCeleb dataset is no longer accessible from the official website

Unlabeled data:

- Easy to obtain
- Covers a broader range of real-world conditions
- Less privacy concern

Speaker Modeling

Training paradigms: from supervised to unsupervised/semi-supervised/self-supervised



(a) Supervised Training (b) Self-supervised Training (c) Semi-supervised Training

- GMM and i-vector can be seen as one-layer MLPs
- d-vector, j-vector, x-vector: fewer than 10 layers
- ResNet-based models (common: 34 layers; up to 293 or 500+ in challenges)

- Pure audio modality
- Audio-visual speaker embeddings

- Train speaker-discriminative models from scratch
- Leverage large pretrained speech models such as WavLM
- Semi-supervised: iterative clustering and supervised fine-tuning

- Pretrained embeddings used across tasks
- Explicit joint optimization with task-specific objectives
- Implicit speaker modeling in related tasks

Challenges in Speaker Modeling

Technical Challenges:

- Channel mismatch
- Language dependency
- Short utterance handling
- Computational efficiency
- Model interpretability

Practical Challenges:

- Data privacy concerns
- Real-time processing
- Scalability issues
- Cross-domain generalization
- Evaluation standardization

Future Directions

- **Large-scale Pre-training**: Foundation models for speaker modeling
- **Few-shot Learning**: Rapid adaptation with limited data
- **Multimodal Integration**: Audio-visual speaker modeling
- **Edge Computing**: Efficient deployment on mobile devices
- **Federated Learning**: Privacy-preserving distributed training
- **Explainable AI**: Interpretable speaker modeling systems

Outline

Problem setting

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

- 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.

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.

Model

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

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

$$\mathcal{L}_{\mathsf{CE}} = -\log \frac{e^{s_y}}{\sum_{j=1}^{C} e^{s_j}}.$$

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.

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).

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

Geometric view

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

Form

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

- 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.

Form

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

- 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.

Form

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

- 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).

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.

Form

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

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

- Leverage large pretrained models
  - Self-supervised pretrained speech models
  - ASR model initialization
  - Efficient fine-tuning
- Self-supervised learning methods
  - SimCLR/MoCo/DINO
  - Stage-wise iterative training

- Self-supervised pretrained speech models
  - Wav2Vec[a]
  - HuBERT[b]
  - WavLM[c]
  - UniSpeech[d]

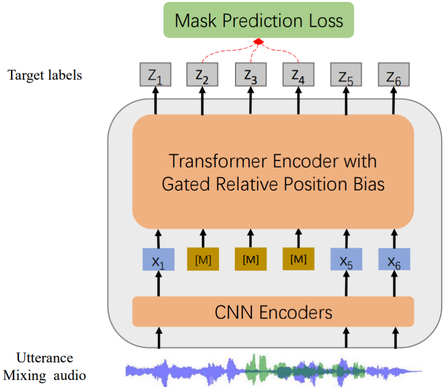


Figure: WavLM architecture

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

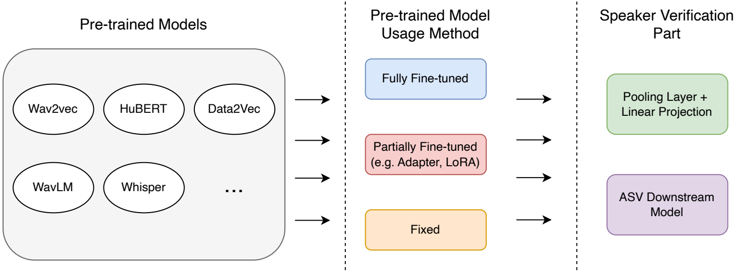
[b] Hsu et al., "Hubert: Self-supervised speech representation learning by masked prediction of hidden units".

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

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

Leveraging Pretrained Models

Integration Strategies of Large Pretrained Speech Models



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

Fine-tune **SSL speech models** on speaker verification[a]

- Replace Fbank with pretrained representations
- Learnable weighted aggregation across layers

[a] Chen et al., "Large-scale self-supervised speech representation learning for automatic speaker verification".

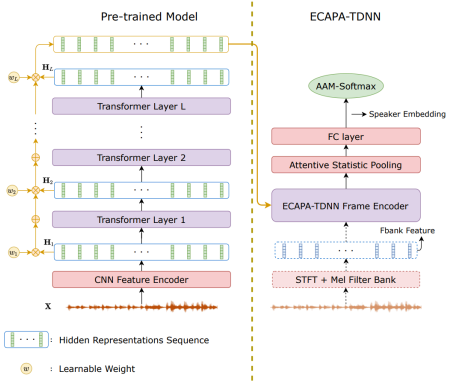
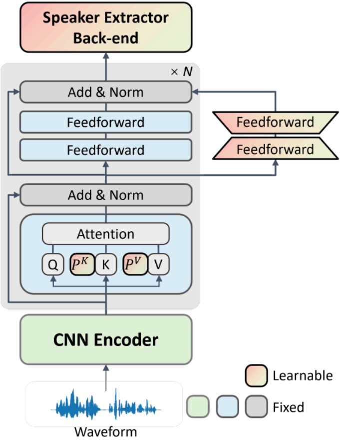


Figure: Using pretrained representations

Efficient fine-tuning for SSL models with adapters on speaker verification[a]

- Freeze large pretrained model
- Use adapters for parameter-efficient tuning



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

Adopt large SSL model w2v-BERT 2.0 for SOTA performance[2]

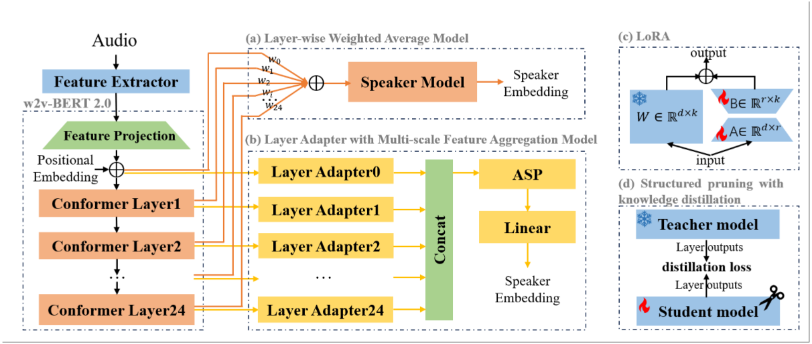


Figure: Enhancing SV with w2v-BERT 2.0 and distillation-guided structured pruning

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

Leveraging Pretrained Models

Fine-tune **ASR models** on speaker verification[a][b]

- Pretrain with ASR datasets
- Initialize for speaker tasks

[a] Liao et al., "Towards a unified conformer structure: from asr to asv task".
[b] Cai et al., "Pretraining Conformer with ASR for Speaker Verification".

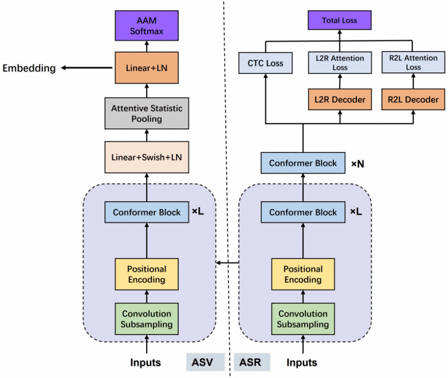


Figure: ASR transfer illustration

Metric learning losses provide contrastive supervision, e.g., Triplet, Prototypical, GE2E[3], and Angular Prototypical[4].

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

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

[3] Wan et al., "Generalized end-to-end loss for speaker verification".

[4] Chung et al., "In defence of metric learning for speaker recognition".

Assumption for SSL on speaker verification[a]

- Segments from the same utterance belong to the same speaker
- Segments from different utterances belong to different speakers

[a] Huh et al., "Augmentation adversarial training for self-supervised speaker recognition".

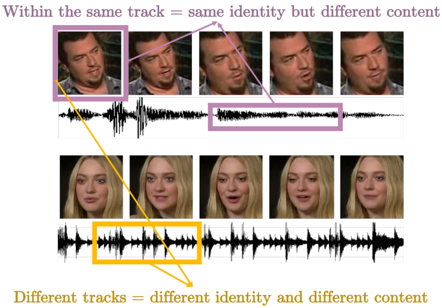


Figure: Illustration of the assumption

Based on SimCLR[a], adapted to speaker tasks[b]

- Crop two segments from an utterance to construct positive/negative pairs
- Use metric loss to attract positives and repel negatives

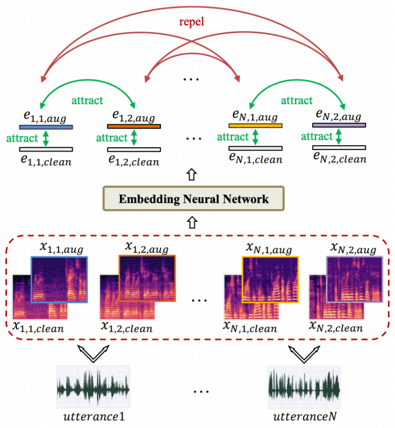


Figure: SimCLR for speaker task

[a] Chen et al., "A simple framework for contrastive learning of visual representations".

[b] Zhang, Zou, and Wang, "Contrastive self-supervised learning for text-independent speaker verification".

Based on DINO[a], adapted to speaker tasks[bc]

- Crop several segments from one utterance and build only positive pairs
- Use cross-entropy to attract positive pairs

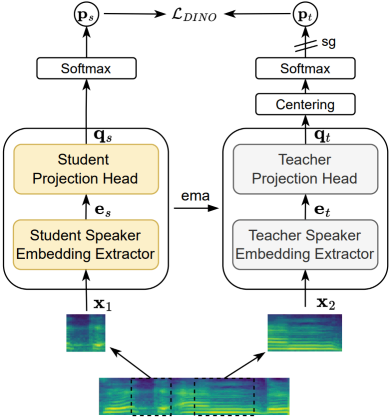


Figure: DINO for speaker task

[a] Caron et al., "Emerging properties in self-supervised vision transformers".

[b] Han, Chen, and Qian, "Self-supervised speaker verification using dynamic loss-gate and label correction".

[c] Chen et al., "A comprehensive study on self-supervised distillation for speaker representation learning".

Two-stage iterative framework[a,b,c]

- I: Contrastive training
- II: Iterative clustering and representation learning

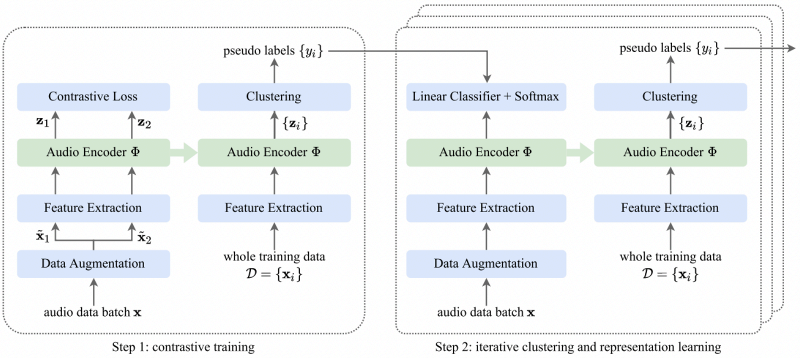


Figure: Iterative framework for SSL speaker verification

[a] Cai, Wang, and Li, "An iterative framework for self-supervised deep speaker representation learning".

[b] Han, Chen, and Qian, "Self-Supervised Learning with Cluster-Aware-DINO for High-Performance Robust Speaker Verification".

[c] Tao et al., "Self-supervised speaker recognition with loss-gated learning".

Outline

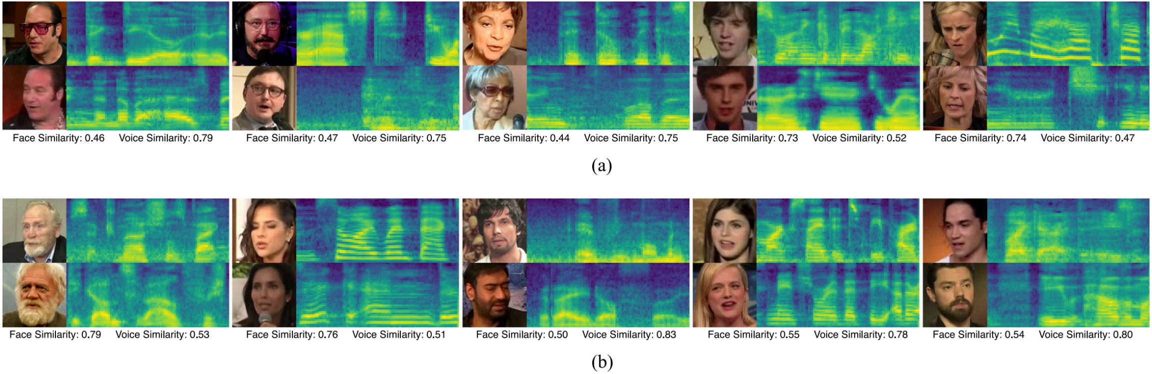


Figure: Speaker similarity from audio or visual information[5]

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

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

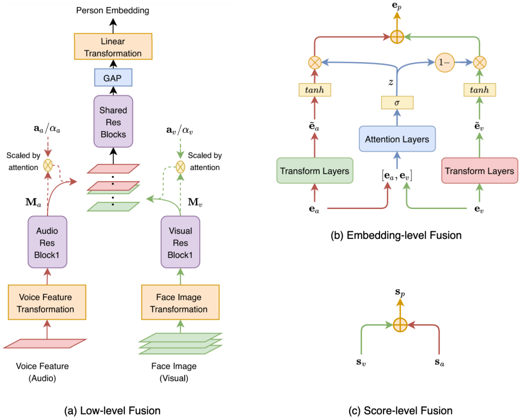


Figure: Different levels of audio-visual fusion[a]

- Embedding-level fusion outperforms low-level fusion
- Attention in embedding-level fusion is more noise-robust than score-level fusion

