MAXIMAL SPEAKER SEPARABILITY VIA ROBUST SPEAKER REPRESENTATION LEARNING

ZHE LI

PhD

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University
Department of Electrical and Electronic Engineering

Maximal Speaker Separability via Robust Speaker Representation Learning

Zhe Li

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
April 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Zhe Li

Abstract

Speaker representation learning aims to extract compact, discriminative embeddings that encapsulate unique vocal characteristics regardless of linguistic content or environmental conditions. The objective is to learn an embedding space with two key properties: same-class compactness, where embeddings from the same speaker are closely clustered, and different-class dispersion, where embeddings from different speakers are well separated. However, existing methods face several challenges. First, conventional speaker verification methods treat the task as a classification problem, relying on softmax-based loss functions to maximize inter-class differences. However, these loss functions often struggle to reduce intra-class variation. Second, directly applying a pre-trained model to speaker verification can only achieve sub-optimal performance because the pre-trained model is insufficient for extracting task-specific features, leading to limited transferability. Full fine-tuning of these models introduces significant computational and storage costs while risking catastrophic forgetting. Third, while the pre-trained speech models offer robust feature representations, their effectiveness relies on an unrealistic assumption: the speaker identity information and the linguistic content in the representations can be easily disentangled.

To address these challenges, we propose three key solutions in this thesis. First, we propose a supervised contrastive learning framework incorporating an additive angular margin to effectively reduce intra-class variation. By maximizing the mutual information between frame-level features and speaker representations, our method

preserves nonshared speaker information across diverse augmentations. Extensive evaluations on CN-Celeb, VoxCeleb, and CU-MARVEL datasets demonstrate that the resulting ECAPA-TDNN embedding space exhibits robust inter-speaker separability and intra-speaker consistency. Second, we investigate parameter-efficient fine-tuning strategies for pre-trained Transformer models in speaker verification. By integrating dynamic prompt tuning—where prompts are clustered based on speaker-specific traits—and incorporating spectral information into a LoRA-based adaptation process, our approach efficiently captures task-relevant features while significantly reducing memory and computational overhead. Third, we introduce a diffusion-based approach within a variational autoencoder framework to disentangle speaker timbre from spoken content. Leveraging a conditional diffusion model in the latent space, our method yields content-invariant speaker embeddings that are resilient to language mismatches, outperforming traditional sequential VAE techniques. Experiments on the VoxCeleb and CN-Celeb datasets demonstrate that our method effectively isolates speaker features from speech content using pre-trained speech representations.

Publications Arising from the Thesis

Refereed Journal Articles

1. **Zhe Li**, Man-Wai Mak, Mert Pilanci, and Helen Meng, "Mutual Information-Enhanced Contrastive Learning with Margin for Maximal Speaker Separability," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 33, pp. 2961–2972, 2025. (Published)

2. **Zhe Li**, Man-Wai Mak, Mert Pilanci, Jen-Tzung Chien and Helen Meng, "Disentangling Speech Representations Learning with Latent Diffusion for Speaker Verification," in *IEEE/ACM Transactions on Audio, Speech and Language Processing* (Accepted)

3. **Zhe Li**, Man-Wai Mak, Mert Pilanci, Hung Yi Lee, and Helen Meng, "Towards a Unified View of Parameter-Efficient Speech Pretrained Models for Speaker Verification," in *IEEE/ACM Transactions on Audio, Speech and Language Processing* (Submitted)

Conference Publications

1. **Zhe Li**, Man-Wai Mak, Jen-Tzung Chien, Mert Pilanci, Zezhong Jin, and Helen Meng, "Disentangling Speaker and Content in Pre-trained Speech Models with

Latent Diffusion for Robust Speaker Verification," in *Proceedings of InterSpeech*, Rotterdam, Netherlands, 2025.

2. **Zhe Li**, Man-Wai Mak, Mert Pilanci, Hung Yi Lee, and Helen Meng, "Spectral-Aware Low-Rank Adaptation for Speaker Verification," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hyderabad, India, 2025.

3. **Zhe Li**, Man-Wai Mak, Hung Yi Lee, and Helen Meng, "Parameter-efficient fine-tuning of speaker-aware dynamic prompts for speaker verification," in *Proceedings of InterSpeech*, Kos, Greece, 2024.

4. **Zhe Li**, Man-Wai Mak, and Helen Meng, "Dual parameter-efficient fine-tuning for speaker representation via speaker prompt tuning and adapters," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seoul, South Korea, 2024.

5. **Zhe Li**, Man-Wai Mak, and Helen Meng, "Discriminative Speaker Representation via Contrastive Learning with Class-Aware Attention in Angular Space," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023.

6. **Zhe Li** and Man-Wai Mak, "Speaker Representation Learning via Contrastive Loss with Maximal Speaker Separability," in *Proceedings of the 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 962–967, Chiang Mai, Thailand, 2022.

7. Zezhong Jin, Youzhi Tu, **Zhe Li**, Zilong Huang, Chongxin Gan, and Man-Wai Mak, "Denoising Student Features with Diffusion Models for Knowledge Distillation in Speaker Verification," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hyderabad, India, 2025.

8. Chong-xin Gan, **Zhe Li**, Zezhong Jin, Zilong Huang, Man-Wai Mak, and Kong Aik Lee, "IDIR: Identifying and Distilling Informative Relations for Speaker Verification," in *Proceedings of InterSpeech*, Rotterdam, Netherlands, 2025.

Acknowledgments

I would like to express my sincerest gratitude to my supervisor, Professor Man-wai Mak. Professor Mak has profound expertise in speech processing, machine learning, and deep learning theory, and he provided numerous valuable and constructive suggestions, especially regarding the proper use of mathematical equations in deep learning. His meticulous review of my papers has significantly improved my writing style and greatly enhanced the clarity and logical coherence of my arguments. I was deeply impressed by Professor Mak's patience and encouragement, which substantially elevated my proficiency in academic writing in English, helped me continuously correct shortcomings in my research, and gradually established a systematic and standardized approach to writing. Before every conference, he arranged rehearsals and patiently guided me on how to effectively convey information to the audience, which was crucial for improving my presentation skills. Professor Mak is not only exceptionally knowledgeable but also humorous, I am proud to say he is the best PhD supervisor I have ever met.

I would like to express my gratitude to Professor Helen Meng from The Chinese University of Hong Kong. It was a pleasure to collaborate with the members of her team, and Professor Meng imparted many research philosophies that not only benefited my PhD studies but also had a profound influence on my subsequent academic career.

I also wish to thank Professor Mert Pilanci from Stanford University for hosting my visiting research at Stanford University. His guidance and support opened up entirely

new perspectives for my research.

I am grateful to PhD candidates Zezhong Jin and Chongxin Gan, as well as PhD student Zilong Huang. They are not only members of Professor Mak's research group but also my good friends, and their companionship made my PhD journey truly enjoyable.

Lastly, I would like to thank my parents. Whenever I think of them, I feel a surge of strength and warmth, which grants me the courage and ability to overcome challenges. Through their unwavering support, I have found genuine joy and freedom in life.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Background

Speaker representation learning aims to derive compact and discriminative embeddings from speech signals, capturing the unique acoustic traits of individuals while ensuring consistency across varying linguistic content and acoustic environments. This process involves analyzing vocal patterns and attributes to represent and identify individual characteristics, forming a cornerstone of modern speech-processing technologies. The derived representations encapsulate critical information about speaker identity and vocal features, enabling direct voice-based identification—a capability with broad implications across multiple domains.

The significance of speaker representation learning extends beyond fundamental speaker recognition tasks [10, 11, 12, 13, 14], finding applications in diverse areas, including biometric authentication [15], surveillance systems [16], personalized services [17], and forensic analysis [18]. Furthermore, its impact spans speaker diarization [19, 20] and speech generation tasks such as voice cloning [21], text-to-speech synthesis [22], and voice conversion [23].

The ability to learn and replicate speaker-specific vocal characteristics has revolutionized personalized speech generation, enhancing user experience in virtual assistants and interactive gaming environments. Moreover, voice transformation techniques facilitate speaker anonymization [24, 25] by converting identifiable voices into pseudo-speaker representations, thereby addressing privacy concerns in data security applications.

The advent of using deep neural networks as powerful feature extractors has significantly advanced representation learning [26], focusing on deriving universal, low-dimensional features with robust generalization capabilities. The evolution of speaker modeling has witnessed several milestones, beginning with the d-vector approach [27], which pioneered neural network-based speaker representation extraction. Subsequent advancements, including multi-task learning frameworks [28], segmental-level aggregation techniques [29, 30], and end-to-end metric learning [31], gradually shifted the paradigm from traditional i-vector approaches to the more effective x-vector systems [32, 33].

The widespread adoption of x-vectors was further accelerated by their integration into the Kaldi toolkit [34], leading to substantial performance improvements across standard speaker recognition benchmarks. This breakthrough spurred extensive research into alternative neural architectures [3, 35, 36], innovative training objectives [37, 38, 39, 40], and advanced aggregation methods [37, 41, 42, 43], culminating in the current state-of-the-art framework characterized by segment-level training and margin-based optimization.

As speaker modeling technology continues to evolve, emerging applications present new challenges in areas such as model robustness, computational efficiency, unsupervised learning, and multimodal fusion. These practical considerations necessitate careful optimization of representation learning objectives, which can be formalized as follows:

1. **Discrimination**: Representation vectors should maximize inter-class variance, ensuring distinct embeddings for different speakers.

2. **Robustness**: Intra-class variance should be minimized, maintaining consistent representations for the same speaker across varying utterances, environmental conditions, and channel characteristics.

3. **Compactness**: The embedding dimensionality should be optimized for computational and storage efficiency while preserving sufficient discriminative information for accurate identification.

1.2 Research Contents

Contrastive learning across various augmentations of the same utterance can enhance the speaker representations' ability to distinguish new speakers. This dissertation introduces a supervised contrastive learning objective that optimizes a speaker embedding space using label information from training data. Besides augmenting different segments of an utterance to form a positive pair, our approach generates multiple positive pairs by augmenting various utterances from the same speaker. However, employing contrastive learning for speaker verification presents two challenges: (1) the softmax loss is ineffective in reducing intra-class variation, and (2) previous research has shown that contrastive learning can share information across the augmented views of an object but could discard unshared information, suggesting that it is essential to keep nonshared speaker information across the augmented views when constructing a speaker representation space. To overcome the first challenge, we incorporate an additive angular margin in the contrastive loss. For the second challenge, we maximize the mutual information (MI) between the acoustic features and speaker representations to extract the nonshared information. Evaluations on CN-Celeb, VoxCeleb, and CU-MARVEL validate that our new learning objective enables ECAPA-TDNN

to identify an embedding space that exhibits robust speaker discrimination.

Fine-tuning pre-trained Transformer models (PTMs) for speech tasks in a parameter-efficient fine-tuning (PEFT) optimizes memory usage while leveraging rich representations from large-scale unlabeled data. Although effective, interconnections between various PEFT methods are not fully understood. This dissertation analyzes state-of-the-art PEFT methods and introduces a unified framework to clarify their interrelationships. Specifically, we employ a dynamic prompt tuning strategy that selects optimal prompts from a predefined pool, ensuring each prompt is fine-tuned by its most closely matched speaker to capture speaker-specific traits. The goal is to cluster the prompts in the pool according to speaker traits, improving speaker prediction in the downstream classifier while preserving the flexibility of the pre-trained Transformers. Additionally, we improve existing PEFT techniques by incorporating spectral information from pre-trained weight matrices into the LoRA-based fine-tuning process. Extensive experiments on VoxCeleb, CNCeleb, and CU-MARVEL demonstrate that the proposed method offers a memory- and computation-efficient solution for fine-tuning pre-trained Transformers.

Disentangled speech representation learning for speaker verification aims to separate spoken content and speaker timbre into distinct representations. However, existing variational autoencoder (VAE)–based methods for speech disentanglement rely on latent variables that lack semantic meaning, limiting their effectiveness for speaker verification. To address this limitation, we propose a diffusion-based method that disentangles and separates speaker features and speech content in the latent space. Building upon the VAE framework, we employ a speaker encoder to learn latent variables representing speaker features while using frame-specific latent variables to capture content. Unlike previous sequential VAE approaches, our method utilizes a conditional diffusion model in the latent space to derive speaker-aware representations. Experiments on the VoxCeleb and CN-Celeb datasets demonstrate that our method effectively isolates speaker features from speech content using pre-trained

speech representations. The learned embeddings are robust to language mismatches since the speaker embeddings become content-invariant after content removal.

1.3 Organization of the Dissertation

This thesis is organized as follows:

Chapter 1 introduces the research background and content.

Chapter 2 reviews the literature on speaker verification, establishing the context and identifying the gaps our research aims to address.

Chapter 3 presents our contrastive learning framework designed to maximize speaker separability. Specifically, we introduce a supervised contrastive loss with an additive angular margin to effectively reduce intra-class variation. Moreover, our approach preserves nonshared speaker information across diverse augmentations by maximizing the mutual information between acoustic features and speaker representations. Experiments on CN-Celeb, VoxCeleb, and CU-MARVEL datasets demonstrate the robustness of the proposed method.

Chapter 4 and Chapter 5 focus on parameter-efficient fine-tuning of pre-trained Transformer models for speaker verification. Our approach efficiently captures task-relevant features by employing dynamic prompt tuning, which clusters prompts based on speaker-specific traits. Additionally, incorporating spectral information from pre-trained weight matrices into a LoRA-based fine-tuning process further improves VoxCeleb, CNCeleb, and CU-MARVEL performance while reducing memory and computational demands.

Chapter 6 addresses the challenge of disentangled speech representation learning for speaker verification. We propose a diffusion-based method within a VAE framework that separates speaker timbre from spoken content in the latent space via a condi-

tional diffusion model. This strategy yields content-invariant speaker embeddings, as validated by experiments on VoxCeleb and CN-Celeb datasets, and further enhances speaker-discriminative representation.

Finally, chapter 7 concludes with a summary of our contributions, a critical discussion of the experimental findings, and an outlook on future research directions.

Chapter 2

Literature Review

2.1 Speaker Encoders

In this section, we survey a range of neural architectures that have been proposed for learning speaker representations [44, 45, 46, 47]. We then examine the dominant encoder designs employed across different learning frameworks.

Typically, speech processing begins with a framing step, which yields frame-level input features. This approach, however, creates a gap between the low-level frame representations and the desired higher-level, segment-based speaker embeddings. As a result, methods for speaker embedding learning can be classified into two categories based on the stage at which optimization occurs: those operating at the frame level and those at the segment level. In frame-level approaches, the aggregation of frame representations is performed as a post-processing step, whereas segment-level approaches embed the aggregation process directly within the neural network to facilitate end-to-end optimization.

2.1.1 Frame-Level Optimized Speaker Encoders

The d-vector approach [27] operates at the frame level, optimizing a cross-entropy (CE) loss function for individual frames. Given an input speech feature sequence $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_T\} \in \mathbb{R}^{T \times D}$, where T is the number of frames and D is the feature dimension, the network produces frame-level outputs $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_T\} \in \mathbb{R}^{T \times D'}$ from the hidden layers near the output. These frame-level representations are then aggregated into a sentence-level speaker embedding, typically via averaging:

$$\mathbf{v}_{\text{dvec}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{f}_t. \tag{2.1}$$

While the d-vector was an early milestone in neural-network-based speaker representation learning, its limited architecture and frame-level optimization hinders its scalability and adoption in large-scale applications.

2.1.2 Segment-Level Optimized Speaker Encoders

To address the granularity mismatch between training and inference in frame-level methods, segment-level optimization techniques explicitly bind frames from the same utterance and optimize at the segment level. This is achieved by incorporating an aggregation layer into the network, which maps frame-level features to a unified segment-level representation.

Aggregation Layers

Aggregation layers in speaker representation learning range from simple statistical methods to more sophisticated mechanisms like attention [41, 48, 49, 50] and dictionary learning [37]. Two widely used methods are Temporal Average Pooling (TAP) and Temporal Statistics Pooling (TSTP). TAP computes the mean of frame-level

features, while TSTP concatenates the mean and standard deviation vectors to capture additional variability. Although TSTP generally outperforms TAP, studies such as [42, 51] have shown that using only variance (Temporal Standard Deviation Pooling, TSDP) can also be effective. Higher-order statistics have been explored but have not yielded significant performance gains.

Time-Delay Neural Networks

The x-vector framework [32, 33] represents a significant advancement over d-vector by introducing segment-level optimization and employing a Time-Delay Neural Network (TDNN) as the feature extractor. TDNNs use one-dimensional convolutions to hierarchically expand the receptive field, enabling more robust modeling of temporal dependencies. As the first widely adopted deep neural network (DNN) for speaker embedding, x-vector sets a new standard for segment-level speaker representation.

ECAPA-TDNN (Emphasized Channel Attention, Propagation, and Aggregation in TDNN) [3] further enhances TDNNs through several key innovations: 1) A channel attention mechanism to emphasize speaker-specific features; 2) Multi-scale feature learning for capturing patterns at varying temporal resolutions; 3) Multi-level feature aggregation to integrate information across layers; 4) Residual connections to improve gradient flow; and 5) Dense connections to preserve information from earlier layers.

These improvements reflect the broader evolution of speaker encoders, which we discuss in detail in Section 2.1.5.

Residual Networks

While increasing model depth has been a common strategy for improving performance, researchers have found that indiscriminate depth scaling does not always yield better results. The introduction of Deep Residual Neural Networks (ResNet) by He

et al. [52] marked a significant breakthrough in image recognition and deep learning. ResNet addresses the vanishing and exploding gradient problems through residual learning, enabling the construction of deeper networks with improved performance. Since its inception, ResNet has set new benchmarks in various image-related tasks and has been adapted for speaker modeling. For instance, Joon *et al.* [36] employed the ResNet34 and ResNet50 architectures for speaker verification, but their direct adaptation from image processing led to suboptimal results. Subsequent work by Li *et al.* [53] introduced the Inception-ResNet structure, exploring its robustness across different speech durations. Studies such as [35] and [37] further refined ResNet by removing shallow pooling modules, leading to the widely adopted r-vector architecture.

Transformer-Based Models

The Transformer architecture, introduced in [54], has achieved remarkable success across natural language processing (NLP) [55], computer vision (CV) [56], and speech-related tasks [57]. Unlike traditional recurrent neural networks (RNNs) and convolutional neural networks (CNNs), Transformers leverage self-attention mechanisms to model global dependencies and enable parallel computation. Initial attempts to apply vanilla Transformers to speaker verification tasks [58, 59] were hindered by their inability to capture local features effectively. To address this limitation, researchers have explored hybrid approaches, such as incorporating local attention mechanisms [60, 61] and integrating CNNs to balance global and local feature modeling. Recent work has further advanced this direction by inserting CNNs into the self-attention modules [47, 62, 63, 64] or using dual-branch architectures with cross-fusion [65, 66, 67]. These innovations have enabled Transformer-based models to achieve performance comparable to state-of-the-art convolutional models in speaker verification tasks.

2.1.3 Frame-Level and Segment-Level Optimization

Recent advancements in neural network-based speaker encoders have seen a shift from frame-level to segment-level optimization. The latter is generally more suitable for practical applications as it facilitates the learning of speaker embeddings at the utterance level. However, there remain scenarios where frame-level modeling is advantageous. For example, Tawara *et al.* [68] extended segment-level multi-task and adversarial training techniques [69] to frame-level speaker encoders, demonstrating that frame-level embeddings perform better in short-duration speaker recognition tasks (e.g., under 1.4 seconds). Intuitively, fine-grained, frame-level representations should be more effective in short-duration scenarios. Similarly, in speaker diarization tasks, such as those in EEND [70] and TS-VAD [71], frame-level modeling is essential due to the precise timestamping required for speaker diarization.

2.1.4 1D Convolution and 2D Convolution

Speech signals are inherently one-dimensional and sequential in nature. Thus, models processing raw waveforms, such as RawNet [72, 73, 74] and SincNet [75], often use one-dimensional convolutional backbones. However, when the model input is converted to time-frequency representations like spectrograms or filter banks (Fbank), two-dimensional convolutions can be applied.

Models like TDNN [3, 32, 33] typically use one-dimensional convolutions applied across the time axis. These convolutions are computationally efficient, enabling fast processing of long time-series data. Additionally, by using larger convolutional kernels (e.g., dilated convolutions in TDNN), TDNNs are capable of modeling long-range dependencies. However, this approach has limitations in capturing relationships between different frequency bands, and does not fully exploit the complexity of both time and frequency dimensions.

In contrast, two-dimensional convolutions, which span both the time and frequency dimensions, offer a more comprehensive method for modeling the time-frequency structure of speech. These models, such as those represented by ResNet [35, 37, 76], can better capture complex relationships across both time and frequency, but they require more computational resources and memory, leading to longer training time and a higher risk of overfitting. Consequently, the choice between 1D and 2D convolutions depends on the task, computational constraints, and dataset size. Nevertheless, using 2D convolutions in layers associated with time-frequency representations has shown promising results. For instance, ECAPA-TDNN2 [77] introduces 2D convolutions in the lower layers, a strategy also adopted in ECAPA-CNN-TDNN [78] and MFA-TDNN [79], where a 2D-CNN module is appended before the original ECAPA-TDNN architecture.

2.1.5 Taxonomies of Speaker Encoder Improvement

Deepening Network Architecture

To facilitate the construction of deeper neural networks, various types of highway connections, such as residual connections [3, 35] and dense connections [80], have been introduced. These architectures help mitigate the vanishing gradient problem, making it feasible to train deeper models and achieve better performance.

Utilizing Contextual Information

In early speaker models, contextual information was typically limited to the size of the input window the model could process. For instance, d-vectors based on DNNs often use frame extension to expand the receptive field [27]. The x-vector [32] structure has a more limited receptive field, covering only tens of frames of contextual data. The TDNN [33] addresses this issue by incorporating dilated convolutions, which

provide greater flexibility in capturing long-range dependencies. Newer architectures, such as Transformers, can model long-range dependencies due to their self-attention mechanism. To more effectively utilize contextual information, three key strategies are frequently employed:

- *Multi-scale modeling*: Different subnets operate on various scales or feature bins, processing either raw inputs [81] or intermediate features [3, 76, 79, 82, 83, 84, 85, 86].
- *Cross-layer aggregation*: By aggregating features across layers with different temporal resolutions, models can better utilize contextual information [3, 47].
- *Explicit local and global information modeling*: For certain tasks, incorporating global contextual data from longer speech signals has proven beneficial. This approach can improve the model's robustness to noise and degradation in long-duration speech [60, 79, 87, 88, 89].

Automatic Feature Selection and Reweighting

The importance of features across different frequency bins varies, and their relevance for speaker modeling is not uniform [90]. Attentive pooling serves as an automatic reweighting mechanism along the time axis, while similar techniques for reweighting and feature selection can be applied along the frequency or channel axes [91, 92, 93, 94, 95].

2.2 The Evolution of Learning Paradigms

This section examines the significant paradigm shifts in machine learning over recent years. Traditionally, models were predominantly trained from scratch using supervised methods. However, recent advancements have introduced more sophisticated

approaches, including self-supervised learning [96, 97, 98, 99, 100] and the development of pre-trained large-scale speech models [101, 102, 103]. Furthermore, this section will also delve into the progress made in multi-modality and cross-modality learning frameworks.

2.2.1 From Supervised to Self-Supervised Learning

Supervised Learning Methods

In supervised learning, the training dataset comprises inputs paired with their corresponding expected outputs (labels). This methodology is effective for achieving precise modeling outcomes, provided the dataset is extensive and accurately annotated. In the realm of speaker representation learning, the loss functions employed in supervised training are generally categorized into two types: those based on softmax classification and those derived from metric learning.