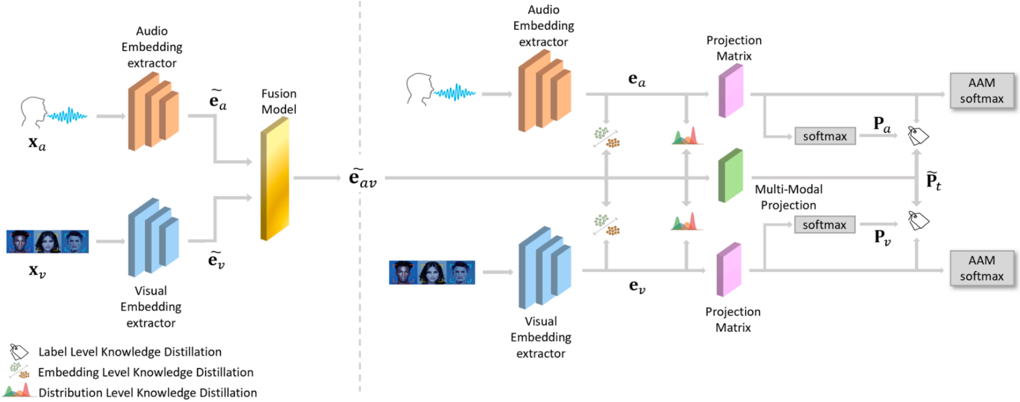


Figure: Knowledge distillation from audio-visual to single-modality systems[6]

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

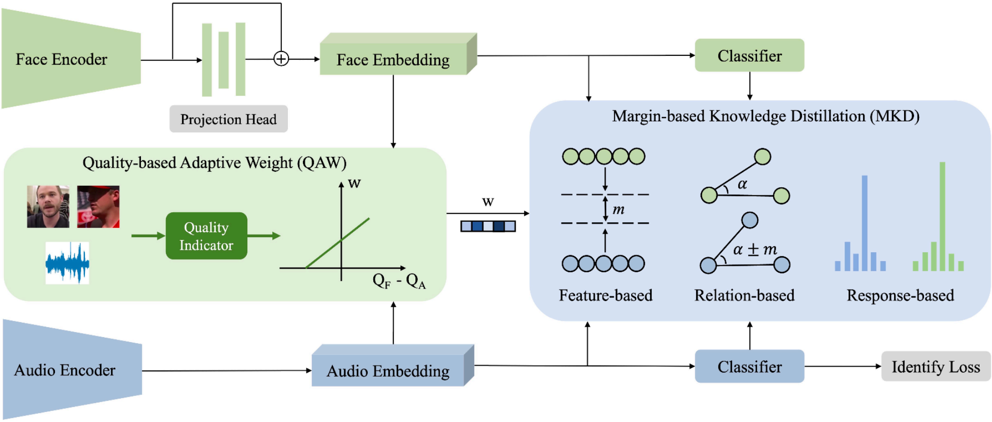


Figure: Knowledge distillation from visual to audio systems[7]

Model Efficiency

We mainly optimize model efficiency from two perspectives in speaker representation learning: computation and memory.

- Computation efficiency
  - Knowledge distillation
  - Quantization
  - Efficient architecture design
- Memory efficiency
  - Reversible neural networks

Knowledge distillation for **SV**

- Teacher-to-student distillation[a]
- Self-distillation with feature augmentation[b]
- Distillation from multimodal to audio-only[c]

[a]Wang et al., "Knowledge Distillation for Small Foot-print Deep Speaker Embedding".

[b]Liu et al., "Self-Knowledge Distillation via Feature Enhancement for Speaker Verification".

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

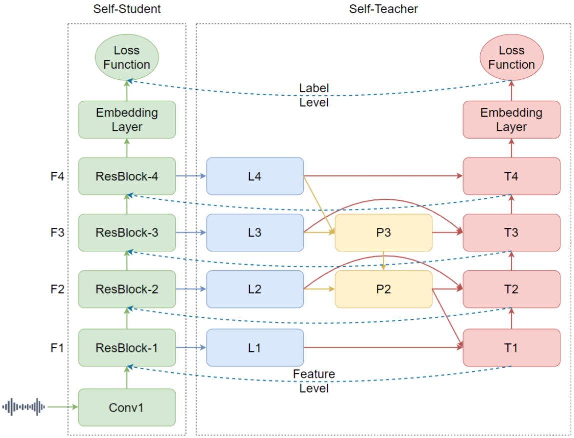
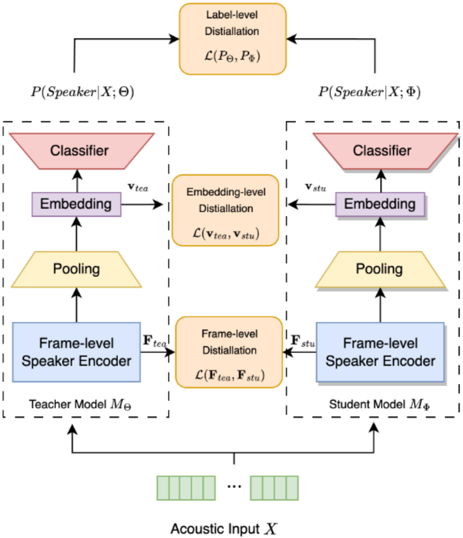


Figure: Self-distillation with feature augmentation

Knowledge distillation at different levels

- Feature-level distillation
- Embedding-level distillation
- Label-level distillation



Quantization compresses models by lowering parameter precision

- Binary neural networks[a]
- Linear and PoT (power-of-two) quantization[b]
- K-Means based quantization[c]
- Static/adaptive quantizers for binary quantization[d]

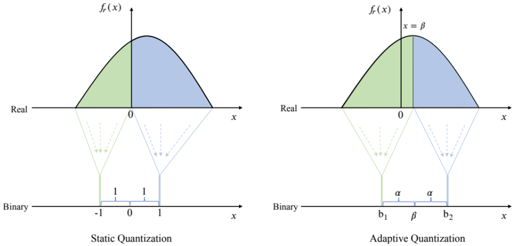


Figure: Static and adaptive binary quantization overview

[a] Zhu, Qin, and Li, "Binary Neural Network for Speaker Verification".

[b] Liu et al., "Self-Knowledge Distillation via Feature Enhancement for Speaker Verification".

[c] Wang et al., "Adaptive Neural Network Quantization For Lightweight Speaker Verification".

[d] Liu, Wang, and Qian, "Extremely Low Bit Quantization for Mobile Speaker Verification Systems Under 1MB Memory".

Efficient architectures for **SV**

- Depth-first networks with attentive feature fusion[a]
- CS-CTCSCONV1D[b] (channel-split time-channel-time separable 1D conv)
- Asymmetric enroll-test design (ECAPA-TDNNLite[c])

[a] Liu, Chen, and Qian, "Depth-First Neural Architecture With Attentive Feature Fusion for Efficient Speaker Verification".

[b] Cai et al., "CS-CTCSCONV1D: Small footprint speaker verification with channel split time-channel-time separable 1-dimensional convolution".

[c] Li et al., "Towards Lightweight Applications: Asymmetric Enroll-Verify Structure for Speaker Verification".

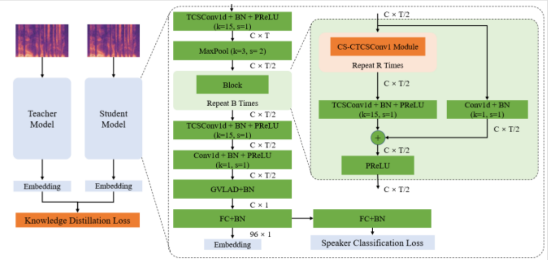
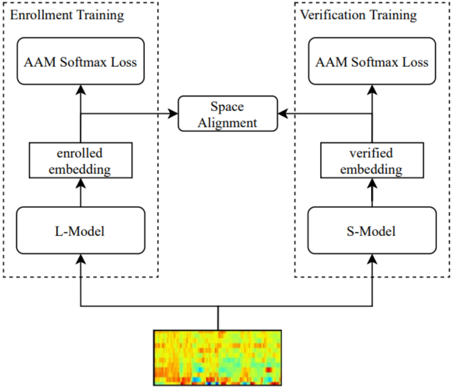


Figure: CS-CTCSCONV1D illustration

Computation Efficiency

Asymmetric Enroll-Test Design



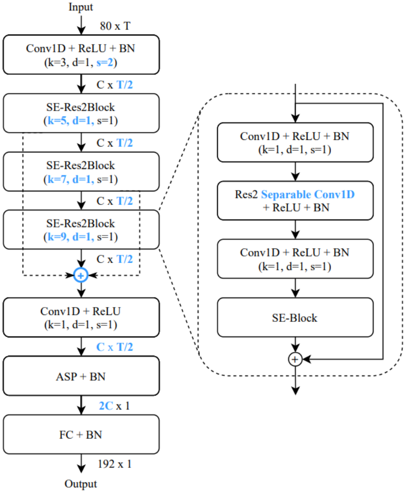


Figure: Training with asymmetric structure.

Frame-level features are fed to both large and small

Table: Results of compressed/quantized ResNet34 and other compact full-precision models.

Model	Size (MB)	Bitwidth (bit)	Vox1-O EER(%)
KMQAT-ResNet34[8]	3.45	4	**0.957**
PoT-ResNet34[9]	3.45	4	1.09
TWN-ResNet34[10](our impl.)	1.80	2	1.473
b-vector(adaptive)[11]	0.97	**1**	**1.72**
ResNet34(binary)[12]	**0.66**	**1**	5.355
CS-CTCSConv1d	0.96	32	2.62
ECAPA-TDNNLite	1.2	32	3.07

[8] Wang et al., "Adaptive Neural Network Quantization For Lightweight Speaker Verification".

[9] Li et al., "Model Compression for DNN-based Speaker Verification Using Weight Quantization".

[10] Li, Zhang, and Liu, "Ternary weight networks".

[11] Liu, Wang, and Qian, "Extremely Low Bit Quantization for Mobile Speaker Verification Systems Under 1MB Memory".

[12] Zhu, Qin, and Li, "Binary Neural Network for Speaker Verification".

Reversible neural networks[a] (RevNets) alleviate the need to store activations in memory during backprop. Thus, memory usage is nearly constant with depth.

- Partially reversible networks
- Fully reversible networks

[a]Liu and Qian, "Reversible Neural Networks for Memory-Efficient Speaker Verification".

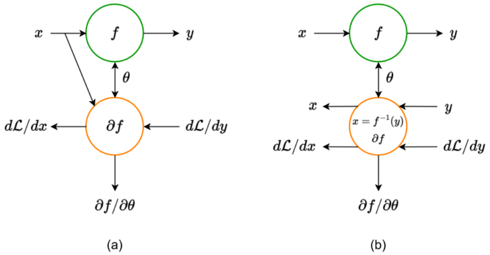
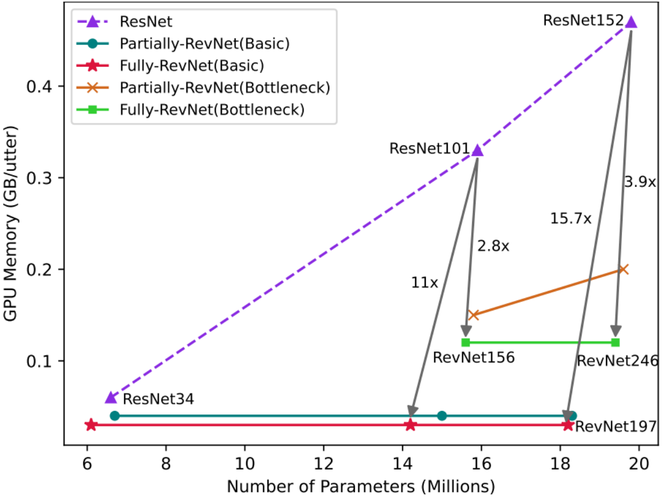


Figure: Non-invertible (a) vs. invertible (b) operators

Memory Efficiency

GPU Memory vs. Param Count

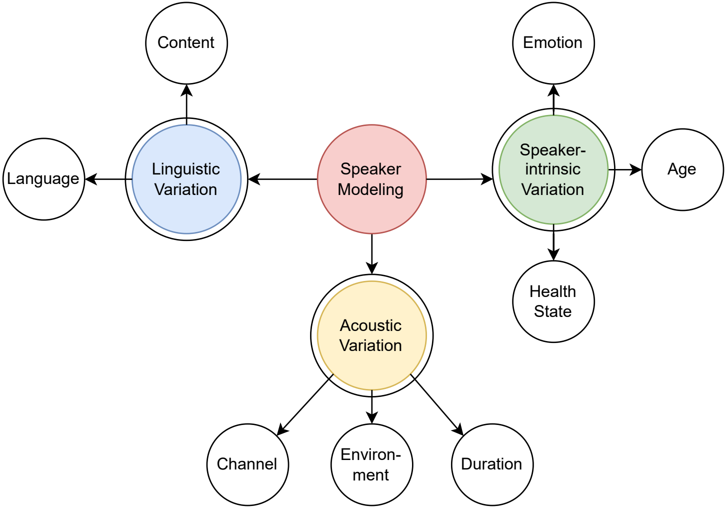


Other works on model efficiency

- Thin-ResNet[13]
- Fast-ResNet[14]
- ADMM[15]
- Small Footprint Text-Independent Speaker Verification[16]

[13] Cai, Chen, and Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system".
[14] Chung et al., "In Defence of Metric Learning for Speaker Recognition".
[15] Xu et al., "Mixed Precision Low-Bit Quantization of Neural Network Language Models for Speech Recognition".
[16] Balian et al., "Small footprint text-independent speaker verification for embedded systems".

Robustness in Speaker Representation Learning



Model Robustness

Recording environments introduce variability to speaker identity modeling due to device and microphone distance, etc. Various domain adaptation methods are adopted in speaker recognition to enhance robustness across devices, including

- Discrepancy-based alignment
- Adversarial learning
- Domain-specific adapters

Model Robustness to Devices

Discrepancy-based alignment aims to minimize domain differences in the latent space and promote domain-invariant representations. Proper divergence measures are central to these methods, e.g., MMD[17], CORAL[18], etc.

$$\mathcal{L}_{\mathrm{mmd}} \triangleq \sup_{\phi \in \Phi} \left(\mathrm{E}_S \left[\phi \left(S \right) \right] - \mathrm{E}_T \left[\phi \left(T \right) \right] \right) \tag{2}$$

[17] Li, Han, and Song, "CDMA: Cross-Domain Distance Metric Adaptation for Speaker Verification".