Classification Based Objectives: The softmax loss function is widely used for training deep neural networks (DNNs) to discriminate between speakers. It is mathematically expressed as:

$$L_{\text{softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\mathbf{w}_{y_i}^T\boldsymbol{z}_i+b_{y_i}}}{\sum_{j=1}^{C}e^{\mathbf{w}_j^T\boldsymbol{z}_i+b_j}}, \tag{2.2}$$

where N denotes the batch size, and C represents the number of classes. Here, $\boldsymbol{z}_i \in \mathbb{R}^d$ is the i-th sample input to the projection layer, and y_i is its corresponding label index. The weight matrix and bias in the projection layer are denoted by $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_C] \in \mathbb{R}^{d\times C}$ and $\mathbf{b} = [b_1, \ldots, b_C]^\intercal \in \mathbb{R}^C$, respectively. While the softmax loss effectively penalizes misclassifications, it does not inherently optimize for intra-class compactness or inter-class separation. This limitation has spurred the development of enhanced variants like A-Softmax [104, 105], AM-Softmax [106, 107], and AAM-Softmax [38, 108], which incorporate margins in an angular space or in the cosine similarities to enhance discrimination between classes. A comprehensive

analysis of these margin-based softmax functions in speaker embedding learning is available in [38].

Metric Learning Based Objectives: In addition to classification-based objectives, metric learning-based loss functions have also been explored for speaker representation learning. For example, the Triplet [109, 110, 111, 112] and Quadruplet Losses [113, 114] aim to create an embedding space such that the positive instance (same class as the anchor) is moved closer to the anchor and the negative instance (different class from the anchor) is moved away from it. Meanwhile, Center Loss [37, 40, 115] aims to reduce intra-class variations and is often used alongside traditional softmax loss to balance intra-class compactness with inter-class separability.

Self-Supervised Methods

Acquiring large-scale datasets with speaker labels is often labor-intensive and may raise privacy concerns. Consequently, there is a growing need to uncover latent labels and inherent structures directly from the data, leading to the development of self-supervised training methodologies. Self-supervised learning can be broadly categorized into two paradigms: generative and discriminative. Generative methods facilitate model learning by reconstructing input data [116, 117]. However, in the domain of self-supervised speaker representation learning, the focus has predominantly been on discriminative approaches, which are highlighted below.

Contrastive Learning Based Method: Contrastive learning in self-supervised contexts resembles metric-based techniques, operating under two fundamental assumptions: 1) A single utterance or a brief consecutive interval contains only one speaker, and 2) Distinct utterances feature different speakers. Leveraging the first assumption, segments from the same speaker can be paired to form positive pairs, $(\boldsymbol{z}_i, \boldsymbol{z}_i^+)$. The second assumption allows for the identification of negative pairs, $(\boldsymbol{z}_i, \boldsymbol{z}_i^-)$, by sampling segments from different utterances. This framework enables the design of

contrastive loss functions aimed at minimizing the distance between the instances in positive pairs while maximizing the distance between the instances in negative pairs:

$$\mathcal{L}_{con} = \sum_{N^+} \mathrm{d}\left(\boldsymbol{z}_i, \boldsymbol{z}_i^+\right) - \sum_{N^-} \mathrm{d}\left(\boldsymbol{z}_i, \boldsymbol{z}_i^-\right), \tag{2.3}$$

where, $\mathrm{d}(\cdot,\cdot)$ represents any valid distance metric, and the loss function can take various forms beyond the basic formulation above. The overarching objective, however, remains consistent: to reduce the distance between the instances in positive pairs and increase it between the instances in negative pairs. For example, Jati *et al.* [118] employed the L_1 distance to measure segment similarity and utilized a binary classification loss to differentiate between positive and negative pairs. Ravanelli *et al.* [119] focused on maximizing mutual information (MI) for positive pairs while minimizing it for negative pairs. To enhance robustness against channel variations, Zhang *et al.* [120] introduced data augmentations to segments from the same utterance and incorporated an additional loss term to regulate positive pair distances. Xia *et al.* [121] further enriched negative pair diversity by maintaining a buffer of speaker embeddings from previous batches.

Non-Contrastive Learning Based Method: Despite the efficacy of contrastive learning in extracting speaker representations from unlabeled data, its reliance on the assumption that "different utterances contain different speakers" can lead to false negative pairs—instances where segments from different utterances belong to the same speaker. Han *et al.* [122] highlighted that nearly every batch of size 256 in the VoxCeleb2 dataset [123] contains at least one such false negative pair. To address this limitation, the 'self-**di**stillation with **no** labels (DINO)' strategy [124, 125, 126, 127] has been proposed. DINO employs two parallel networks: a student network and a teacher network. Positive pairs, derived from the same utterance, are fed into these networks, which map the inputs to high-dimensional distributions. The loss function is designed to minimize the divergence between the output distributions of the two networks:

$$\mathcal{L}_{DINO} = \mathrm{CrossEntropy}\left(student(\boldsymbol{z}_i), teacher(\boldsymbol{z}_i^+)\right). \tag{2.4}$$

During optimization, the student network is updated via backpropagation, while the teacher network is updated as a moving average of the student network. Jung *et al.* [125] applied DINO to a raw waveform-based system, achieving superior performance compared to prior self-supervised methods. Additionally, Heo *et al.* [126] demonstrated that progressively increasing the number of speakers during training further enhances performance. Chen *et al.* [127] provided a comprehensive analysis of DINO-based methods, examining the impact of data augmentation, speaker diversity, and the number of sampled segments on speaker representation learning.

2.2.2 Semi-Supervised Training

In practical applications, datasets often consist of a small portion of labeled data alongside a significantly larger pool of unlabeled data, defining a semi-supervised learning scenario. A common approach in this context involves initially pre-training a model using the labeled data. This model is subsequently utilized to generate pseudo-labels for the unlabeled data. Following this, a new model is trained on a combination of the original labeled data and the newly pseudo-labeled data [128, 129, 130]. The effectiveness of this method heavily relies on the accuracy of the pseudo-labels, which in turn is influenced by the quality and quantity of the initial labeled data used for pre-training. Additionally, innovative approaches by Inoue *et al.* [131] and Choi *et al.* [132] have integrated self-supervised and supervised learning objectives into a unified framework, enabling joint training on both labeled and unlabeled datasets. Typically, speaker recognition systems operate in two phases: the training phase for the speaker embedding extractor and the inference phase. Semi-supervised techniques are predominantly applied during the training phase. However, Chen *et al.* [133] introduced a graph-based label propagation method that utilizes both labeled enrollment data and additional unlabeled data during the inference phase, enhancing speaker recognition performance in scenarios such as household smart speakers.

2.2.3 Leveraging Large Pre-trained Models

Recent years have witnessed a surge in interest towards large-scale self-supervised speech pre-trained models [102, 134, 135, 136, 137]. These models are first pre-trained on extensive unlabeled datasets and then adapted to various speech-related downstream tasks. Yang *et al.* [138] developed a benchmark named SUPERB to systematically evaluate the performance of pre-trained models across different downstream tasks. Given the diversity in task paradigms, fine-tuning strategies are tailored accordingly. Fan *et al.* [139] pioneered the application of the wav2vec 2.0 model [135] to speaker verification and language identification tasks by augmenting the model with a pooling layer and a linear transformation to derive fixed-dimensional embeddings encapsulating speaker and language information. Vaessen *et al.* [140] further investigated the wav2vec 2.0 model's efficacy in speaker verification, experimenting with various pooling methods and loss functions, though the results did not surpass those of the ECAPA-TDNN network [3]. To bridge this performance gap, Chen *et al.* [141] integrated weighted representations from all layers of the pre-trained model directly into the ECAPA-TDNN framework, achieving notable improvements. While these methods typically involve fine-tuning the entire pre-trained model, which necessitates storing separate model parameters for each task, Peng *et al.* [142] proposed a parameter-efficient fine-tuning strategy that updates only lightweight adapters, yielding competitive results. Furthermore, Cai *et al.* [143] highlighted the advantages of using a Conformer model pre-trained specifically for Automatic Speech Recognition (ASR) tasks over generic pre-trained models for speaker verification, noting that ASR-specific pre-training mitigates overfitting in speaker recognition training.

Chapter 3

Mutual Information-Enhanced Contrastive Learning with Margin for Maximal Speaker Separability

3.1 Introduction

Speaker representation learning is crucial for speaker verification (SV). Its goal is to learn a feature embedding space characterized by 1) same-class compactness, ensuring that the embedding vectors of the same speaker are close; 2) different-class dispersion, where the embedding vectors belonging to different speakers are far apart. Recent years have witnessed significant advancements in this area, a result of the advancements in deep neural network (DNN) architectures [3, 32, 33], complex loss functions [39, 108, 144, 145], innovative pooling strategies [37, 146], and effective domain adaptation methods [147, 148, 149]. However, the models are still not sufficiently robust to noisy labels [150, 151] and are sensitive to input perturbation unless a notion of margin is introduced to their loss function [152, 153]. Research indicates that these shortcomings can reduce the models' generalization capabilities [154, 155, 156, 157].

Several methods have been developed to increase intra-class compactness and bolster inter-class separation in embedding spaces [158]. Wen *et al.* [115] proposed a regularization term to penalize the gaps between features and their corresponding centers. Building upon this idea, Ranjan *et al.* [159] and Wang *et al.* [160] suggested constraining the L_2 norm of the feature representations for the softmax loss so that they lie on a hypersphere with a fixed radius, and Liu *et al.* [161] proposed optimizing the cosine similarity between the feature representations and their class centroids. These adjustments result in well-separated classes in the representation space and reduce intra-class dispersion, resulting in larger gradients during training. Furthermore, Liu *et al.* [153] argued for an enlarged classification margin, emphasizing that a more challenging learning objective can stimulate the acquisition of more discriminative features. Similarly, Liu *et al.* [162] introduced an angular distance metric. This metric evaluates the dissimilarity of objects based on their geodesic distance within a hypersphere manifold and uses an angular margin to heighten the strictness of decisions.

Contrastive learning is increasingly gaining attention in the SV community [120, 163, 164, 165]. This approach creates positive pairs using augmented samples of an utterance from the same speaker. It considers different utterances and their augmented versions as being from distinct speakers, thus forming negative pairs. The overarching goal is to draw the embeddings of the positive pairs closer while distancing the embeddings of the negative pairs. Because the supervised information for one view comes from the other view, contrastive learning can leverage the shared information across views, but often overlooks nonshared task-relevant information. Shared information corresponds to speaker features relevant to the SV task, and the features are shared across different views of the utterances. For example, a waveform after noise contamination will still contain some information about the same speaker. Nonshared information, on the other hand, refers to speaker features that are specific to the test speakers but not shared between different views during contrastive learning; Wang

et al. [166] also theoretically proved that the nonshared information cannot be ignored; otherwise, the representation learned through contrastive learning may not be sufficient for the downstream tasks.

A speaker embedding network optimized by contrastive learning will also tend to ignore nonshared information because the network can never see the test speaker population during contrastive training. An intuitive approach to reinforce the nonshared information in the embeddings is to explicitly maximize the mutual information between low-level features and segment-level embeddings. This encourages the embeddings to capture speaker information preserved in lower-level representations [156, 167].

To facilitate maximal speaker separability, many investigations [168, 169, 170] have employed the NT-Xent loss, which is essentially a variant of the cross-entropy loss integrated with a softmax function. Nevertheless, recent findings [108, 106, 171] suggest that while the conventional softmax-based loss can effectively enlarge inter-class discrepancies, it is ineffective in minimizing intra-class variations. This phenomenon implies that the resulting features, although discriminative for closed-set classification, are inadequate for open-set speaker recognition. Moreover, the widespread contrastive strategies emphasize distinguishing between positive and negative pairs [172, 173, 121], with little attention to exploring optimization objectives.