[18] Li, Zhang, and Chen, "The coral++ algorithm for unsupervised domain adaptation of speaker recognition".

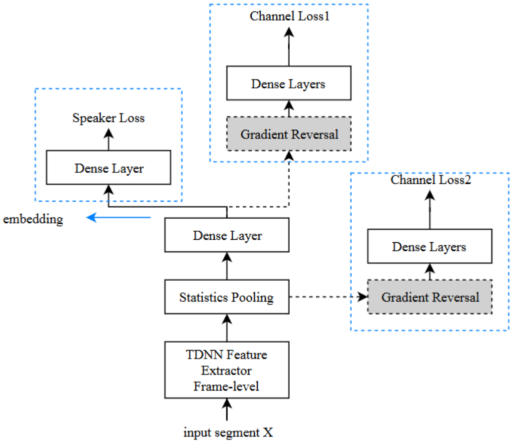


Figure: Channel-level adversarial learning[a]

Adversarial learning uses a domain classifier to remove discriminative domain information from features. The min-max optimization in domain-adversarial training reduces domain gaps and enforces domain-invariant feature extraction[a].

[a]Chen et al., "Channel invariant speaker embedding learning with joint multi-task and adversarial training".

Instead of directly aligning domains with discrepancy measures, adding domain-specific adapters and related modules helps capture and mitigate domain variance, yielding domain-invariant embeddings.

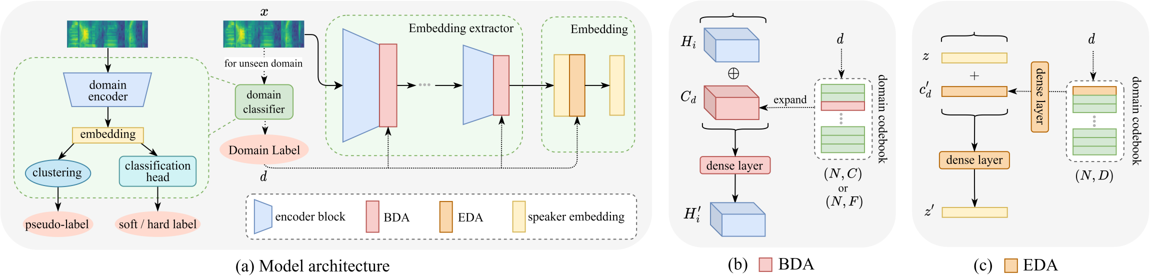


Figure: Framework with domain-specific adapters[19]

[19] Huang et al., "Enhancing Cross-Domain Speaker Verification through Multi-Level Domain Adapters".

Model Robustness to Language Mismatch

Observation: In real scenarios, SV systems trained in one language may degrade when tested in another language.

Model Robustness to Language Mismatch

Over 40% of the world's population is bilingual. Mismatch happens when enrollment and test use different languages.

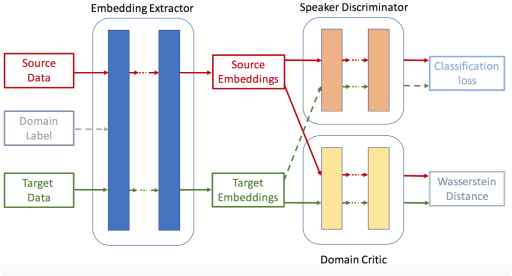


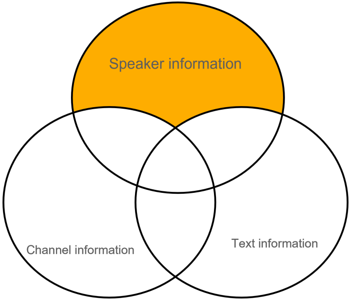
Figure: Adversarial learning for language mismatch

Adversarial learning uses a language classifier to remove discriminative language information from features. The min-max optimization minimizes language gaps and enforces language-invariant feature extraction[a][b].

[a] Rohdin et al., "Speaker verification using end-to-end adversarial language adaptation".

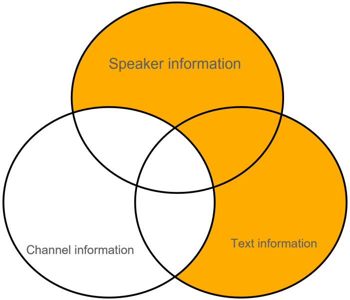
[b] Xia, Huang, and Hansen, "Cross-lingual text-independent speaker verification using unsupervised adversarial discriminative domain adaptation".

In addition to speaker identity, text/content is key to speech communication.



For text-independent tasks, we only need speaker information.

Enrollment: Hey Siri; Test: arbitrary phrase

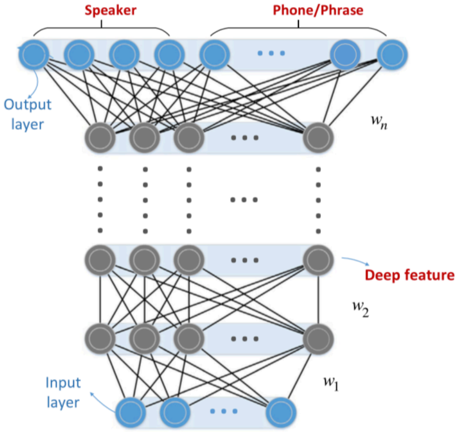


For text-dependent tasks, we also need content information.

Enrollment: Hey Siri; Test: Hey Siri

Content Representations

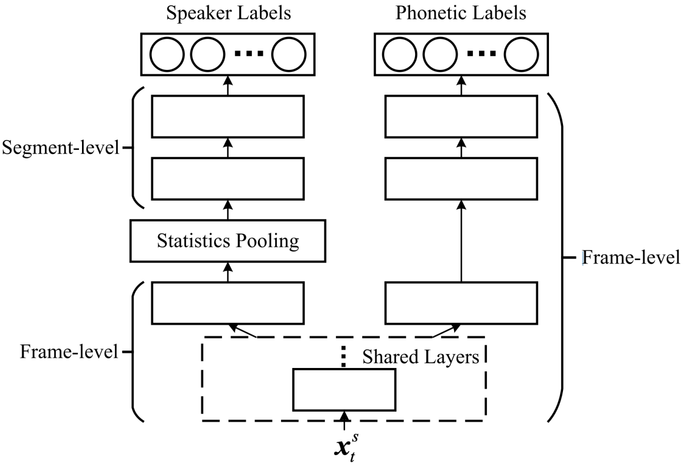
- Phone indices
- ASR-predicted phone posteriors
- Hidden representations of ASR models
- Phrase IDs (fixed-phrase datasets)
- Normalized phone distribution



- **Text-dependent** task
- Multi-task at **frame-level**
- Performance improvement

Explicitly modeling phonetic information is intuitive for text-dependent SV.

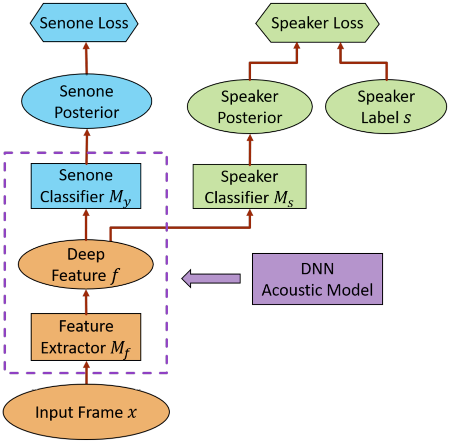
[20] Liu, Yuan, et al. "Deep feature for text-dependent speaker verification." Speech Communication 73 (2015): 1-13.



- **Text-independent** task
- Multi-task at **frame-level**
- Performance improvement

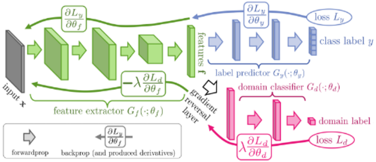
[21] Liu, Yi, et al. "Speaker Embedding Extraction with Phonetic Information." Proc. Interspeech 2018 (2018): 2247-2251.

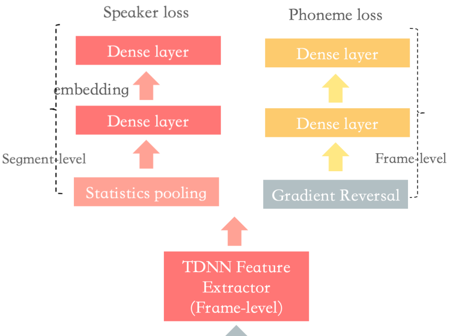
Speaker-invariant Training for ASR[22]



- Acoustic modeling
- Adversarial training to suppress speaker effect
- Performance improvement

[22] Meng, Zhong, et al. "Speaker-invariant training via adversarial learning." ICASSP 2018.



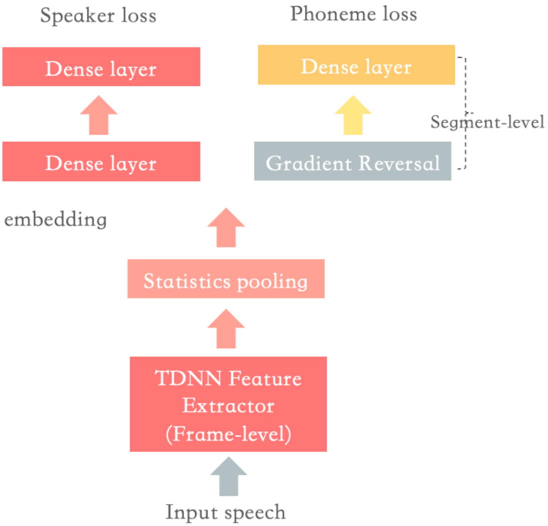


$$\mathcal{L}_s = \mathrm{CE}(M_s(M_f(\mathbf{X})), \mathbf{y}^s)$$

$$\mathcal{L}_p = \frac{1}{N}\sum_{i=1}^{N} \mathrm{CE}(M_p(M_f(\mathbf{x}_i)), \mathbf{y}_i^p)$$

$$\mathcal{L}_{total} = \mathcal{L}_s + \mathcal{L}_p$$

Systems	voxceleb1_O	voxceleb1_E	voxceleb1_H
x-vector baseline	2.361	2.470	4.260
FRM-MT	2.165	2.198	3.911
FRM-ADV	3.143	3.214	5.419

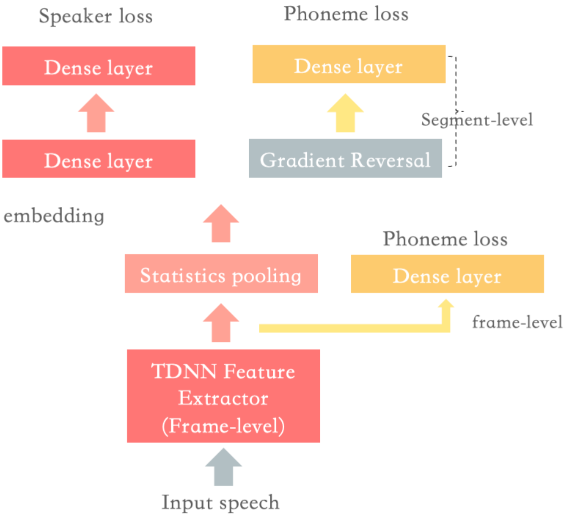


$$\mathcal{L}_s = \text{CE}(M_s(M_f(\mathbf{X})), \mathbf{y}^s)$$
$$\mathcal{L}_p = \text{CE}(M_p(M_f(\mathbf{x}_i)), \mathbf{y}^p)$$
$$\mathcal{L}_{total} = \mathcal{L}_s + \mathcal{L}_p$$

For a segment $\mathbf{x}$ with N frames, the segment-level phonetic label $\mathbf{y}^p$ is

$$\mathbf{y}^p = \{y_1, y_2, \dots, y_C\}$$
$$y_c = \frac{N_c}{N}$$

where C is the size of the selected phone set and N_c is the number of occurrences of phone c in $\mathbf{x}$.



Systems	voxceleb1_O	voxceleb1_E	voxceleb1_H
x-vector baseline	2.361	2.470	4.260
SEG-MT	2.175	2.330	4.059
SEG-ADV	2.154	2.198	3.923

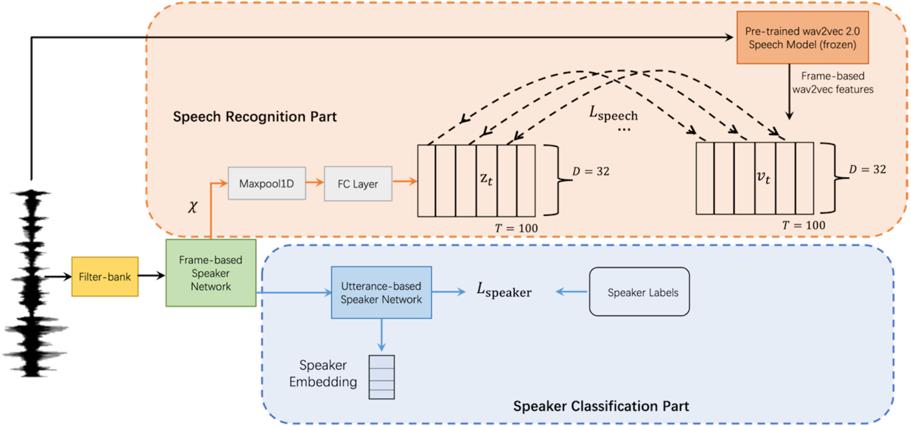
Figure: Segment-level multi-task/adversarial

Systems	voxceleb1_O	voxceleb1_E	voxceleb1_H
x-vector baseline	2.361	2.470	4.260
FRM-MT	2.165	2.198	3.911
SEG-ADV	2.154	2.198	3.923
COMBINE	2.013	2.030	3.819

Figure: Frame-level multi-task + segment-level adversarial

Text Robustness

Multi-task Training with Advanced Content Representations[23]



[23] Jin, Tu, and Mak, "Phonetic-aware speaker embedding for far-field speaker verification".