To alleviate the above challenges, we designed a speaker verification framework to learn discriminative speaker representations using mutual information-enhanced contrastive learning with margin. The capability of speaker representation is enhanced by incorporating an additive angular margin into the supervised contrastive loss. Meanwhile, our framework increases the mutual information between the speaker representation and the first convolutional layer of the speaker encoder to capture more nonshared information.

This paper substantially extends our earlier work in [174, 175]. Firstly, the paper

empirically verifies that the minimal sufficient representation [166] is not sufficient for speaker verification because it misses the nonshared information across the augmented views. We maximize the mutual information between the low-level features and utterance-level embeddings to enhance the preservation of useful information. Our comprehensive experiments demonstrate that incorporating squeezed frame-level phonetic information into the embedding extractor consistently improves speaker verification performance. Secondly, the paper adds comprehensive analyses to investigate the impacts of the proposed method on speaker representation. The analyses include a detailed exploration of the angular margin's influence, an investigation into the impact of varying the number of positive samples, and an analysis of alignment and uniformity. Thirdly, we have extended our experimental evaluations from the VoxCeleb1 dataset to encompass the larger and more challenging VoxCeleb2 dataset. This expansion ensures a more comprehensive validation of our proposed method across different datasets. Fourthly, we investigate the behavior of our method under low-resource scenarios using a Cantonese dataset called CU-MARVEL. The dataset was originally developed for dementia detection, and we repurposed it for speaker verification research under low-resource conditions.

3.2 Methodology

We aim to develop a speaker representation network based on contrastive learning using labeled audio data. The embedding vectors should cluster together for similar speakers and be distant apart for dissimilar speakers. To this end, for each training batch, we apply data augmentation to create diverse samples for each utterance in the batch. Despite various augmentations, the embedding vectors of the same instance should remain consistent. Conversely, embeddings from different samples should be distinct.

As depicted in Fig. 3.1, an encoder network processes the spectrograms of both the

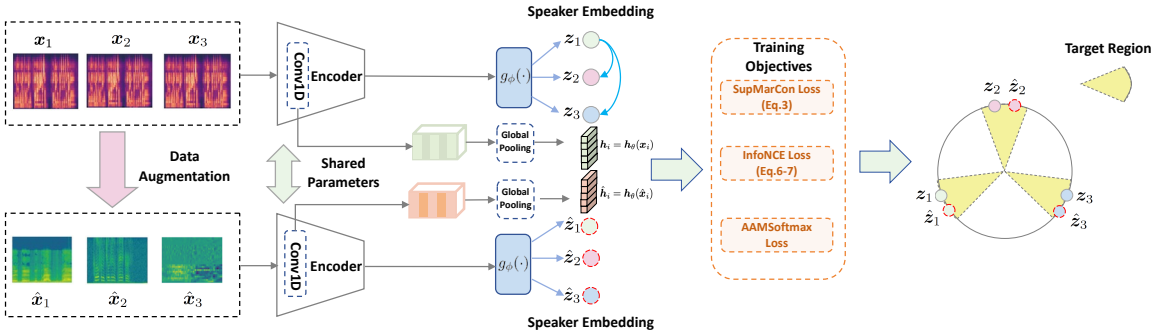


Figure 3.1: Our architecture uses additive angular margin for mutual information-enhanced supervised contrastive learning. The encoder transforms acoustic features (MFCC or FBank) into normalized embedding vectors. Invariance occurs for the embeddings (e.g., $\boldsymbol{z}_1$ and $\hat{\boldsymbol{z}}_1$) whose acoustic features ($\boldsymbol{x}_1$ and $\hat{\boldsymbol{x}}_1$) come from the same speaker. On the other hand, embeddings (e.g., $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$) whose acoustic features ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) belong to different speakers are far apart.

original instances and their augmented samples. This process yields a set of normalized embeddings. At the end of this process, we maximize the mutual information between the representation and the frame-level embedding from the encoder. We also compute the contrastive loss with an additive angular margin on the network's output.

3.2.1 Representation Learning Framework

Inspired by recent contrastive learning methods, our method aims to enhance representation learning. It maximizes the agreement across various augmented views of the same data via contrastive loss in the embedding space. As illustrated in Fig. 3.1, the framework consists of four pivotal components.

Data Augmentation For each input sample, we generated one or multiple random augmentations, denoted as $\hat{\boldsymbol{x}}_i = Augmentation(\boldsymbol{x}_i)$. Each augmentation provides a

unique data perspective and comprises some of the original sample's information. Following the Kaldi's recipe [176], we employed augmentation techniques such as adding noise, music, and chatter from the MUSAN dataset [177]. Additionally, we generated reverberation effects by convolving the original waveforms with room impulse responses (RIR) from the RIR dataset [178]. We also employ speed perturbation [179].

Encoder Network Our primary objective involves training an encoder network using a labeled audio dataset $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$. $h_\theta(\cdot)$ denotes the first convolutional layer in the speaker encoder followed by global pooling, transforms each input audio $\boldsymbol{x}_i$ into a low-dimensional vector $\boldsymbol{h}_i = h_\theta(\boldsymbol{x}_i) \in \mathbb{R}^{T \times d}$, where T is the number of frames and d represents the dimension. Both original and augmented samples are independently fed into the same encoder, resulting in two representation vectors $\boldsymbol{h}_i$ and $\hat{\boldsymbol{h}}_i$.

Projection Network The projection network, denoted as $g_\phi(\cdot)$ in Fig. 3.1, is a shallow network with one linear output layer responsible for transforming the encoder's output into a space where we apply the contrastive loss. We normalize the network's output to ensure the embedding vectors lie on a unit hypersphere. This normalization enables us to estimate distances in the projection space using inner products.

3.2.2 Recap of Supervised Contrastive Learning with Margin

Supervised Contrastive Learning

As shown in Fig. 3.2, we explore the supervised contrastive loss, where positive examples of a given class are contrasted with negative examples from different classes, utilizing the provided labels. We incorporated the original and augmented speaker

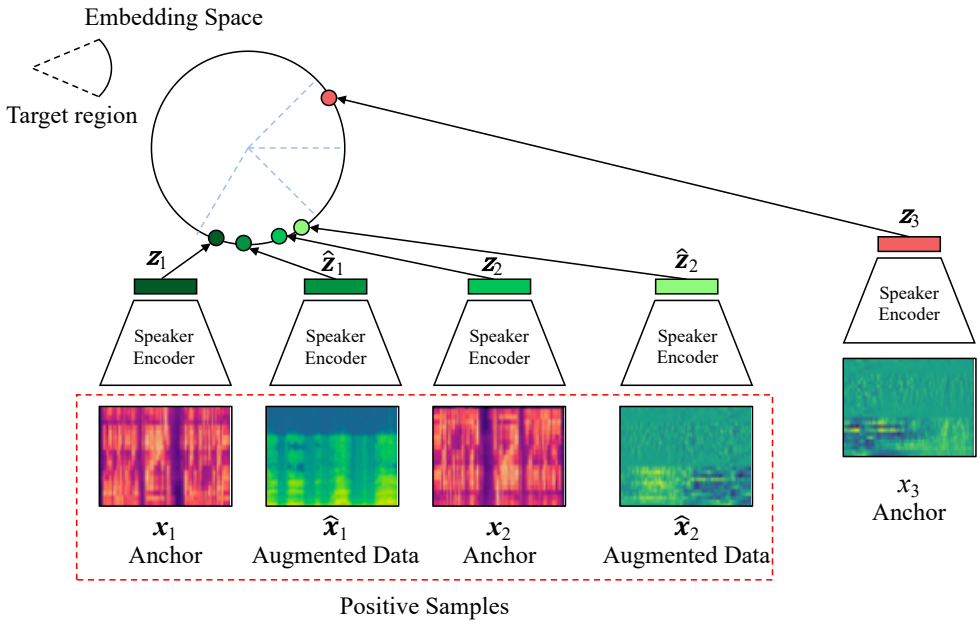


Figure 3.2: Our basic idea is illustrated by contrasting all samples from the same class (positives) against those from other classes (negatives) in a batch. By incorporating class label information, we create an embedding space where similar speakers stay close to each other while dissimilar ones are far apart. In this example, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ come from the same speaker, whereas $\boldsymbol{x}_3$ comes from another speaker.

embeddings into a supervised contrastive loss [174, 180]:

$$\mathcal{L}_{SupCon} = \sum_{i=1}^{N} \frac{-1}{|\mathcal{P}(i)|} \sum_{p \in \mathcal{P}(i)} \log \frac{\exp(sim(\boldsymbol{z}_i, \boldsymbol{z}_p)/\tau)}{\sum_{a \in \mathcal{A}(i)} \exp(sim(\boldsymbol{z}_i, \boldsymbol{z}_a)/\tau)}, \tag{3.1}$$

where $sim(\boldsymbol{z}_i, \boldsymbol{z}_p)$ is the cosine similarity. In Eq. 3.1, $\boldsymbol{z}_i$ is an anchor, $\boldsymbol{z}_a$ is a negative sample, $\mathcal{A}(i)$ comprises the indices of the negative samples with respect to $\boldsymbol{z}_i$, $\boldsymbol{z}_p$ is a positive sample with respect to $\boldsymbol{z}_i$, and $\mathcal{P}(i)$ contains the indices of positive samples in the augmented batch (original + augmentation). $\tau \in \mathcal{R}^+$ is a scalar temperature parameter.

Angular Margin Based Contrastive Learning

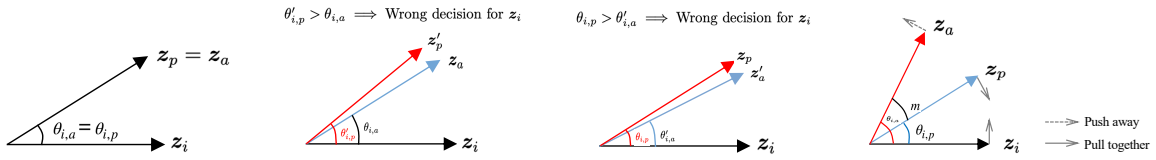
Although the training objective attempts to pull the representations of similar speakers closer together and push the representations of different speakers apart, these representations may not be sufficiently discriminative or robust against noise. Let us denote the cosine similarity as

$$\cos\theta_{i,p} = \frac{\boldsymbol{z}_i^\top \boldsymbol{z}_p}{\|\boldsymbol{z}_i\| \, \|\boldsymbol{z}_p\|}, \tag{3.2}$$

where $\theta_{i,p}$ is the angle between the embeddings $\boldsymbol{z}_i$ and $\boldsymbol{z}_p$. A similar formula applies to $\boldsymbol{z}_i$ and $\boldsymbol{z}_a$. The decision boundary of $\boldsymbol{z}_i$ for specific p and a is $\theta_{i,p} = \theta_{i,a}$, where p and a index to the positive and negative samples, respectively (Fig. 3.3a. A small perturbation of the embedding vectors around the decision boundary may result in an incorrect decision if no decision margin exists (Figs. 3.3b and 3.3c. To overcome this problem, we advocate adding an additive angular margin m to the decision boundary. We name the resulting objective as **sup**ervised **margin con**trastive (SupMarginCon) loss [175], which is formulated as:

$$\mathcal{L}_{SupMarginCon} = \sum_{i=1}^{N} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\cos\left(\theta_{i,p} + m\right)/\tau\right)}{\sum_{a \in A(i)} \exp\left(\cos\left(\theta_{i,a}\right)/\tau\right)}. \tag{3.3}$$

As shown in Fig. 3.3d, in this loss, the decision boundary of $\boldsymbol{z}_i$ for specific p and a is $\theta_{i,p} + m = \theta_{i,a}$. The minimization of Eq. 3.3 will push $\boldsymbol{z}_i$ further towards the area where $\theta_{i,p}$ decreases and $\theta_{i,a}$ increases. Therefore, adding a margin can increase the compactness of same-speaker representations and the divergences between the different-speaker representations. This aid improves alignment and uniformity – two quality measures fundamental to contrastive learning [181]. These metrics indicate how close positive-pair embeddings are to one another and how uniformly distributed the embeddings are. These properties make the SupMarginCon loss more discriminative than the conventional loss, such as the SupCon loss (Eq. 3.1).