Extract PBN from pretrained ASR and combine with filterbanks[24]

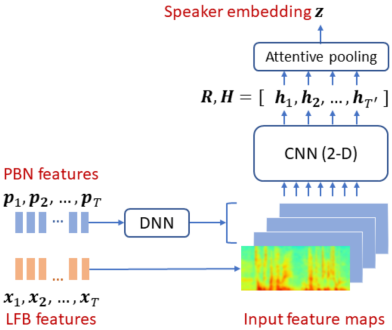


Fig. 1: *Implicit phonetic attention by combining LFB and PBN features at the input layer (LFB: log filter bank; PBN: phonetic bottleneck).*

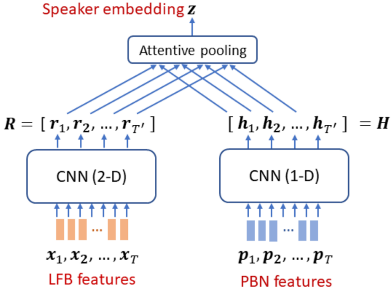
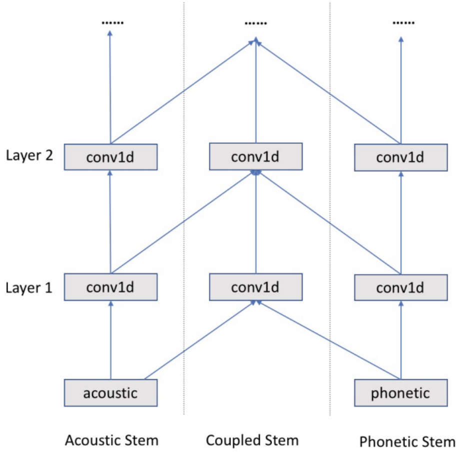


Fig. 2: *Explicit phonetic attention by routing LFB and PBN features through separate networks (LFB: log filter bank; PBN: phonetic bottleneck).*

[24] Zhou T, Zhao Y, Li J, et al. CNN with phonetic attention for text-independent speaker verification, ASRU 2019



25

Table 1: *Network configurations of PacNet*

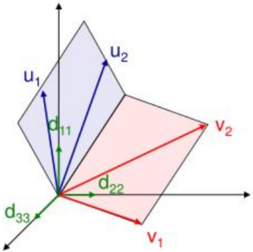
Layer 7	Linear	In=1024 Out=1000		
Layer 6	Pooling	In=1024 Out=1024		
Layer 5	Conv1d	In=2048 Out=1024		
Layer 4	Conv1d kernel=5	Out=512 In=512	Out=1024 In=2048	Out=512 In=512
Layer 3	Conv1d kernel=5	Out=512 In=512	Out=1024 In=2048	Out=512 In=512
Layer 2	Conv1d kernel=5	Out=512 In=512	Out=1024 In=2048	Out=512 In=512
Layer 1	Conv1d kernel=5	Out=512 In=40	Out=1024 In=140	Out=512 In=100
Stem		Acoustic	Coupled	Phonetic

- Use Triplet loss instead of softmax

[25] Zheng, Lei, and Suo, "Phonetically-Aware Coupled Network For Short Duration Text-Independent Speaker Verification."

Joint factor analysis for speaker representations[26]

$$\mathbf{M} = \mathbf{M}^{\text{UBM}} + \mathbf{V}\mathbf{y} + \mathbf{D}\mathbf{z} + \mathbf{U}\mathbf{x}$$

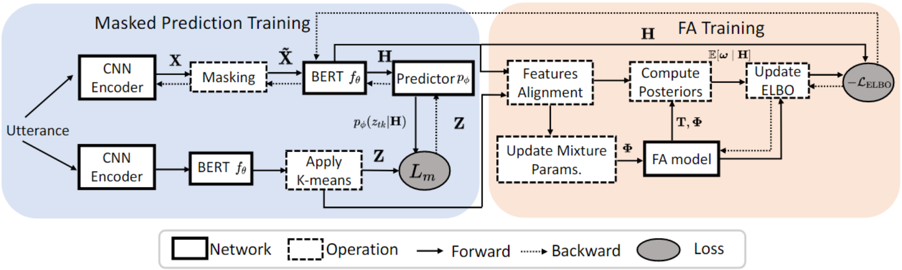


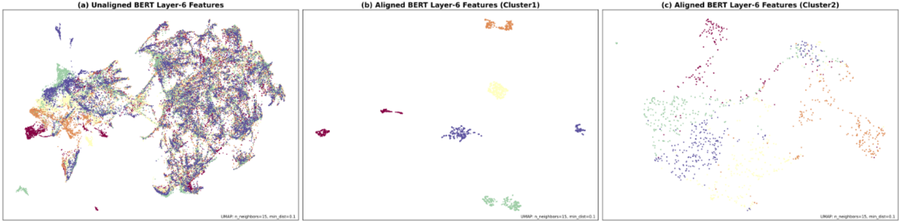
- Gaussian priors over factors $\mathbf{y}$, $\mathbf{z}$, $\mathbf{x}$
- Estimate $\mathbf{M}^{\text{UBM}}$, $\mathbf{V}$, $\mathbf{D}$, $\mathbf{U}$ with EM
- $\mathbf{V}$ captures primary speaker variability (speaker factors)
- $\mathbf{D}$ captures channel variability
- $\mathbf{U}$ captures residual variability

[26] Kenny, Patrick, et al. "Joint factor analysis versus eigenchannels in speaker recognition." TASLP 2007

Disentangling Speech Representations

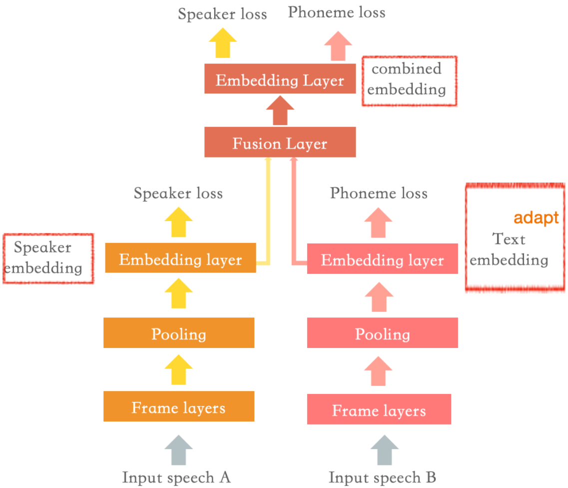
Neural Factor Analysis: ignore phonetic variability via extra alignment [27]





Disentangling Speech Representations

Factorizing and Re-composing Phonetic Information[28]



- Segment-level reconstruction
- Factorize speaker and text information
- For text-independent tasks, ignore text information
- For text-dependent tasks, use combined embeddings
- Text adaptation: modify text-related info in embeddings while preserving speaker identity (change enrolled keyword)

[28] Yang Y*, Wang S*, Gong X, et al. Text adaptation for speaker verification with speaker-text factorized embeddings. ICASSP 2020

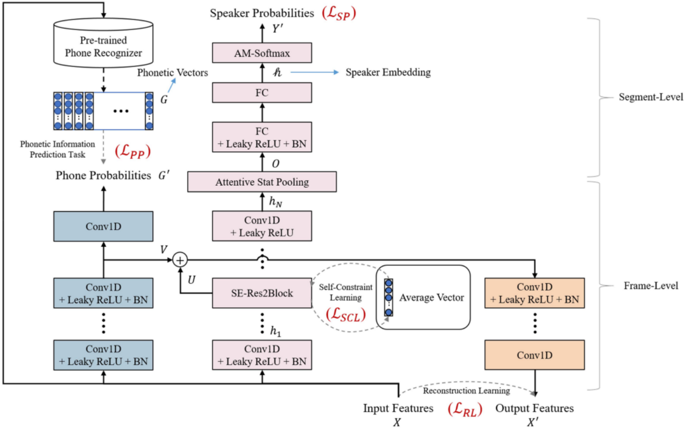
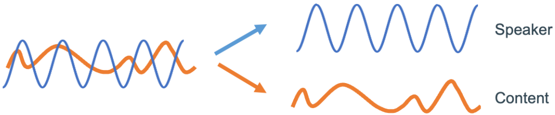


Fig. 3. The architecture of the proposed DROP-TDNN x-vector system. DROP-TDNN consists of three training procedures, including phonetic information prediction, reconstruction and speaker recognition.

- Frame-level reconstruction
- Mean-center frame-level speaker reps
- Coarse phonetic classes (vowel, semivowel, affricate, ...)

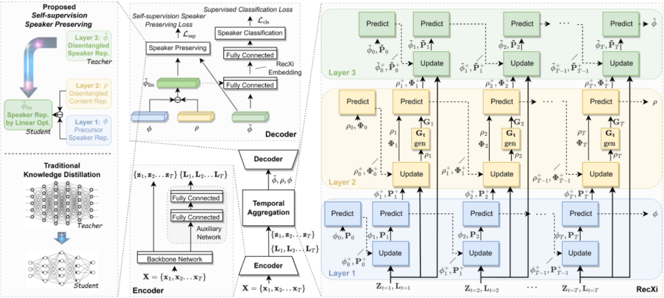
[29] Hong Q B, Wu C H, Wang H M. Decomposition and Reorganization of Phonetic Information for Speaker Embedding Learning. TASLP 2023

Disentangle **Static** and **Dynamic** components in Speech



by three **Gaussian inference layers**

and a novel **speaker preserving self-supervision**



[30] Liu T C, Disentangling Voice and Content with Self-Supervision for Speaker Recognition

Ensure the first-layer representation contains content-related information; residual layers naturally fill the remaining details—specifically, paralinguistic information.[31]

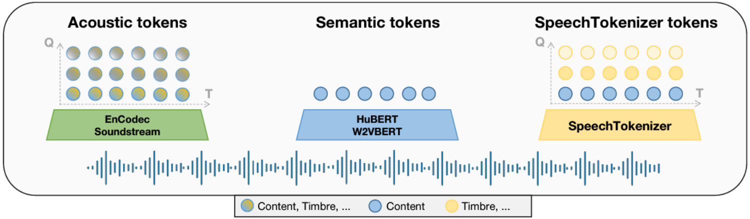


Figure 1: Illustration of information composition of different discrete speech representations. Speech tokens are represented as colored circles and different colors represent different information.

[31] Zhang X, Zhang D, Li S, et al. SpeechTokenizer: Unified Speech Tokenizer for Speech Large Language Models, 2023.

Semantic distillation for disentanglement

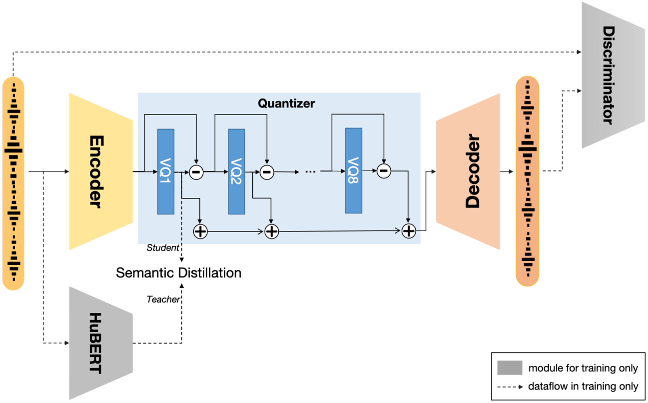


Figure 2: Illustration of SpeechTokenizer framework.

- **Continuous distillation** HuBERT layer-9 outputs / layer-mean
  $\mathcal{L}_{\text{distll}} = \frac{1}{T}\sum_{t=1}^{T} \log \sigma\left(\cos\left(Aq_1^t, s^t\right)\right)$
- **Discrete distillation** pseudo-label prediction
  $\mathcal{L}_{\text{distll}} = -\frac{1}{T}\sum_{t=1}^{T} u^t \log\left(\text{Softmax}\left(Aq_1^t\right)\right)$

Assume HuBERT is a perfect semantic encoder

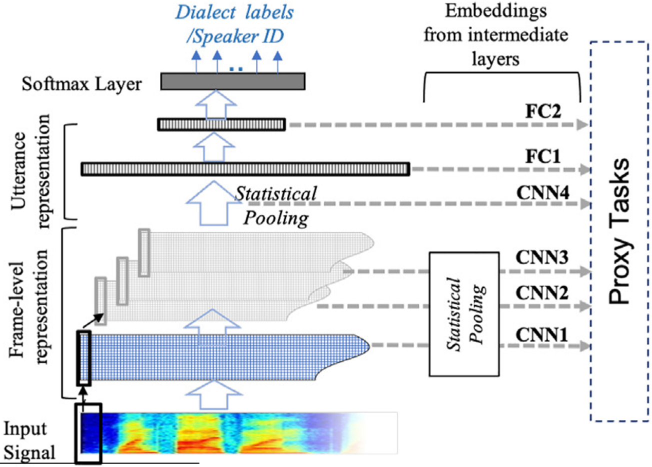
Outline

Assumption: If an attribute is encoded in speaker representations, a classifier predicting that attribute will achieve accuracy depending on how well it is embedded.[32],[33],[34],[35]

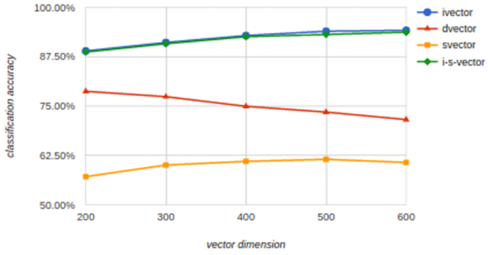
- Speaker-related attributes: identity, gender, speaking rate.
- Text-related factors: spoken words, word order, utterance length.
- Channel-related elements: phone ID, noise type.

[32] Wang, Qian, and Yu, "What does the speaker embedding encode?"
[33] Belinkov and Glass, "Analyzing hidden representations in end-to-end automatic speech recognition systems".
[34] Raj et al., "Probing the information encoded in x-vectors".
[35] Zhao et al., "Probing Deep Speaker Embeddings for Speaker-related Tasks".

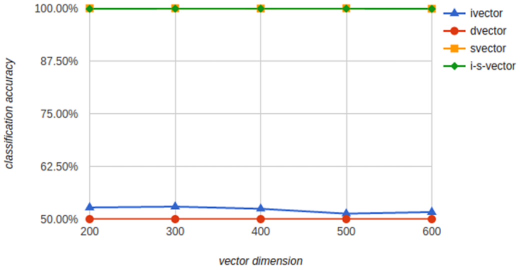
Paradigm: probe pretrained embeddings with proxy tasks[36]



Examples of speaker identity and word order tasks[37]



Speaker identity task



Word order task

[37] Wang, Qian, and Yu, "What does the speaker embedding encode?"