(a) (b) (c) (d)

Figure 3.3: (3.3a) Without a decision margin, the decision boundary for $\boldsymbol{z}_i$ is $\theta_{i,p} = \theta_{i,a}$. (3.3b) – (3.3c) A small perturbation on $\boldsymbol{z}_p$ or $\boldsymbol{z}_a$ but in the wrong directions can lead to incorrect decisions. (3.3d) *SupMarginCon* incorporates an additive angular margin m, ensuring the decision boundary for $\boldsymbol{z}_i$ satisfies $\theta_{i,p} + m = \theta_{i,a}$ for specific positive and negative samples. With the tolerance m, both $\boldsymbol{z}_p$ and $\boldsymbol{z}_a$ can be subject to a larger perturbation without causing a wrong decision for $\boldsymbol{z}_i$.

3.2.3 Leveraging Nonshared Speaker Information

In contrastive learning, the augmented views provide supervision information for an anchor. For example, the input $\hat{\boldsymbol{x}}_i$ in Fig. 3.1 and Fig. 3.2 provides a supervision signal to $\boldsymbol{x}_i$ because they come from the same utterance. The signal plays a similar role as class labels in supervised learning [182]. The analyses in [166] suggest that in contrastive learning, the minimal sufficient representation falls short for downstream tasks due to the missing nonshared task-related information in the representations. Additionally, contrastive learning tends to produce a minimal sufficient representation (i.e., ignoring the nonshared information between multiple views of the same object), thus risking overfitting the shared information across views.

Unlike contrastive learning methods that use InfoNCE [183, 184] to maximize the similarity between positive samples [185, 186, 187] or the approaches that leverage mutual information to disentangle speaker embeddings from factors such as age and domain [188, 189, 190], our method employs InfoNCE [184] to increase the mutual information between low-level features and utterance-level embeddings. We extract additional nonshared information from $\boldsymbol{h}_i$ and $\hat{\boldsymbol{h}}_i$. $\boldsymbol{h}_i$ is the output of the first con-

volutional layer of the speaker encoder followed by global pooling, sharing the same dimensionality as $\boldsymbol{z}_i$. $\hat{\boldsymbol{h}}_i$ is the augmented version of $\boldsymbol{h}_i$. We maximize the mutual information $I(\boldsymbol{z}_i, \boldsymbol{h}_i)$ and $I(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{h}}_i)$ to enhance the speaker information in $\boldsymbol{z}_i$ and $\hat{\boldsymbol{z}}_i$. Given the symmetry between $\boldsymbol{h}_i$ and $\hat{\boldsymbol{h}}_i$, our objective is to maximize

$$I\left(\boldsymbol{z}_i, \boldsymbol{h}_i\right) + I\left(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{h}}_i\right). \tag{3.4}$$

For optimizing $I\left(\boldsymbol{z}_i, \boldsymbol{h}_i\right)$ and $I\left(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{h}}_i\right)$, we choose the InfoNCE as the lower bound estimates of mutual information. Concretely, the InfoNCE lower bound is [166]

$$\hat{I}_{NCE}(\boldsymbol{z}, \boldsymbol{h}) = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} \ln \frac{p\left(\boldsymbol{z}_i \mid \boldsymbol{h}_i\right)}{\frac{1}{N}\sum_{l=1}^{N} p\left(\boldsymbol{z}_l \mid \boldsymbol{h}_i\right)}\right], \tag{3.5}$$

where $\{\boldsymbol{z}_l\}_{l=1}^{N}$ are sampled from the conditional distribution $p(\boldsymbol{z}|\boldsymbol{h})$, with $\boldsymbol{h}_i$ drawn from the mini-batch, and N is the batch size.

To compute the InfoNCE [184] lower bound, we require a probabilistic model for $p(\boldsymbol{z}|\boldsymbol{h})$ from which N samples of $\boldsymbol{z}$, $\{\boldsymbol{z}_l\}_{l=1}^{N}$, are drawn. Following [166, 191], we employ the reparameterization trick during training. Specifically, we model $p(\boldsymbol{z}|\boldsymbol{h})$ as a Gaussian distribution $\mathcal{N}(\boldsymbol{z}; f_\theta(\boldsymbol{h}), \sigma^2\boldsymbol{I})$, where σ^2 is a pre-defined variance and $f_\theta(\boldsymbol{h})$ is a deterministic function implemented by a DNN parameterized by θ. Consequently, we have $\boldsymbol{z} = f_\theta(\boldsymbol{h}) + \sigma\boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Then, $\hat{I}_{NCE}$ is equivalent to

$$\hat{I}_{NCE}(\boldsymbol{z}, \boldsymbol{h}) = \mathbb{E}\left[-\frac{1}{N}\sum_{i=1}^{N} \ln \sum_{l=1}^{N} \exp\left(-\rho \left\|\boldsymbol{z}_l - f_\theta\left(\boldsymbol{h}_i\right)\right\|_2^2\right)\right], \tag{3.6}$$

where ρ is a scale factor. For estimating $I(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{h}}_i)$, the approach is the same. Therefore, the loss function for maximizing the mutual information is:

$$\mathcal{L}_{InfoNCE} = -\hat{I}_{NCE}(\boldsymbol{z}, \boldsymbol{h}) - \hat{I}_{NCE}(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{h}}_i). \tag{3.7}$$

3.2.4 Model Training

After finishing the contrastive loss minimization, the encoder's parameters are typically frozen before training a linear classification layer. However, we advocate achiev-

ing both contrastive and classification learning simultaneously. To this end, we introduce AAMSoftmax [108] to our classification task, which is optimized alongside the contrastive loss during training.

The SupMarginCon, incorporating InfoNCE loss [166, 184], can be added to the total loss as a regularization term. The combination can be implemented as follows:

$$\mathcal{L} = \mathcal{L}_{AAMSoftmax} + \mathcal{L}_{SupMarginCon} + \lambda \mathcal{L}_{InfoNCE}. \tag{3.8}$$

We aim to enhance the sufficiency of the information in the representations without compressing it. Additionally, we must avoid introducing excessive nonshared information to $\boldsymbol{z}$ from $\boldsymbol{h}$. We utilize a coefficient λ to control this.

3.2.5 Analysis and Discussion

Our proposed SupMarginCon loss function (Eq. 3.3) incorporated margin into SupCon (Eq. 3.1). The SupCon loss leads to an innovative contrastive approach that allows multiple positives for each anchor. Our proposed loss effectively leverages label information in contrastive learning and derives highly discriminative features essential for speaker verification. The proposed supervised contrastive learning-based framework has the following advantages:

- **Generalization to arbitrary positives**. Within a multiview batch, every anchor benefits from its augmented sample and other samples with the same label, contributing to the loss function's numerator. The supervised loss guides the encoder to generate representations that closely align with their respective classes, resulting in denser speaker clusters within the embedding space.
- **Enhanced contrastive capability with increased negatives:** As indicated by Eq. 3.3, the loss function has a sum over the negatives in the denominator. As a result, the capability to distinguish between noise and signal is enhanced when more negative samples are added.

- **Additive margin increases discriminative power:** The additive-angular-margin supervised contrastive loss improves speaker discrimination by increasing the decision margin in the angular space.

3.3 Experiments and Results

3.3.1 Implementation Details

We incorporated the proposed loss function into the models in the 3D-Speaker toolkit [192] and evaluated them on the VoxCeleb [193, 194], CN-Celeb [195, 196], and CU-MARVEL [197] datasets for speaker verification. We used various architectures in the 3D-Speaker toolkit and the ERes2NetV2 architecture [198] for the encoder. We utilized 80-dimensional Fbank vector as input features. Our experiments incorporated four types of data augmentations: room impulse responses, music, background noise, and babble noise. We employ speed perturbation [179] with scaling factors of 0.9 and 1.1. We use mini-batches of size 1024, and for each utterance in a mini-batch, we randomly extracted a 3-second segment. The Adam optimizer was used. The parameter m in Eq. 3.3 was set to 0.2 or 0.3. The margin and scale in AAM-Softmax were set to 0.3 and 32, respectively. The contrastive learning temperature τ was set to 0.07. For Eq. 3.6, $\sigma = 0.1$ and $\rho = 0.05$.

3.3.2 Results and Analysis

Table 3.1 presents a comprehensive evaluation of various speaker encoders trained on VoxCeleb2 and tested on VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H. Multiple architectures including Res2Net, ResNet34, ECAPA-TDNN, ERes2Net, CAM++, and ERes2NetV2 were evaluated across several loss functions: Cross-Entropy, AM-Softmax, AAM-Softmax, and our proposed loss function combining AAM-Softmax,

supervised contrastive learning with margin, and mutual information enhancement (Eq. 3.8).

Across all architectures and test sets, our proposed method consistently achieved superior performance. Specifically, the ERes2NetV2 architecture with our proposed loss achieved the best overall results, obtaining EERs of 0.53%, 0.66%, and 1.21%, and minDCFs of 0.049, 0.071, and 0.121 on VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H respectively. This represents notable improvements over the baseline AAM-Softmax loss with relative reductions of approximately 14.5%, 14.3%, and 17.1% in EER, highlighting the effectiveness of our approach.

The results further illustrate that margin-based losses (AM-Softmax, AAM-Softmax, and our loss) significantly outperform traditional cross-entropy across all tested architectures and datasets, reinforcing the efficacy of margin-based constraints in speaker verification. Moreover, our approach consistently outperforms standard AAM-Softmax, demonstrating the complementary advantages provided by supervised contrastive learning and mutual information enhancement, particularly in the more challenging test set VoxCeleb1-H.

Our proposed method demonstrates consistent superiority on the CN-Celeb evaluation set, achieving the best results across all speaker encoders and loss functions. As shown in Table 3.2, the ERes2NetV2 architecture combined with our loss function achieves an EER of 5.73% and a minDCF of 0.341, outperforming the second-best AAM-Softmax baseline (6.14% EER, 0.370 minDCF). This trend holds across all architectures, with our method consistently reducing EER and minDCF over traditional loss functions. The improved performance on CN-Celeb, which includes diverse and challenging real-world scenarios, further underscores the generalization capability of our method. These results align with the findings on VoxCeleb, confirming the effectiveness of our hybrid optimization strategy across different datasets and evaluation protocols.

3.3.3 Comparing with Margin-based Contrastive-based Loss

To verify the effectiveness of our proposed loss function, we compare it against several well-known contrastive learning and margin-based methods, including AMC-loss [201], triplet loss [148, 202], angular prototypical loss (Ang-Prototy) [203], and CBRW-BCE [204]. AMC-Loss [201] combines traditional cross-entropy loss with an angular margin, explicitly minimizing geodesic distances within classes and maximizing inter-class angular separations. Triplet loss [202] is closely related to supervised contrastive learning, representing a special case of contrastive loss that uses exactly one positive and one negative sample per anchor. Angular prototypical loss [203] does not require explicit speaker identities for each utterance; instead, positive pairs are sampled from within the same utterance and negative pairs from different utterances. CBRW-BCE [204] leverages a bipartite ranking method to mitigate the imbalance of trials, integrating curriculum learning that gradually selects harder negative samples, thus improving training stability and model performance.

Table 3.3 summarizes the comparison results. When ECAPA-TDNN was used as the speaker encoder, the proposed loss achieved an EER of 0.74% and minDCF of 0.096, significantly outperforming AMC-Loss (2.54% EER), triplet loss (2.30% EER), angular prototypical loss (1.19% EER), and CBRW-BCE (1.10% EER). These results demonstrate the clear advantage of our method in terms of speaker discriminative capability.