In speaker modeling, f is the speaker classifier, c denotes the class, and θ denotes trainable parameters.

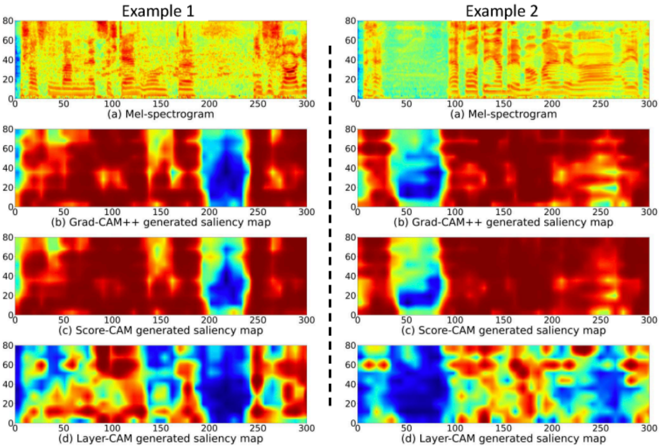
$$y^c = f_c(x;\theta)$$

For the k-th activation map A^k (e.g., k indexes channels), define each entry w_{ij}^{kc} as

$$w_{ij}^{kc} = \mathrm{ReLU}\left(\frac{\partial y^c}{\partial A_{ij}^k}\right)$$

The saliency map is the linear combination

$$S_{ij}^c = \mathrm{ReLU}\left(\sum_k w_{ij}^{kc} \cdot A_{ij}^k\right)$$

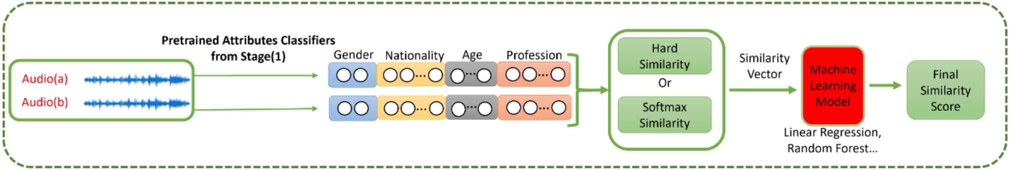


[38] Li et al., "Reliable visualization for deep speaker recognition".

[39] Li et al., "Visualizing data augmentation in deep speaker recognition".

Explainable attribute-based SV:

Use voice attributes (gender, age, nationality, profession) for more transparent SV process.

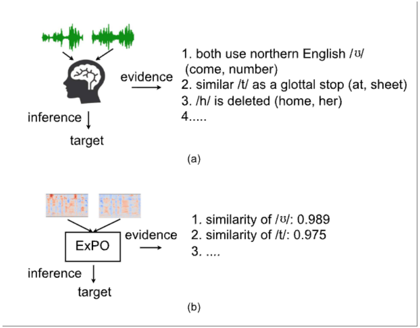


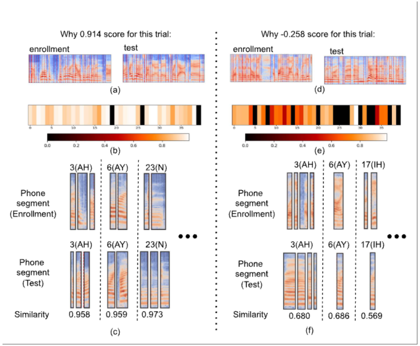
[40] Wu et al., "Explainable attribute-based speaker verification".

ExPO[a]

[a]Ma et al., "ExPO: Explainable Phonetic Trait-Oriented Network for Speaker Verification".

ExPO not only generates utterance-level speaker embed- dings but also allows for fine-grained analysis and visualization of phonetic traits

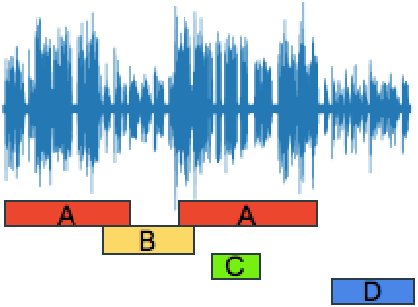




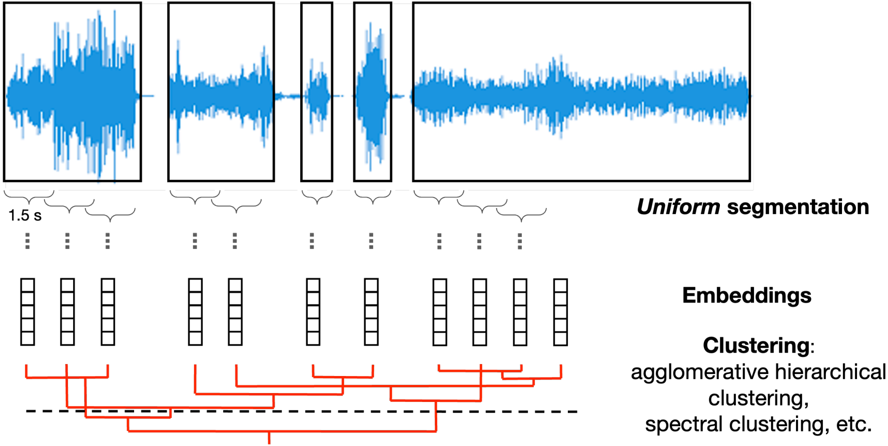
Outline

1. Pretrained speaker embeddings as extra input
2. Joint training to learn task-specific embeddings
3. Implicit speaker modeling

- 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



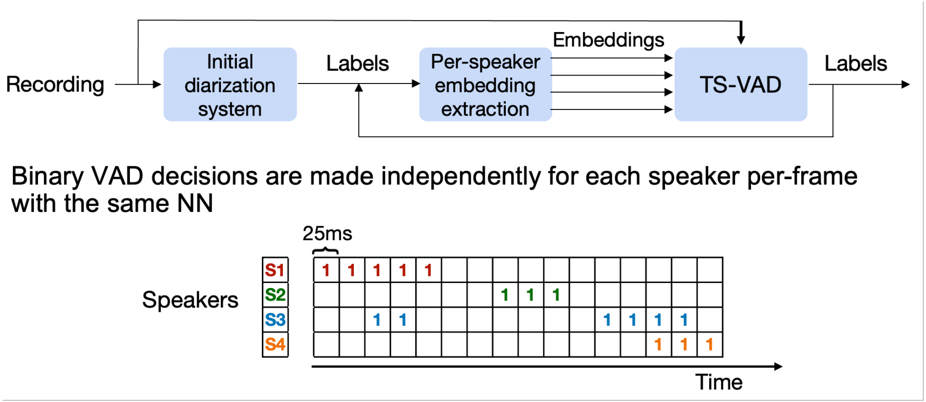
南京大學 NANJING UNIVERSITY
智能科学与技术学院 School of Intelligence Science and Technology



[41] Thanks to Mireia Diez Sánchez for the figure

Speaker Modeling Across Tasks

TS-VAD based Speaker Diarization[42][43]

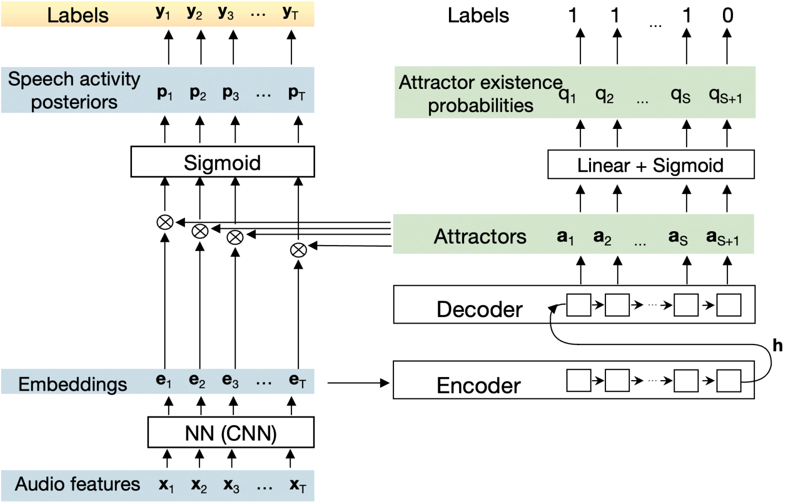


[42] Medennikov et al., "Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario".

[43] Thanks to Mireia Diez Sánchez for the figure

Speaker Modeling Across Tasks

EEND-EDA based Speaker Diarization[44],[45][46]



[44] Fujita et al., "End-to-end neural speaker diarization with self-attention".

- 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

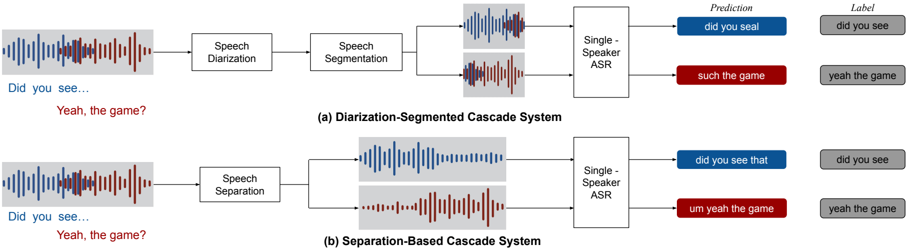
Challenges

- **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

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

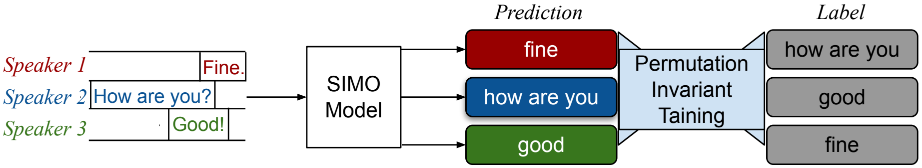
Speaker Modeling across Tasks

Multi-speaker ASR: Cascaded Systems[48]

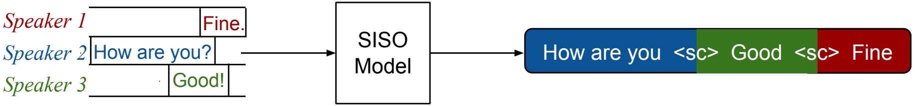


- Diarization-Segmented: first ̈speaker segmentation ̈, then single-speaker ASR
- Separation-based: first ̈speech separation ̈, then single-speaker ASR

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



(a) Single-Input Multiple-Output Framework

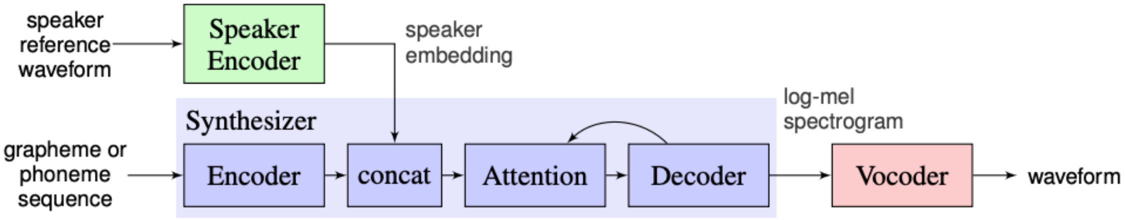


(b) Single-Input Single-Output Framework

- 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 Modeling across Tasks

Example: Explicit Speaker Modeling for Zero-shot TTS[49,50,51]



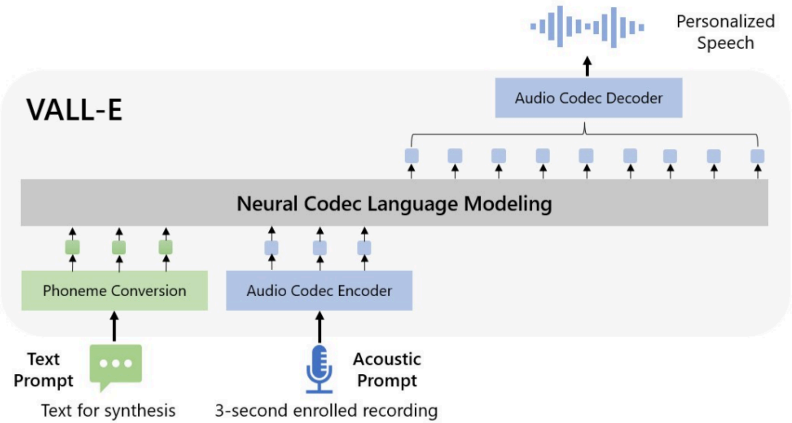
[49] Jia et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis".

[50] Casanova et al., "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone".

[51] Wu et al., "Adaspeech 4: Adaptive text to speech in zero-shot scenarios".

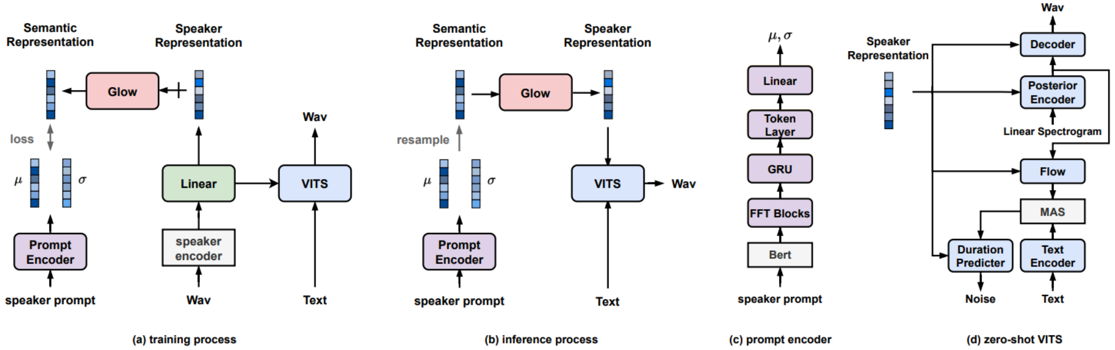
Speaker Modeling across Tasks

Example: Implicit Speaker Modeling for Zero-shot TTS[52,53,54]



[52] Wang et al., "Neural codec language models are zero-shot text to speech synthesizers".
[53] Du et al., "UniCATS: A Unified Context-Aware Text-to-Speech Framework with Contextual VQ-Diffusion and Vocoding".
[54] Le et al., "Voicebox: Text-guided multilingual universal speech generation at scale".

Example: Towards Controllability and Novel Voice Generation[55,56,57,58]



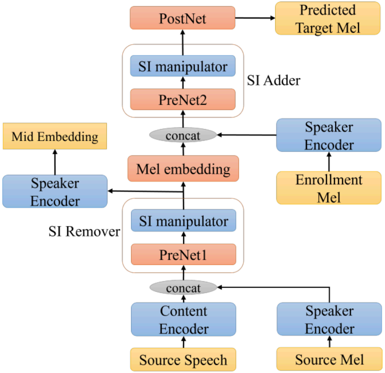
---

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

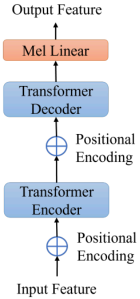
[56] Stanton et al., "Speaker generation".

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

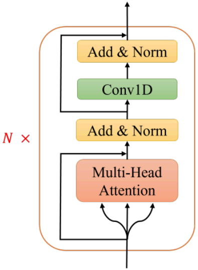
[58] Bilinski et al., "Creating new voices using normalizing flows".



(a) Proposed System



(b) SI Manipulator



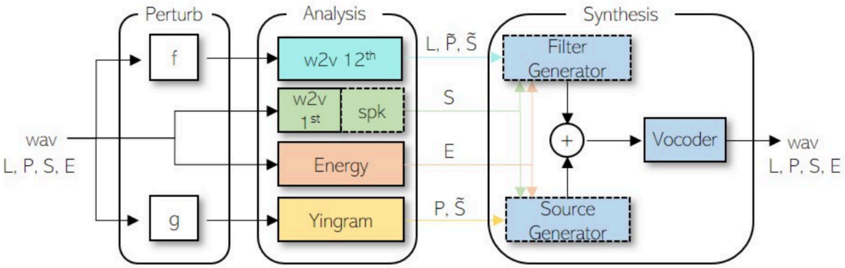
(c) Encoder and decoder architecture

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

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

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

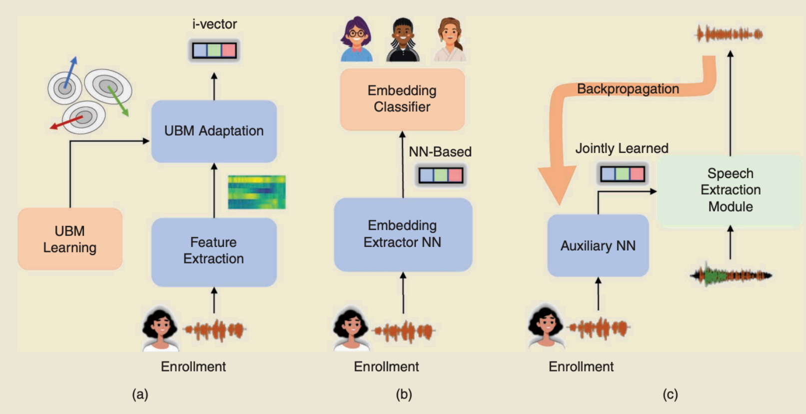
Example: Implicit Speaker Modeling for Zero-shot VC[62]



[62] Choi et al., 2021; Wu et al., 2020; Wu et al., 2020

Speaker Modeling across Tasks

Example: Explicit Speaker Modeling for Target Speaker Extraction[63][64][65]



[63] Zmolikova et al., "Neural Target Speech Extraction: An overview".

[64] Delcroix et al., "Single channel target speaker extraction and recognition with speaker beam".

[65] Ge et al., "Spex+: A complete time domain speaker extraction network".

Speaker Modeling across Tasks

Example: Implicit Speaker Modeling for Target Speaker Extraction[66]

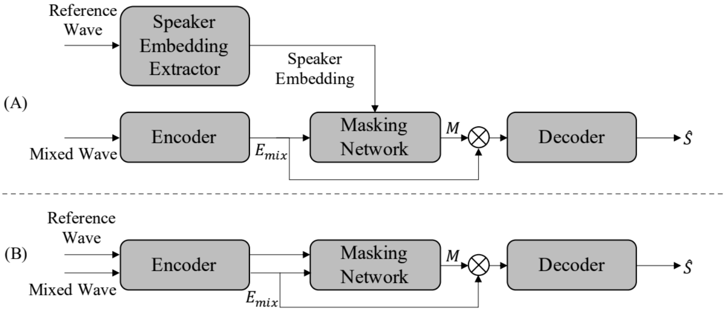


Figure 1: *(A) is the diagram of a typical time-domain target speaker extraction method. (B) is the diagram of our proposed method. $\otimes$ is an operation for element-wise product.*

[66] e.g., Zeng; Yang 2023

Common Assumption

Speaker embeddings optimized for recognition work for all tasks

Status quo:

- x-vector, ECAPA-TDNN, ResNet widely used
- Assumed universal applicability
- Directly transferred to other tasks

Problem

SV $\neq$ TTS $\neq$ VC $\neq$ Target-Speaker Processing

Key Question

What constitutes an "ideal" speaker representation?

Speaker Recognition

- Maximize discriminability
- Minimize within-speaker variance
- Robust to noise/channel
- Compact representations

Generative Tasks

- Capture fine-grained details
- Preserve prosody/emotion
- Natural synthesis
- Rich, expressive representations

Target-Speaker Processing

- Relative discriminability
- Maximize correlation with target in mixtures
- Compact representations

Conflicts

Discriminative optimization vs. generative richness
Absolute discriminability vs. relative discriminability

- SV/verification: suppress within-speaker variability
- TTS/VC: preserve and leverage such variability
- TSE, target-speaker ASR, speaker separation: relative discriminability within small sets

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

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

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

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?

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

Convenience Trap

Easy to use, but often suboptimal

Reasons:

- Availability of pretrained models
- Early success in SV
- Convenience

Costs:

- Suboptimal performance
- Information bottleneck
- Limited innovation
- Task mismatch

Evidence

USEF-TSE outperforms embedding-based methods

YourTTS custom encoders > SV embeddings

"Averaging" Problem

SV embeddings compress diverse acoustic expressions into one point

Lost Content:

- Emotional variation
- Prosody patterns
- Speaking rate changes
- Style nuances
- Contextual features

Impact on Applications:

- Monotonic TTS output
- Limited VC expressiveness
- Poor emotion control
- Unnatural prosody

Key Question

How to retain intra-speaker variability while preserving discriminability?

Intrinsic Complexity

Speech factors are inherently entangled rather than independently encoded

Entanglements:

- F0 contour: emotion + linguistic structure
- Spectral features: timbre + content
- Prosody: speaker + emotion + content
- No clear boundaries

Current Solutions:

- Adversarial learning
- Mutual information minimization
- Multi-encoder architectures
- Specialized losses

Reality

Perfect disentanglement remains open

Audio LLMs (ALLMs):

- Inspired by text LLMs (GPT, Qwen)
- Strong on diverse audio tasks:
  - ASR
  - Audio captioning
  - Music QA
- Excellent generalization
- Can identify speaker attributes (gender, age, accent)

Key Question

Can ALLMs perform speaker verification?

Observation

ALLM-based systems are still largely insensitive to speaker identity in dialogue.

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

Key idea: reformulate speaker verification as audio QA.

Four prompting strategies:

1. **Separate**: feed two utterances independently
   - Prompt: "Audio1: [audio1], Audio2: [audio2]. Same speaker?"
2. **Concat**: concatenate two utterances
   - Prompt: "How many speakers are in this audio?"
3. **Concat+Silence**: concatenate with 1s silence
   - Prompt: same as Concat
   - **Hypothesis: silence helps distinguish speakers**
4. **Mix**: overlap two utterances
   - Prompt: "This audio mixes two tracks. Same speaker?"

Model	Gender	Lang	Age	Device	Dur$<$2s	Dur$>$6s
Kimi (C+S)	70.20	68.40	63.40	52.67	53.70	**73.60**
Qwen2 (C+S)	59.40	58.60	53.87	52.20	50.60	59.10
Step (C+S)	64.20	60.40	56.80	57.47	54.60	71.80

Observations:

- 70% accuracy on long utterances
- Significant drop under challenging conditions:
  - Cross-device: 52–57%
  - Short duration: 50–55%
- **Choose Kimi-Audio with Concat + Silence for fine-tuning**

Conclusion

Zero-shot ALLMs have limited SV ability $\rightarrow$ motivates fine-tuning

Model	Gender	Lang	Age	Device	Dialog	Dur<2s	Dur>6s
Kimi (zero-shot)	70.20	68.40	63.40	52.67	55.00	53.70	73.60
Kimi (fine-tuned)	**95.07**	**97.00**	**92.40**	**88.20**	**89.00**	**80.90**	**89.50**
Kimi (random sampling)	94.80	93.07	92.27	85.60	80.53	77.00	89.00
ECAPA-TDNN	99.33	99.27	94.13	94.67	93.00	78.80	95.60

Findings:

1. **Huge improvements:** e.g., gender 70% → 95%
2. **Hard negative sampling matters:** consistently better than random sampling
3. **ALLM surpasses ECAPA-TDNN on short duration!** (80.90% vs 78.80%)
4. **Still a gap on easy conditions** (e.g., gender: 95% vs 99%)

Insight

ALLMs show **stronger robustness** in challenging scenarios, indicating potential in noisy real-world settings.

Joint verification formulation:

- Enrollment: [audio1], Test: [audio2], Target text: “Hello world”
- Question: “Same speaker as enrollment? Does test match the target text?”
- Answer: “Speaker: yes/no, Content: yes/no”

Evaluation on LibriSpeech:

Model	Spk Acc (%)	Text Acc (%)	Overall (%)
Kimi (zero-shot)	62.09	89.61	52.31
Kimi (fine-tuned)	98.92	**99.95**	**98.87**
Whisper + ECAPA	**99.08**	99.75	98.83

Toolkit	Speaker-specific	SSL	Pre-trained Models	Deployment
Kaldi	No	No	No	No
VoxCeleb_Trainer	Yes	No	No	No
ASV-Subtools	Yes	No	No	Yes
SpeechBrain	No	No	No	No
NeMo	No	No	No	Yes
Espnet	No	No	Yes	No
3D-Speaker	Yes	Yes	No	No
Wespeaker	Yes	Yes	Yes	Yes

Table: Common toolkits for speaker modeling

Popular Datasets

Dataset	Year	Speakers	Utterances	Duration
VoxCeleb1	2017	1,251	153,516	351h
VoxCeleb2	2018	6,112	1,128,246	2,442h
CN-Celeb1	2020	1,000	130,109	274h
CN-Celeb2	2020	2,000	529,485	1,090h
3D-Speaker	2023	10,000	579,013	1,124h
VoxBlink	2023	38,065	1,455,190	2,135h

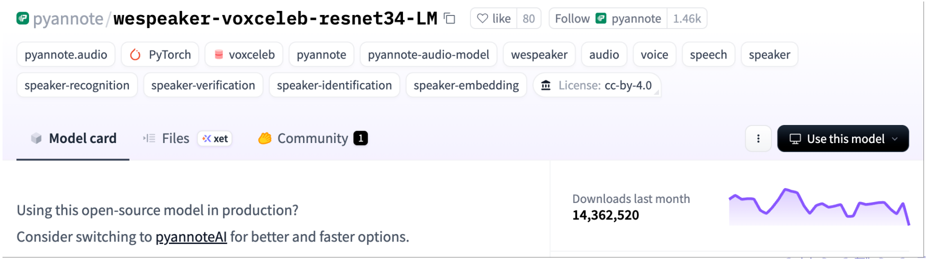
Table: Representative Speaker Recognition Datasets

Performance Trends:

- VoxCeleb is approaching saturation
- Need more challenging scenarios
- Cross-genre and far-field datasets
- Large-scale unlabeled datasets for SSL

Wespeaker is a speaker embedding toolkit for both **research** and **production**, featuring

- Lightweight codebase
- SOTA performance
- Discriminative and SSL paradigms
- Runtime/deployment support
- Adopted by many groups in industry and academia:



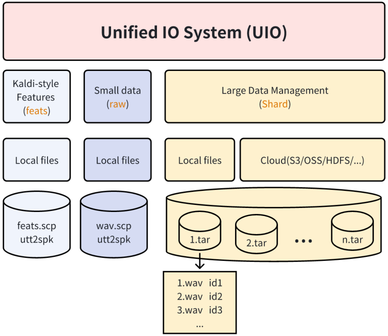


Figure: Unified I/O system

Unified I/O

- Also adopted in WeNet ASR
- Inspired by webdataset and tfrecord

Idea

- Raw: load wav and label files from disk (small-scale)
- Shard:
  - Pack many small files into larger shards
  - Read and decompress shards on the fly
- Feat: compatible with Kaldi-style features
- Efficiently load large-scale datasets

Step 1: Download and prepare metadata

```
if [ ${stage} -le 1 ] && [ ${stop_stage} -ge 1 ]; then
  echo "Prepare datasets ..."
  ./local/prepare_data.sh --stage 2 --stop_stage 4 --data ${data}
fi
```

Step 2: Convert train/test data

```
if [ ${stage} -le 2 ] && [ ${stop_stage} -ge 2 ]; then
  echo "Covert train and test data to ${data_type}..."
  for dset in vox2_dev vox1; do
    if [ $data_type == "shard" ]; then
      python tools/make_shard_list.py --num_utts_per_shard 1000 \
          --num_threads 16 \
          --prefix shards \
          --shuffle \
          ${data}/$dset/wav.scp ${data}/$dset/utt2spk \
          ${data}/$dset/shards ${data}/$dset/shard.list
    else
      python tools/make_raw_list.py ${data}/$dset/wav.scp \
          ${data}/$dset/utt2spk ${data}/$dset/raw.list
    fi
  done
fi
```

Step 3: Start training

```
if [ ${stage} -le 3 ] && [ ${stop_stage} -ge 3 ]; then
  echo "Start training ..."
  num_gpus=$(echo $gpus | awk -F ',' '{print NF}')
  torchrun --standalone --nnodes=1 --nproc_per_node=
      $num_gpus \
    wespeaker/bin/train.py --config $config \
      --exp_dir ${exp_dir} \
      --gpus $gpus \
      --num_avg ${num_avg} \
      --data_type "${data_type}" \
      --train_data ${data}/vox2_dev/${data_type}.list \
      --train_label ${data}/vox2_dev/utt2spk \
      --reverb_data ${data}/rirs/lmdb \
      --noise_data ${data}/musan/lmdb \
      ${checkpoint:+--checkpoint $checkpoint}
fi
```