Table 3.3 also shows that under the AAM-Softmax loss, the speaker encoder CAM++ [36] and ECAPA++ [205] achieve a similar performance (0.66% and 0.65% EER, respectively) but outperform IM-ECAPA-SimAM [206] and NeXt-TDNN [84] (0.79% EER). Notably, when trained with our proposed loss, CAM++ [36] can achieve an even better performance (0.59% EER, 0.076 minDCF), surpassing both ECAPA++ [205] and NeXt-TDNN [84], despite these two encoders are more advanced. This observation indicates that our proposed loss function can significantly enhance the performance

of encoders with simpler architectures, demonstrating its robustness and effectiveness across different speaker encoders.

3.3.4 Ablation Study

To further understand the contributions of each component in our proposed loss function, we conducted an ablation study using the ERes2NetV2 architecture, shown in Table 6.2. Specifically, we evaluated the individual and combined effects of MI (Eq. 3.6), SupCon (Eq. 3.1), and SupMarginCon (Eq. 3.3).

Table 6.2 shows that incorporating MI alone with AAM-Softmax provides a slight performance improvement, reducing the EER from 0.65% to 0.63% on VoxCeleb1 and from 6.14% to 6.08% on CN-Celeb1. The addition of supervised contrastive learning (SupCon) significantly enhanced performance, further decreasing EER to 0.57% on VoxCeleb1 and 5.90% on CN-Celeb1. When introducing the margin into supervised contrastive learning (SupMarginCon), additional gains were observed, reducing the EER to 0.54% and 5.80%, respectively. Finally, combining all three components (AAM-Softmax, SupMarginCon, and MI) results in the best overall performance, achieving an EER of 0.53% on VoxCeleb1 and 5.73% on CN-Celeb1, demonstrating the effectiveness of each component and their synergistic combination.

3.3.5 Effect of Maximizing Mutual Information

We selected three classic contrastive learning models—SimCLR [96], MOCO [97], and SupCon [180] (Eq. 3.1)—as our baselines to evaluate the impact of increasing the mutual information between frame-level features and speaker embeddings. The results on VoxCeleb1-O and CN-Celeb1-test are displayed in Table 3.5. Maximizing mutual information between the frame-level output and the utterance-level representation introduces nonshared information, enhancing performance notably in self-supervised

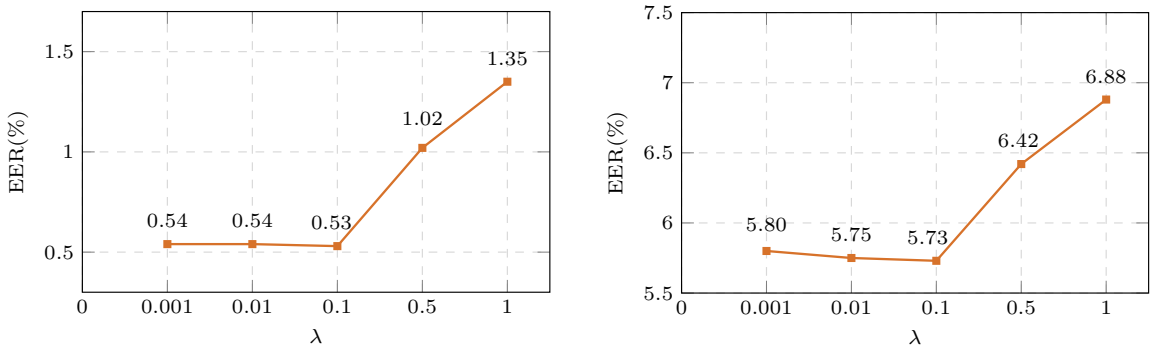
learning. This suggests that the shared information between views is insufficient for speaker verification, where enhanced mutual information leads to substantial improvements. Previous best practices [157] have shown that using the output of the first convolutional layer of the speaker encoder followed by global pooling as $\boldsymbol{h}$ achieves the optimal results. We follow this recipe in our approach. Furthermore, its effectiveness across different contrastive learning models indicates that our findings are broadly applicable.

3.3.6 Effect of Increasing the Role of Mutual Information

Accurately quantifying mutual information between high-dimensional variables is challenging and frequently results in imprecise estimations. We hypothesize that the hyper-parameter λ plays an important role in regularizing the nonshared information in the embedding. Specifically, a larger λ will introduce more nonshared information across views, enriching speaker information in the embeddings. To test this hypothesis, we varied λ and set it to 0.001, 0.01, 0.1, 0.5, and 1.0 and assessed the performance of the proposed loss function (Eq. 3.8) using ERes2NetV2 as the speaker encoder. Fig. 3.4 shows the EER against different values of λ. We observe a non-monotonic V-shape in EER with varying λ, suggesting that increasing mutual information consistently enhances performance in speaker verification, but excessively increasing mutual information could introduce noise along with useful information.

3.3.7 Effect of Angular Margin

The angular margin m in the SupMarginCon (Eq. 3.3) loss function affects the model's ability to discriminate. To explore this effect further, we systematically varied m from 0 to 0.5 degree in increments of 0.1 degrees. When the margin m is set to 0, SupMarginCon naturally degenerates into the SupCon (Eq. 3.1). The results are shown in Fig. 3.5.



(a) Results are based on the VoxCeleb2-dev training and VoxCeleb1-O test sets.

(b) Results are based on the CN-Celeb1&2 training and CN-Celeb1 test sets.

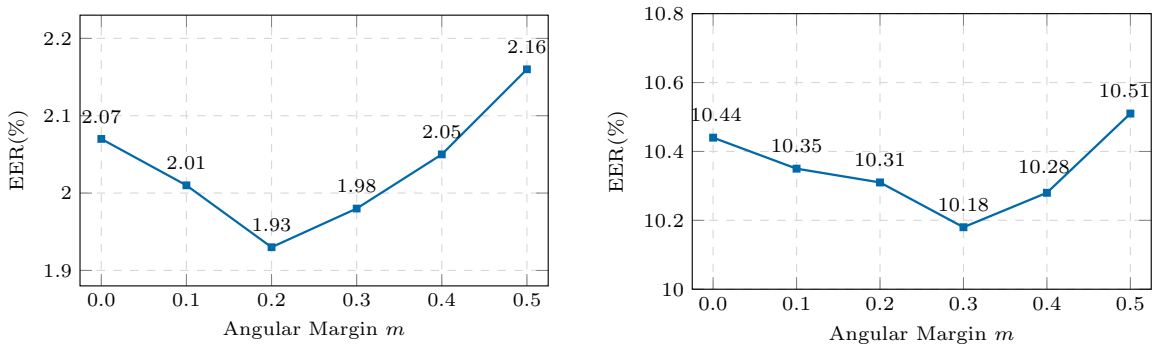
Figure 3.4: EER with varying hyper-parameter λ in Eq. 3.8.

As shown in Fig. 3.5a, the model achieves optimal performance on VoxCeleb1-O when $m = 0.2$. Fig. 3.5b shows that the best performance (EER = 10.18%) on CN-Celeb is achieved when $m = 0.3$. Any deviation from these optimal values results in a performance drop. This observation aligns with the common intuition, i.e., excessively small m causes the contrastive objective to lose discriminative power, while unnecessarily large m makes training difficult, causing suboptimal embedding networks.

3.3.8 Effect of Contrastive Learning

We further validated the capability of the proposed contrastive learning paradigm by visualizing the embedding of 20 speakers in the VoxCeleb1. Each speaker has 100 utterances. We used t-SNE to project the high-dimensional embedding vectors to a 2D space.

Fig. 3.6a shows the embeddings obtained from AAMSoftmax, while Figs. 3.6b, 3.6c, and 3.6d provide visualizations of SupCon, SupMarginCon, and our loss, respectively. Figs. 3.6c and 3.6d show that incorporating the margin makes the speaker clusters more compact. Our loss combines AAMSoftmax, SupMarginCon, and MI, leverag-



(a) Results are based on VoxCeleb2-dev for training and VoxCeleb1-O for evaluation.

(b) Results are based on CN-Celeb1&2 for training and CN-Celeb1-test for evaluation.

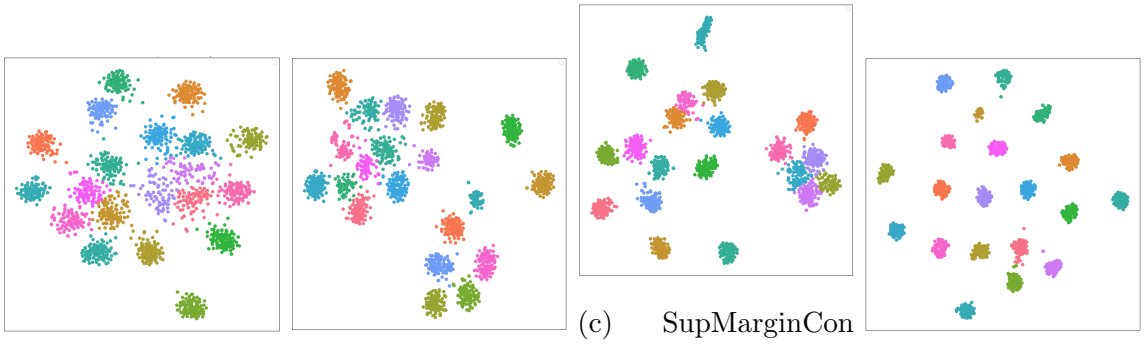
Figure 3.5: Effect of the angular margin m in the SupMarginCon (Eq. 3.3) loss on EER.

ing both the enhanced classification capability of AAMSoftmax and the distinctive feature separation ability of SupMarginCon. This combination not only results in tighter speaker clusters but also leads to clear boundaries between different speakers. This enhanced clustering confirms the effectiveness of our model in distinguishing between different speakers and further validates the efficacy of our proposed contrastive learning approach.

3.3.9 Effect of Number of Positives

We investigated the effect of positive samples by incrementally increasing their number up to k per anchor. It is important to ensure that these samples do not appear in the denominator of the loss function in Eq. 3.3. This exclusion ensures that the model does not consider them negative.

We utilized the ECAPA-TDNN encoder and conducted experiments on the VoxCeleb1 dataset with a batch size of 1024. We trained the network for 300 epochs. Fig. 3.7 shows the results. A noticeable trend emerges from the result: introducing more positives consistently enhances the model's performance. Therefore, we conclude that



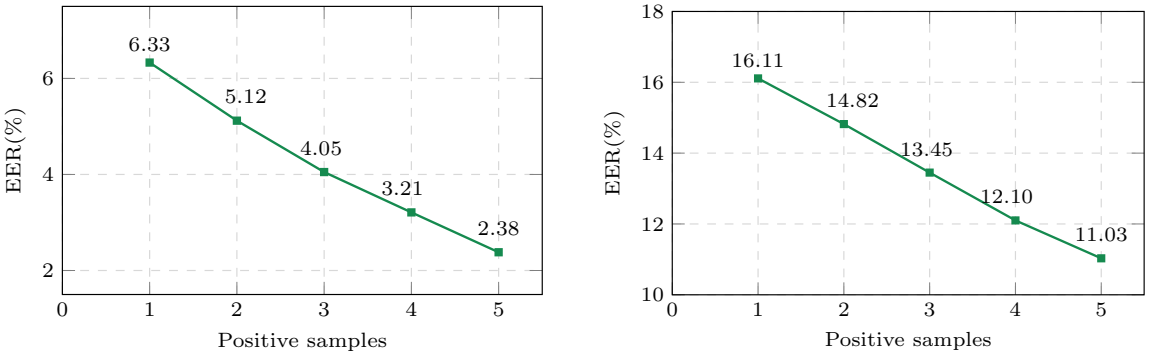
(a) AAMSoftmax (b) SupCon (Eq. 3.1) (c) SupMarginCon (Eq. 3.3) (d) Our loss (Eq. 3.8)

Figure 3.6: t-SNE plots of the embeddings of 20 speakers in VoxCeleb1. Each color represents one speaker. The graphs show the speaker clustering effects produced by four different loss functions using the ECAPA-TDNN: (3.6a) AAMSoftmax (1st term of Eq. 3.8), (3.6b) SupCon (Eq. 3.1), (3.6c) SupMarginCon (2nd term of Eq. 3.8), and (3.6d) our proposed loss (Eq. 3.8).

more positive samples encourage the encoder to give closely aligned representations, resulting in compact speaker clusters in the embedding space.