Dataset config:

```
dataset_args:
  speed_perturb: True
  num_frms: 200
  aug_prob: 0.6
  # prob to add reverb & noise
      aug per sample
  fbank_args:
    num_mel_bins: 80
    frame_shift: 10
    frame_length: 25
    dither: 1.0
  spec_aug: False
  spec_aug_args:
    num_t_mask: 1
    num_f_mask: 1
    max_t: 10
    max_f: 8
    prob: 0.6
```

Data augmentation:

```
# add noise
dataset = Processor(dataset,
      processor.
      add_reverb_noise,
      reverb_data, noise_data,
       resample_rate, aug_prob
      )
# speed perturb
dataset = Processor(dataset,
      processor.speed_perturb,
       len(spk2id_dict))
# specaug
dataset = Processor(dataset,
      processor.spec_aug, **
      configs['spec_aug_args'
      ])
```

Architectures:

- ResNet family
- TDNN
- ECAPA-TDNN
- RepVGG
- CAM++
- ReDimNet
- Pretrained frontend (e.g., WavLM)

Pooling:

- TSTP
- ASTP
- MQMHASTP

Losses:

- add_margin
- arc_margin
- sphere
- sphereface2
- intertopk
- subcenter

Model config:

```
model: ResNet34
    # ECAPA, CAMPPlus, REPVGG,
        ResNet152
model_args:
  feat_dim: 80
  embed_dim: 256
  pooling_func: "TSTP" # TSTP,
        ASTP, MQMHASTP
  two_emb_layer: False
projection_args:
  project_type: "arc_margin"
  # add_margin, arc_margin,
        sphere, sphereface2,
        softmax, aam_intertopk
  scale: 32.0
```

Back-ends:

- Cosine
- LDA
- PLDA
- PSDA
- Adapt-PLDA

Others:

- Score normalization
- QMF-based calibration

Scoring:

```
if [ ${stage} -le 5 ] && [ ${stop_stage} -ge 5 ]; then
  echo "Score ..."
  local/score.sh \
    --stage 1 --stop-stage 2 \
    --data ${data} \
    --exp_dir $exp_dir \
    --trials "$trials"
fi

if [ ${stage} -le 6 ] && [ ${stop_stage} -ge 6 ]; then
  echo "Score norm ..."
  local/score_norm.sh \
    --stage 1 --stop-stage 3 \
    --score_norm_method $score_norm_method \
    --cohort_set vox2_dev \
    --top_n $top_n \
    --data ${data} \
    --exp_dir $exp_dir \
    --trials "$trials"
fi
```

Model	Params	vox1-O-clean	vox1-E-clean	vox1-H-clean
ReDimNetB0	1.0M	1.128	1.181	2.008
ReDimNetB3	3.2M	0.537	0.790	1.433
XVEC	4.61M	1.590	1.641	2.726
Res2Net34_Base	4.68M	1.234	1.232	2.162
ECAPA_TDNN_GLOB_c512	6.19M	0.782	1.005	1.824
RepVGG_TINY_A0	6.26M	0.824	0.953	1.709
Gemini_DFResNet114	6.53M	0.638	0.839	1.427
ResNet34	6.63M	0.659	0.821	1.437
ERes2Net34_Base	7.88M	0.744	0.896	1.603
CAM++	7.18M	0.659	0.803	1.569
ECAPA_TDNN_GLOB_c1024	14.6M	0.707	0.894	1.615
ResNet221	23.8M	0.505	0.676	1.213
SimAM_ResNet34 (VoxBlink2 Pretrain)	25.2M	0.372	0.559	0.997
ResNet293	28.6M	0.425	0.641	1.146
SimAM_ResNet100 (VoxBlink2 Pretrain)	50.2M	0.202	0.421	0.795
WavLM+EcapaTDNN		0.415	0.551	1.118

Figure: List of supported models

Export Jit:

```
if [ ${stage} -le 7 ] && [ ${stop_stage} -ge 7 ]; then
  echo "Export the best model ..."
  python wespeaker/bin/export_jit.py \
    --config $exp_dir/config.yaml \
    --checkpoint $exp_dir/models/avg_model.pt \
    --output_file $exp_dir/models/final.zip
fi
```

Export Onnx:

```
exp=exp  # Change it to your experiment dir
onnx_dir=onnx
python wespeaker/bin/export_onnx.py \
  --config $exp/config.yaml \
  --checkpoint $exp/avg_model.pt \
  --output_model $onnx_dir/final.onnx
```

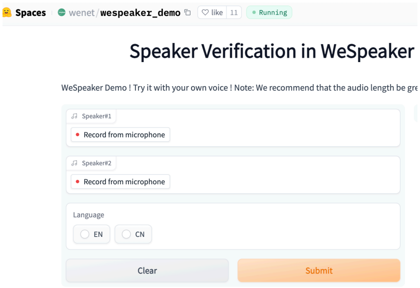


Figure: Wespeaker demo

Command line usage:

```
wespeaker --task embedding --audio_file audio.wav --output_file
    embedding.txt -g 0
wespeaker --task embedding_kaldi --wav_scp wav.scp --output_file /path/
    to/embedding -g 0
wespeaker --task similarity --audio_file audio.wav --audio_file2 audio2.
    wav --g 0
```

Python API:

```
import wespeaker

model = wespeaker.load_model('chinese')
# set_gpu to enable the cuda inference, number < 0 means using CPU
model.set_gpu(0)
embedding = model.extract_embedding('audio.wav')
utt_names, embeddings = model.extract_embedding_list('wav.scp')
similarity = model.compute_similarity('audio1.wav', 'audio2.wav')
diar_result = model.diarize('audio.wav')
```

WeSep Toolkit

The first open-source toolkit for Target Speaker Extraction.[69]

Main Contributions

- Propose WeSep, focusing on target speaker extraction
- Provide versatile speaker modeling capabilities
- Implement online data simulation and scalability
- Offer end-to-end training and deployment support

Technical Highlights

- Seamless integration with WeSpeaker
- Unified I/O data management
- Dynamic speaker mixing strategies
- Multiple fusion method support

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

UIO framework

- Efficient for both lab-scale and production-scale datasets
- Supports tens of thousands of hours of data
- Handles massive small-file shards
- Integrated in WeNet and WeSpeaker

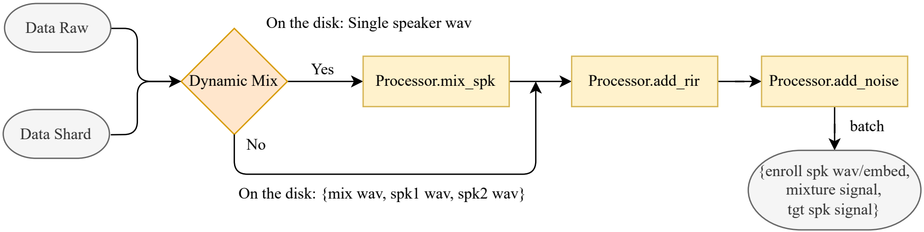


Figure: WeSep online data preparation pipeline (2-speaker example)

Online Data Simulation

Issues of traditional offline simulation:

- Store preprocessed data
- Large disk footprint
- Limited diversity

Advantages of online simulation:

- Save storage
- Create diverse training data
- Improve robustness
- Flexible data generation

Supported functions

- Online noise addition
- Reverb generation (standard RIR + fast random approx.)
- Dynamic speaker mixing

Seamless Integration with WeSpeaker

```
pip install git+https://github.com/wenet-e2e/wespeaker.git
```

```
# psudo-codes for integrating wespeaker models
from wespeaker import get_speaker_model
# TDNN/ECAPA/ResNet/CAM++/WavLM...
s = get_speaker_model(spk_model_name)(**spk_args)
m = BSRNN(**sep_args) # Or other backbones
m.speaker_model = s
if use_pretrain_spk_encoder:
    m.spk_model.load_state_dict(pretrain_path)
    m.speaker_model.freeze()
```

```
spk_fuse_type: 'multiply'
use_spk_transform: False
multi_fuse: False
joint_training: True
####### ResNet
spk_model: ResNet34
spk_model_init: False
#./wespeaker_models/model.pt
```

Supported Backbones

1. **ConvTasNet**
   - Convolutional network in time domain
   - Learn and estimate separation masks
   - Support Spex+ variants
2. **BSRNN**
   - Band-split RNN
   - Explicitly split spectrogram into bands
   - Fine-grained modeling
3. **DPCCN**
   - Dense-connected pyramid complex CNN
   - Combine DenseUNet, TCN and DenseNet
   - Improved separation
4. **TF-GridNet**
   - Operate in T-F domain
   - Stack multi-path blocks
   - Leverage local/global T-F information

Given $\mathbf{e}_s$ and intermediate outputs $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_T\}$:

1. **Concat**: replicate $\mathbf{e}_s$ and concatenate
2. **Add**: project and add element-wise
3. **Multiply**: project and multiply element-wise
4. **FiLM**: feature-wise linear modulation

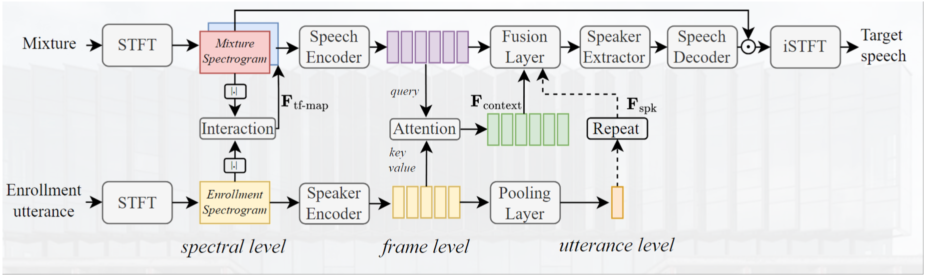
$$\mathbf{h}'_t = \gamma(\mathbf{e}_s) \odot \mathbf{h}_t + \beta(\mathbf{e}_s) \tag{3}$$

Why FiLM

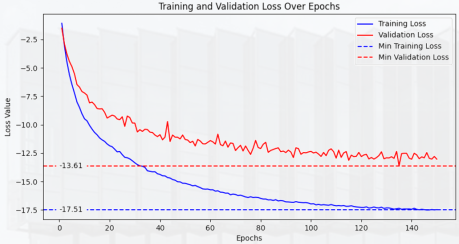
- γ and β are functions of $\mathbf{e}_s$
- $\odot$ is element-wise multiplication
- Learn an affine transform conditioned on $\mathbf{e}_s$

Beyond embedding-level guidance, add finer-grained context guidance[a]

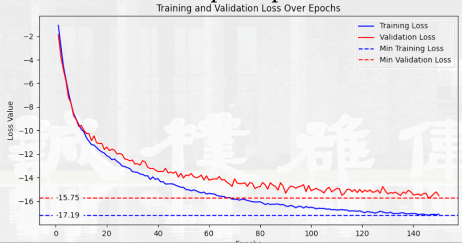
[a]Zhang et al., "Multi-level speaker representation for target speaker extraction".



Speaker embedding (Baseline)



TF-Map (Proposed)



Model	Speaker Model	Speaker Model	SI-SDRi on Libri2mix	Accuracy / %	Pub.
TD-SpeakerBeam	ResNet	Joint	13.03	95.2	ICASSP, 2020
SpEx+*	ResNet	Joint	13.41	-	Interspeech, 2020
sDPCCN	ConvNet	Joint	11.61	-	ICASSP, 2022
Target-Confusion*	ResNet	Joint	13.88	-	Interspeech, 2022
MC-SpEx*	ResNet	Joint	14.61	-	Interspeech, 2023
X-T-TasNet	d-vector	Pretrained	13.48	95.3	Interspeech, 2024
SSL-TD-SpeakerBeam	ResNet + WavLM	Pretrained	14.01	96.1	ICASSP, 2024
		Pretrained + Fine-tuning	14.65	97.0	
BSRNN	Campplus + SHuBERT	Pretrained	15.39	-	SPL, 2024
BSRNN	Ecapa-TDNN	Pretrained	**15.91**	**97.0**	Proposed
BSRNN			**17.99**	**98.6**	

- ✓ SOTA Performance, with simple but effective multi-level speaker modeling
- ✓ The generalization ability is largely enhanced (The gap between training and validation error)

Overall Summary

A holistic view of speaker modeling

- **Evolution**: from traditional GMM-UBM to deep learning
- **Multi-task applications**: speaker recognition, separation, synthesis, conversion, etc.
- **Challenges**: robustness, efficiency, interpretability, multi-modal fusion
- **Trends**: self-supervised learning, joint modeling, tooling

- **Beyond recognition**: speaker modeling is more than verification
- **Beyond embeddings**: speaker modeling is more than embedding learning
- **Task-oriented**: tailor methods to the target application
- **Evaluation-oriented**: use task-specific metrics and protocols

Baevski, Alexei et al. “wav2vec 2.0: A framework for self-supervised learning of speech representations”. In: Advances in neural information processing systems 33 (2020), pp. 12449–12460.

Balian, Julien et al. “Small footprint text-independent speaker verification for embedded systems”. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2021, pp. 6179–6183.

Belinkov, Yonatan and James Glass. “Analyzing hidden representations in end-to-end automatic speech recognition systems”. In: Advances in Neural Information Processing Systems 30 (2017).

Bilinski, Piotr et al. “Creating new voices using normalizing flows”. In: (2022).

- Cai, Danwei, Weiqing Wang, and Ming Li. "An iterative framework for self-supervised deep speaker representation learning". In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2021, pp. 6728–6732.
- Cai, Danwei et al. "Pretraining Conformer with ASR for Speaker Verification". In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2023, pp. 1–5.
- Cai, Linjun et al. "CS-CTCSCONV1D: Small footprint speaker verification with channel split time-channel-time separable 1-dimensional convolution". In: Proc. Interspeech 2022. 2022, pp. 326–330. DOI: 10.21437/Interspeech.2022-913.
- Cai, Weicheng, Jinkun Chen, and Ming Li. "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system". In: arXiv preprint arXiv:1804.05160 (2018).