3.3.10 Sensitivity of Temperature Parameter

The loss function in contrastive learning is typically constructed from a softmax function of feature similarities to contrast between the positive and negative pairs, with the similarity scaled by a temperature parameter τ (see Eq. 3.1). We observe that contrastive loss, a hardness-aware function, optimizes hard negative samples by penalizing them based on their difficulty. The temperature parameter controls the severity of penalties applied to the hard negatives. Specifically, a small temperature in contrastive loss results in stronger penalties on the hardest negative samples, promoting greater separation in their local structure and a more uniform embedding distribution. Conversely, with a large temperature, contrastive loss becomes less responsive to hard negative samples, diminishing its hardness-aware characteristics



(a) Results were based on VoxCeleb2 for training and VoxCeleb1-O for evaluation.

(b) Results were based on CN-Celeb1&2 for training and CN-Celeb1-test for evaluation.

Figure 3.7: EER versus the maximum number of positives in $\mathcal{P}(i)$. Adding more positives reduces EER.

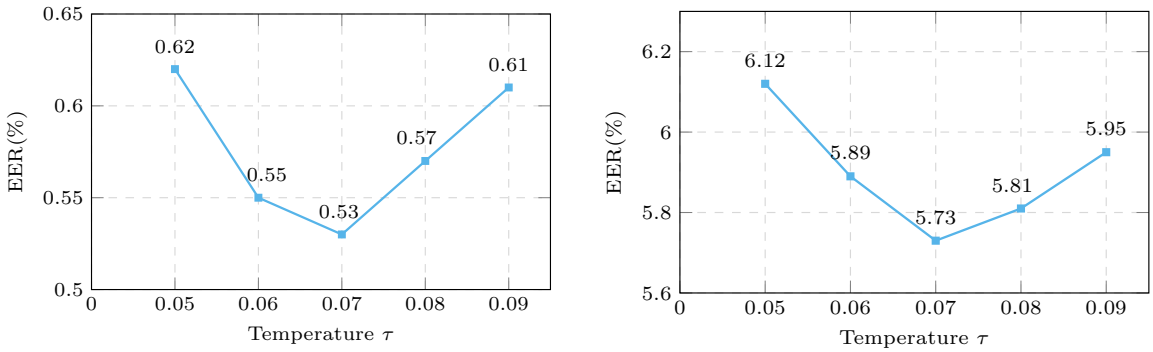
when the temperature approaches $+\infty$. The hardness-aware characteristic significantly contributes to the efficacy of softmax-based contrastive loss. As demonstrated in Fig. 3.8, a temperature of 0.07 leads to competitive SV performances.

3.3.11 Alignment and Uniformity Analysis

Alignment and uniformity are two closely related properties in contrastive learning and serve as valuable metrics for evaluating the quality of representations. Specifically, alignment refers to how an encoder generates similar representations for similar samples. It can be quantitatively defined by calculating the expected distance between the embeddings of positive pairs:

$$\ell_{\text{align}} = \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{x}_p\sim p_{\text{pos}}} \|f(\boldsymbol{x}) - f(\boldsymbol{x}_p)\|_2^2, \tag{3.9}$$

where p_{pos} denotes the distribution of positive samples. Uniformity refers to how uniform the distribution of the embedding is, which helps preserve information. It is



(a) Results are based on VoxCeleb2-dev for training and VoxCeleb1-test for evaluation.

(b) Results are based on CN-Celeb1&2 for training and CN-Celeb1-test for evaluation.

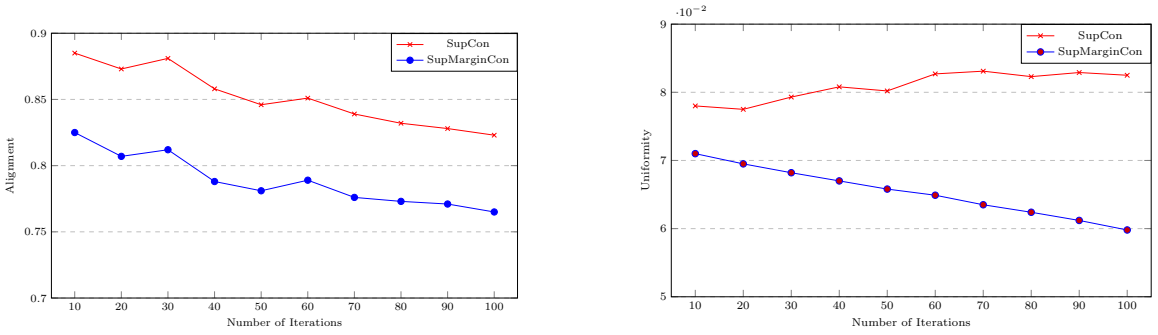
Figure 3.8: EER versus the temperature parameter τ in the loss function in Eq. 3.3. The results are based on an ERes2NetV2 speaker encoder optimized by minimizing the total loss in Eq. 3.8 with $\lambda = 0.1$.

defined as

$$\ell_{\text{uniform}} = \log \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y} \overset{i.i.d}{\sim} p_{\text{data}}} e^{-2\|f(\boldsymbol{x})-f(\boldsymbol{y})\|_2^2}, \tag{3.10}$$

where p_{data} represents the distribution of all data.

To evaluate the alignment and uniformity of our method, we conducted an assessment using the CN-Celeb dataset. For every 10 iterations, we computed the alignment and uniformity of SupMarginCon and compared them against the alignment and uniformity of the original supervised contrastive learning. The results are presented in Fig. 3.9. SupMarginCon consistently enhances alignment and uniformity throughout the training process compared to supervised contrastive learning. These findings validate the intuition behind our approach and indicate that incorporating margin can significantly enhance the quality of speaker representations.



(a) l_{align} (b) $l_{\text{Uniformity}}$

Figure 3.9: The alignment and uniformity of SupCon (Eq. 3.1) and SupMarginCon (Eq. 3.3). (3.9a) l_{align} measures the alignment between positive pairs. (3.9b) $l_{\text{Uniformity}}$ measures the uniformity of the embedding distribution. For both metrics, a lower value indicates better performance.

3.3.12 Low-resource Scenario

We investigated the behavior of our loss under a low-resource scenario. To this end, we applied our proposed loss on an ERes2NetV encoder using the CU-MARVEL dataset, a Cantonese dataset for dementia detection [197]. It comprises 280 speakers with the majority of audio recordings shorter than 2 seconds. We repurposed the dataset for speaker verification, and the statistics of CU-MARVEL are shown in Table 3.6.

Table 3.7 presents the performance and the conventional methods on CU-MARVEL version 0915. When using the ERes2NetV2 encoder with Fbank features, applying data augmentation and utilizing the AAMSoftmax loss function results in an EER of 6.80% and a minDCF of 0.74. When we employed the SupCon loss function, we observed a significant performance improvement, achieving an EER of 5.95% and a minDCF of 0.72. When we sequentially add margin and mutual information, performance improves with each addition. We noted that under low-resource conditions, contrastive learning loss outperforms classification-based loss. We attribute this performance gain to the training objectives. Speaker verification is an open-set task

where a limited number of utterances are insufficient to train a robust classifier for unseen samples. However, the goal of contrastive learning is to enhance discriminative ability, which allows it to perform better on unseen test datasets under insufficient training data scenarios.

3.4 Conclusions

We introduce a supervised contrastive learning framework designed to learn discriminative speaker representations. Our approach incorporates mutual information into contrastive learning, enhancing speaker-related information. We use an angular margin to improve the discriminative power of the contrastive learning loss. These enhancements improve speaker representation learning. The experimental results from CN-Celeb, VoxCeleb, and CU-MARVEL show that both techniques significantly enhance the performance of the speaker encoder in contrastive learning. On the low-resource CU-MARVEL dataset, our contrastive learning method even outperforms the classification loss.

Table 3.1: Performance comparison of speaker encoders trained on VoxCeleb2 and evaluated on VoxCeleb1 test sets (VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H), using various loss functions, including Cross-Entropy, AM-Softmax, AAM-Softmax, and our proposed one (AAM-Softmax + supervised contrastive learning with margin + mutual information enhancement). The best results are highlighted in **bold**.

Speaker Encoder	Loss function	VoxCeleb1-O		VoxCeleb1-E		VoxCeleb1-H	
		EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF
Res2Net [199]	Cross-Entropy	4.12	0.453	4.31	0.461	5.52	0.552
	AM-Softmax	1.63	0.181	1.52	0.161	2.61	0.301
	AAM-Softmax	1.56	0.151	1.42	0.149	2.48	0.231
	Ours (Eq. 3.8)	1.41	0.132	1.26	0.136	2.21	0.212
ResNet34 [52]	Cross-Entropy	3.95	0.431	4.16	0.441	5.21	0.531
	AM-Softmax	1.26	0.121	1.26	0.121	2.11	0.201
	AAM-Softmax	1.05	0.108	1.12	0.117	1.99	0.193
	Ours (Eq. 3.8)	0.93	0.099	0.99	0.106	1.79	0.176
ECAPA-TDNN [3]	Cross-Entropy	3.71	0.411	3.86	0.421	5.91	0.511
	AM-Softmax	1.06	0.121	1.16	0.121	2.01	0.201
	AAM-Softmax	0.87	0.117	0.98	0.113	1.91	0.194
	Ours (Eq. 3.8)	0.74	0.096	0.83	0.103	1.63	0.166
ERes2Net [200]	Cross-Entropy	4.51	0.391	3.66	0.401	4.61	0.491
	AM-Softmax	1.01	0.101	1.06	0.105	1.81	0.192
	AAM-Softmax	0.85	0.089	0.97	0.103	1.79	0.176
	Ours (Eq. 3.8)	0.69	0.083	0.76	0.091	1.49	0.149
CAM++ [36]	Cross-Entropy	3.31	0.371	3.46	0.381	4.31	0.471
	AM-Softmax	0.71	0.095	0.91	0.101	1.61	0.181
	AAM-Softmax	0.66	0.087	0.82	0.095	1.59	0.164
	Ours (Eq. 3.8)	0.59	0.076	0.71	0.086	1.36	0.136
ERes2NetV2 [198]	Cross-Entropy	3.16	0.351	3.31	0.361	4.01	0.451
	AM-Softmax	0.68	0.076	0.81	0.092	1.58	0.161
	AAM-Softmax	0.62	0.055	0.77	0.083	1.46	0.144
	Ours (Eq. 3.8)	**0.53**	**0.049**	**0.66**	**0.071**	**1.21**	**0.121**

Table 3.2: The performance of the proposed and conventional loss functions on the CN-Celeb evaluation set using different speaker encoders. Each metric's best result is in bold.