Caron, Mathilde et al. "Emerging properties in self-supervised vision transformers". In: Proceedings of the IEEE/CVF international conference on computer vision. 2021, pp. 9650–9660.

Casanova, Edresson et al. "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone". In: International Conference on Machine Learning. PMLR. 2022, pp. 2709–2720.

Chen, Meiying and Zhiyao Duan. "ControlVC: Zero-Shot Voice Conversion with Time-Varying Controls on Pitch and Rhythm". In: arXiv preprint arXiv:2209.11866 (2022).

Chen, Sanyuan et al. "Unispeech-sat: Universal speech representation learning with speaker aware pre-training". In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2022, pp. 6152–6156.

- Chen, Sanyuan et al. "Wavlm: Large-scale self-supervised pre-training for full stack speech processing". In: IEEE Journal of Selected Topics in Signal Processing 16.6 (2022), pp. 1505–1518.
- Chen, Ting et al. "A simple framework for contrastive learning of visual representations". In: International conference on machine learning. PMLR. 2020, pp. 1597–1607.
- Chen, Zhengyang et al. "A comprehensive study on self-supervised distillation for speaker representation learning". In: 2022 IEEE Spoken Language Technology Workshop (SLT). IEEE. 2023, pp. 599–604.
- Chen, Zhengyang et al. "Channel invariant speaker embedding learning with joint multi-task and adversarial training". In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2020, pp. 6574–6578.

- Chen, Zhengyang et al. “Large-scale self-supervised speech representation learning for automatic speaker verification”. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2022, pp. 6147–6151.
- Chowdhury, Shammur Absar, Nadir Durrani, and Ahmed Ali. “What do end-to-end speech models learn about speaker, language and channel information? a layer-wise and neuron-level analysis”. In: Computer Speech Language 83 (2023), p. 101539.
- Chung, Joon Son et al. “In defence of metric learning for speaker recognition”. In: arXiv preprint arXiv:2003.11982 (2020).
- — .“In Defence of Metric Learning for Speaker Recognition”. In: Proc. Interspeech 2020. 2020, pp. 2977–2981. DOI: 10.21437/Interspeech.2020-1064.
- Dehak, Najim et al. “Front-end factor analysis for speaker verification”. In: IEEE Transactions on Audio, Speech, and Language Processing 19.4 (2010), pp. 788–798.

- Delcroix, Marc et al. "Single channel target speaker extraction and recognition with speaker beam". In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE. 2018, pp. 5554–5558.
- Desplanques, Brecht, Jenthe Thienpondt, and Kris Demuynck. "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification". In: arXiv preprint arXiv:2005.07143 (2020).
- Du, Chenpeng et al. "UniCATS: A Unified Context-Aware Text-to-Speech Framework with Contextual VQ-Diffusion and Vocoding". In: arXiv preprint arXiv:2306.07547 (2023).
- Fujita, Yusuke et al. "End-to-end neural speaker diarization with self-attention". In: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE. 2019, pp. 296–303.

- Ge, Meng et al. "Spex+: A complete time domain speaker extraction network". In: arXiv preprint arXiv:2005.04686 (2020).
- Han, Bing, Zhengyang Chen, and Yanmin Qian. "Self-Supervised Learning with Cluster-Aware-DINO for High-Performance Robust Speaker Verification". In: arXiv preprint arXiv:2304.05754 (2023).
- — ."Self-supervised speaker verification using dynamic loss-gate and label correction". In: arXiv preprint arXiv:2208.01928 (2022).
- He, Xinlu and Jacob Whitehill. "Survey of End-to-End Multi-Speaker Automatic Speech Recognition for Monaural Audio". In: arXiv preprint arXiv:2505.10975 (2025).
- Horiguchi, Shota et al. "Encoder-decoder based attractors for end-to-end neural diarization". In: IEEE/ACM Transactions on Audio, Speech, and Language Processing 30 (2022), pp. 1493–1507.

南京大學 NANJING UNIVERSITY 智能科学与技术学院 School of Intelligence Science and Technology

- Hsu, Wei-Ning et al. “Hubert: Self-supervised speech representation learning by masked prediction of hidden units”. In: IEEE/ACM Transactions on Audio, Speech, and Language Processing 29 (2021), pp. 3451–3460.
- Huang, Wen et al. “Enhancing Cross-Domain Speaker Verification through Multi-Level Domain Adapters”. In: (2023).
- Huh, Jaesung et al. “Augmentation adversarial training for self-supervised speaker recognition”. In: arXiv preprint arXiv:2007.12085 (2020).
- Hussain, Shehzeen et al. “ACE-VC: Adaptive and Controllable Voice Conversion Using Explicitly Disentangled Self-Supervised Speech Representations”. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2023, pp. 1–5.

- Jia, Ye et al. "Transfer learning from speaker verification to multispeaker text-to-speech synthesis". In: Advances in neural information processing systems 31 (2018).
- Jin, Yufeng et al. "Cross-modal distillation for speaker recognition". In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 37. 11. 2023, pp. 12977–12985.
- Jin, Zezhong, Youzhi Tu, and Man-Wai Mak. "Phonetic-aware speaker embedding for far-field speaker verification". In: arXiv preprint arXiv:2311.15627 (2023).
- Le, Matthew et al. "Voicebox: Text-guided multilingual universal speech generation at scale". In: arXiv preprint arXiv:2306.15687 (2023).
- Li, Fengfu, Bo Zhang, and Bin Liu. "Ternary weight networks". In: arXiv preprint arXiv:1605.04711 (2016).

- Li, Jianchen, Jiqing Han, and Hongwei Song. "CDMA: Cross-Domain Distance Metric Adaptation for Speaker Verification". In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2022, pp. 7197–7201.
- Li, Jingyu et al. "Model Compression for DNN-based Speaker Verification Using Weight Quantization". In: Proc. INTERSPEECH 2023. 2023, pp. 1988–1992. DOI: 10.21437/Interspeech.2023-1524.
- Li, Pengqi et al. "Reliable visualization for deep speaker recognition". In: arXiv preprint arXiv:2204.03852 (2022).
- — ."Visualizing data augmentation in deep speaker recognition". In: arXiv preprint arXiv:2305.16070 (2023).

- Li, Qingjian et al. "Towards Lightweight Applications: Asymmetric Enroll-Verify Structure for Speaker Verification". In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processi 2022, pp. 7067–7071. DOI: 10.1109/ICASSP43922.2022.9746247.
- Li, Rongjin, Weibin Zhang, and Dongpeng Chen. "The coral++ algorithm for unsupervised domain adaptation of speaker recognition". In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2022, pp. 7172–7176.
- Li, Ze, Ming Cheng, and Ming Li. "Enhancing Speaker Verification with w2v-BERT 2.0 and Knowledge Distillation guided Structured Pruning". In: arXiv preprint arXiv:2510.04213 (2025).
- Liao, Dexin et al. "Towards a unified conformer structure: from asr to asv task". In: ICASSP 2023. IEEE. 2023, pp. 1–5.

南京大学 NANJING UNIVERSITY 智能科学与技术学院 School of Intelligence Science and Technology

- Liu, Bei, Zhengyang Chen, and Yanmin Qian. "Depth-First Neural Architecture With Attentive Feature Fusion for Efficient Speaker Verification". In: IEEE/ACM Transactions on Audio, Speech, and Language Processing 31 (2023), pp. 1825–1838. DOI: 10.1109/TASLP.2023.3273417.
- Liu, Bei and Yanmin Qian. "Reversible Neural Networks for Memory-Efficient Speaker Verification". In: Proc. INTERSPEECH 2023. 2023, pp. 3127–3131. DOI: 10.21437/Interspeech.2023-844.
- Liu, Bei, Haoyu Wang, and Yanmin Qian. "Extremely Low Bit Quantization for Mobile Speaker Verification Systems Under 1MB Memory". In: Proc. INTERSPEECH 2023. 2023, pp. 1973–1977. DOI: 10.21437/Interspeech.2023-800.
- Liu, Bei et al. "Self-Knowledge Distillation via Feature Enhancement for Speaker Verification". In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processi 2022, pp. 7542–7546. DOI: 10.1109/ICASSP43922.2022.9746529.

- Ma, Yi et al. "ExPO: Explainable Phonetic Trait-Oriented Network for Speaker Verification". In: IEEE Signal Processing Letters (2025).
- Medennikov, Ivan et al. "Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario". In: arXiv preprint arXiv:2005.07272 (2020).
- Peng, Junyi et al. "Parameter-efficient transfer learning of pre-trained Transformer models for speaker verification using adapters". In: ICASSP 2023. IEEE. 2023, pp. 1–5.
- Qian, Yanmin, Zhengyang Chen, and Shuai Wang. "Audio-visual deep neural network for robust person verification". In: IEEE/ACM Transactions on Audio, Speech, and Language Processing 29 (2021), pp. 1079–1092.

- Raj, Desh et al. “Probing the information encoded in x-vectors”. In: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE. 2019, pp. 726–733.
- Ren, Yiming et al. “Can Audio Large Language Models Verify Speaker Identity?” In: arXiv preprint arXiv:2509.19755 (2025).
- Reynolds, Douglas A et al. “Gaussian mixture models.”. In: Encyclopedia of biometrics 741.659-663 (2009).
- Rohdin, Johan et al. “Speaker verification using end-to-end adversarial language adaptation”. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2019, pp. 6006–6010.

- Shimizu, Reo et al. "PromptTTS++: Controlling Speaker Identity in Prompt-Based Text-to-Speech Using Natural Language Descriptions". In: arXiv preprint arXiv:2309.08140 (2023).
- Snyder, David et al. "X-vectors: Robust dnn embeddings for speaker recognition". In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE. 2018, pp. 5329–5333.
- Stan, Adriana and Johannah O'Mahony. "An analysis on the effects of speaker embedding choice in non auto-regressive TTS". In: arXiv preprint arXiv:2307.09898 (2023).
- Stanton, Daisy et al. "Speaker generation". In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2022, pp. 7897–7901.

Tao, Ruijie et al. "Self-supervised speaker recognition with loss-gated learning". In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2022, pp. 6142–6146.

Variani, Ehsan et al. "Deep neural networks for small footprint text-dependent speaker verification". In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE. 2014, pp. 4052–4056.

Wan, Li et al. "Generalized end-to-end loss for speaker verification". In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE. 2018, pp. 4879–4883.

Wang, Chengyi et al. "Neural codec language models are zero-shot text to speech synthesizers". In: arXiv preprint arXiv:2301.02111 (2023).

- Wang, Haoyu et al. "Adaptive Neural Network Quantization For Lightweight Speaker Verification". In: Proc. INTERSPEECH 2023. 2023, pp. 5331–5335. DOI: 10.21437/Interspeech.2023-927.
- Wang, Shuai, Yanmin Qian, and Kai Yu. "What does the speaker embedding encode?" In: 2017.
- Wang, Shuai et al. "Knowledge Distillation for Small Foot-print Deep Speaker Embedding". In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processi 2019.
- Wang, Shuai et al. "Wesep: A scalable and flexible toolkit towards generalizable target speaker extraction". In: arXiv preprint arXiv:2409.15799 (2024).
- Wu, Xiaoliang et al. "Explainable attribute-based speaker verification". In: arXiv preprint arXiv:2405.19796 (2024).

- Wu, Yihan et al. "Adaspeech 4: Adaptive text to speech in zero-shot scenarios". In: arXiv preprint arXiv:2204.00436 (2022).
- Xia, Wei, Jing Huang, and John HL Hansen. "Cross-lingual text-independent speaker verification using unsupervised adversarial discriminative domain adaptation". In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2019, pp. 5816–5820.
- Xu, J. et al. "Mixed Precision Low-Bit Quantization of Neural Network Language Models for Speech Recognition". In: IEEE/ACM Transactions on Audio, Speech, and Language Processing 29 (2021), pp. 3679–3693.

NANJING UNIVERSITY 智能科学与技术学院 School of Intelligence Science and Technology

- Zhang, Haoran, Yuexian Zou, and Helin Wang. "Contrastive self-supervised learning for text-independent speaker verification". In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2021, pp. 6713–6717.
- Zhang, Haozhe et al. "SIG-VC: A Speaker Information Guided Zero-Shot Voice Conversion System for Both Human Beings and Machines". In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2022, pp. 6567–65571.
- Zhang, Ke et al. "Multi-level speaker representation for target speaker extraction". In: ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processin IEEE. 2025, pp. 1–5.
- Zhang, Leying, Zhengyang Chen, and Yanmin Qian. "Knowledge Distillation from Multi-Modality to Single-Modality for Person Verification". In: Proc. Interspeech 2021 (2021), pp. 1897–1901.

- Zhang, Leying, Zhengyang Chen, and Yanmin Qian. "Knowledge Distillation from Multi-Modality to Single-Modality for Person Verification". In: Proc. Interspeech 2021. 2021, pp. 1897–1901. DOI: 10.21437/Interspeech.2021-2119.
- Zhang, Yongmao et al. "PromptSpeaker: Speaker Generation Based on Text Descriptions". In: arXiv preprint arXiv:2310.05001 (2023).
- Zhao, Zifeng et al. "Probing Deep Speaker Embeddings for Speaker-related Tasks". In: arXiv preprint arXiv:2212.07068 (2022).
- Zheng, Rong, Shuwu Zhang, and Bo Xu. "Text-independent speaker identification using gmm-ubm and frame level likelihood normalization". In: 2004 International Symposium on Chinese Spoken Language Processing. IEEE. 2004, pp. 289–292.
- Zheng, Siqi, Yun Lei, and Hongbin Suo. "Phonetically-Aware Coupled Network For Short Duration Text-Independent Speaker Verification.". In: INTERSPEECH. 2020, pp. 926–930.

Zhu, Tinglong, Xiaoyi Qin, and Ming Li. "Binary Neural Network for Speaker Verification". In: Proc. Interspeech 2021. 2021, pp. 86–90. DOI: 10.21437/Interspeech.2021-600.

— ."Binary Neural Network for Speaker Verification". In: Proc. Interspeech 2021. 2021, pp. 86–90. DOI: 10.21437/Interspeech.2021-600.

Zmolikova, Katerina et al. "Neural Target Speech Extraction: An overview". In: IEEE Signal Processing Magazine 40.3 (2023), pp. 8–29.