Speaker Encoder	Loss function	CN-Celeb1-Test	
		EER (%)	minDCF
Res2Net [199]	Cross-Entropy	12.12	0.682
	AM-Softmax	8.35	0.523
	AAM-Softmax	7.96	0.452
	Ours (Eq. 3.8)	7.21	0.423
ResNet34 [52]	Cross-Entropy	11.95	0.663
	AM-Softmax	7.39	0.507
	AAM-Softmax	6.92	0.421
	Ours (Eq. 3.8)	6.48	0.395
ECAPA-TDNN [3]	Cross-Entropy	11.63	0.651
	AM-Softmax	8.08	0.442
	AAM-Softmax	8.01	0.445
	Ours (Eq. 3.8)	7.45	0.413
ERes2Net [200]	Cross-Entropy	10.84	0.623
	AM-Softmax	7.11	0.468
	AAM-Softmax	6.69	0.388
	Ours (Eq. 3.8)	5.98	0.348
CAM++ [36]	Cross-Entropy	10.51	0.602
	AM-Softmax	6.93	0.451
	AAM-Softmax	6.78	0.393
	Ours (Eq. 3.8)	5.95	0.353
ERes2NetV2 [198]	Cross-Entropy	10.22	0.585
	AM-Softmax	6.65	0.433
	AAM-Softmax	6.14	0.370
	Ours (Eq. 3.8)	**5.73**	**0.341**

Table 3.3: Performance comparison of the proposed loss function and existing margin-based and contrastive learning methods on VoxCeleb1-O.

Speaker Encoder	Loss Function	VoxCeleb1-O	
		EER(%)	minDCF
ECAPA-TDNN [3]	AMC-Loss [201]	2.54	0.195
	Triplet (semi-hard) [111]	2.30	0.185
	Ang-Prototy Loss [203]	1.19	0.113
	CBRW-BCE [204]	1.10	0.088
	Ours	**0.74**	**0.096**
NeXt-TDNN [84]	AAMSoftmax [108]	0.79	0.086
IM-ECAPA-SimAM [206]	AAMSoftmax [108]	0.79	0.064
ECAPA++ [205]	AAMSoftmax [108]	0.65	0.079
CAM++ [36]	AAMSoftmax [108]	0.66	0.087
CAM++ [36]	Ours	**0.59**	**0.076**

Table 3.4: Ablation study of the proposed loss components on ERes2NetV2.

Loss Components	VoxCeleb1-test		CN-Celeb1-test	
	EER (%)	minDCF	EER (%)	minDCF
AAM-Softmax	0.62	0.055	6.14	0.370
AAM-Softmax + MI	0.60	0.055	6.08	0.368
AAM-Softmax + SupCon	0.57	0.053	5.90	0.350
AAM-Softmax + SupMarginCon	0.54	0.050	5.80	0.345
AAM-Softmax + SupMarginCon + MI (Ours)	**0.53**	**0.049**	**5.73**	**0.341**

Table 3.5: Effect of increasing mutual information on contrastive learning. Results are based on CN-Celeb1&2 or VoxCeleb2-dev for training and CN-Celeb1-test or VoxCeleb1-test for evaluation.

Model	VoxCeleb1-O		CN-Celeb1-test	
	EER(%)	minDCF	EER(%)	minDCF
SimCLR	6.78	0.548	15.88	0.677
SimCLR+MI	**6.63**	**0.523**	**15.47**	**0.652**
MOCO	7.46	0.627	16.17	0.706
MOCO+MI	**7.34**	**0.608**	**15.89**	**0.692**
SupCon	2.07	0.238	10.44	0.597
SupCon+MI	**2.02**	**0.231**	**10.26**	**0.583**

Table 3.6: Statistics of CU-MARVEL.

Data Split	# of Speakers	# of Utterances	# of Trials
Train	280	206,034	N/A
Test	53	43,319	400,000

Table 3.7: The performance of the proposed loss and conventional losses on CU-MARVEL. Fbank features were used as the input to an ERes2NetV2 speaker encoder.

Loss Function	EER(%)	minDCF
AAMSoftmax	6.80	0.74
SupCon	5.95	0.72
SupMarginCon	5.72	0.71
SupMarginCon + MI	5.63	0.70
SupMarginCon + MI + AAMSoftmax (Ours)	**4.98**	**0.67**

Chapter 4

Parameter-efficient Fine-tuning of Speaker-Aware Dynamic Prompts for Speaker Verification

4.1 Introduction

Applying pre-trained models (PTMs) to speaker verification (SV) is a promising direction. The advantage of this approach is the ability to leverage knowledge from large-scale speech datasets, enhancing the robustness of downstream SV tasks. However, full fine-tuning of PTMs is challenging as their size grows from hundreds of millions to billions of parameters. For instance, Whisper [103] contains 1.55 billion parameters.

Recently, researchers have proposed parameter-efficient transfer learning (PETL) methods to tune PTMs using lightweight trainable parameters while keeping most pre-trained parameters frozen [207, 208, 209, 210, 211]. Prompt tuning involves concatenating trainable prompt tokens with Transformer block's inputs to facilitate few-shot learning in speech recognition [212, 213], text-to-speech [214], and other speech

processing tasks [1, 210, 215]. In particular, soft prompts can be appended to the Transformer encoders' input to incorporate additional soft constraints and biases, thereby effectively adapting a Transformer model to a new domain without extensive re-training or fine-tuning.

Recent studies have shown that directly updating trainable tokens may lead to unstable optimization and performance degradation [208, 216]. To tackle these challenges, a prompt encoder, such as a multilayer perceptron (MLP), is employed to reparameterize the token embeddings [208, 217]. In speaker verification, static prompts often lead to poor generalization to unseen speakers and reduced improvements even with additional prompts or tunable parameters. The problem is that most methods associate the prompts with training speakers explicitly, leading to the static prompts overfitting these speakers. Because each utterance has its own prompts, they tend to be associated with the utterance rather than the speaker of the utterance, resulting in poor generalization to unseen speakers. Consequently, increasing the parameters in the prompt encoder does not guarantee the capture of more speaker information, resulting in minimal improvements due to prompt underutilization.

We propose constructing speaker-trait-aware prompts to enhance the generalization to unseen speakers and effectively utilize the prompt embeddings. The speaker-trait-aware prompts have three advantages. First, recent research has shown that allowing the prompts to learn the context from multiple instances can improve generalization to unseen answers in visual question answering [218] and unseen classes in image recognition [219] and reduce catastrophic forgetting in continual learning [220, 221]. Thus, allowing each prompt to learn from multiple instances can enhance prompt generalization. Second, the speaker-trait-aware prompts can capture the complex relationships between speakers, resulting in well-utilized prompt embeddings. Third, by putting the well-utilized prompts into a prompt pool, we can improve performance with fewer parameters, thereby enhancing the parameter efficiency of prompt tuning.

To create a prompt pool, we allow the learning of a set of dynamic prompts that guide

a pre-trained Transformer to extract frame-level features that can generalize to unseen speakers. Specifically, prompts in the pool are organized in dynamic key-prompt pairs, where the dynamic keys are the means of the Transformer encoders' inputs and the dynamic prompts are updated by minimizing the cross-entropy speaker loss. A dynamic selection strategy is developed to find the appropriate prompts for each training utterance. The prompt pool ensures that the shared prompts can encode transferable knowledge across speakers and that the individual prompts can capture speaker-specific knowledge. The selected prompts are prepended to the Transformer encoders' inputs, thus implicitly providing speaker-trait instructions to the pre-trained model.

In summary, this work makes the following contributions:

- We leverage a speaker prompt pool to adapt PTMs. This new mechanism tackles prompt tuning challenges by introducing a prompt pool memory space, which serves as parameterized instructions for pre-trained models to learn speaker identity.

- Our query mechanism dynamically selects prompts relevant to speaker traits, thereby effectively distinguishing speaker identity. This selection strategy minimizes the interference from knowledge unrelated to speaker identity mixed into speaker representations during optimization.

4.2 Methodology

Fig. 4.1 illustrates the proposed prompt tuning method. This section explains the dynamic prompt selection and updating processes and how the prompts can be used for adapting a pre-trained Transformer.

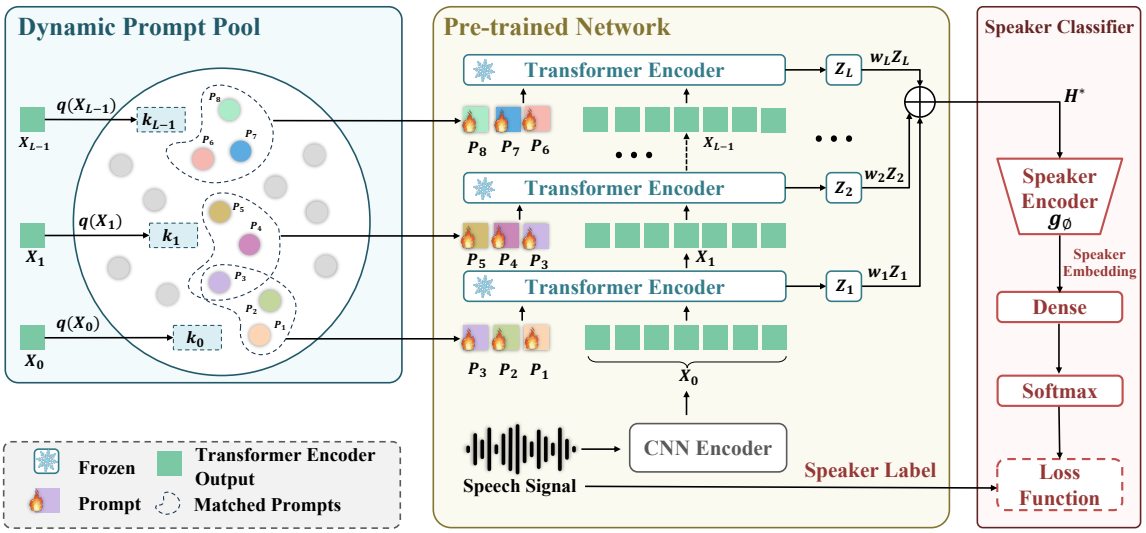


Figure 4.1: Illustration of the dynamic prompt selection and updating processes. First, we select a subset of prompts from a key-prompt paired pool using a query mechanism. Then, the selected prompts are prepended to the input vectors of each Transformer encoder. Finally, the extended vectors are fed into the encoders, and the selected prompts in the prompt pool are optimized by minimizing the AAM-Softmax loss. The objective is to select and update the prompts to guide the PTM's predictions.

4.2.1 Dynamic Prompt Pool

Because the speakers during inferencing are usually different from those for training the speaker embedding network, letting the utterance-dependent prompts be optimized for their respective speakers is not flexible. The limitation is that these prompts are fixed after training and will be used as input to the respective Transformer layers during inferencing. However, these utterance-dependent prompts will limit the model's ability to generalize from seen to unseen speakers.

To overcome this limitation, we employ a dynamic prompt pool with each prompt updated by multiple similar speakers. A dynamic selection strategy that finds the closest match between the prompts and the Transformer encoding layers' inputs de-

termines the association between similar speakers and the prompts. This strategy encourages knowledge sharing and avoids catastrophic forgetting.

We denote $\boldsymbol{X}_i \in \mathbb{R}^{D\times T}$ as the output feature maps of the i-th layer of the PTM before concatenating with the prompts. D is the number of output channels and T is the frame count. We denote $\boldsymbol{X}_0 \in \mathbb{R}^{D\times T}$ as the CNN encoder's output. The prompt pool is defined as:

$$\mathcal{P} = \{\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_M\}, \tag{4.1}$$

where M is the number of prompts in the pool and $\boldsymbol{P}_j \in \mathbb{R}^{D\times T'}$ represents a single prompt of length T' with embedding size D.

4.2.2 Instance-wise Prompt Searching

As illustrated in Fig. 4.1, we employ a dynamic key-to-prompt searching strategy to select suitable prompts for various inputs. The layerwise Transformer outputs determine which prompts to choose via key-to-prompt matching. To achieve this, we introduce a key function $q\colon \mathbb{R}^{T\times D} \to \mathbb{R}^D$, encoding input $\boldsymbol{X}_i$ to match the key's dimension, with $\boldsymbol{k}_i = q(\boldsymbol{X}_i) \in \mathbb{R}^D$. Also we define a prompt function $p\colon \mathbb{R}^{T'\times D} \to \mathbb{R}^D$ to map the prompt $\boldsymbol{P}_j$ to a vector of D dimensions, i.e., $\boldsymbol{p}_j = p(\boldsymbol{P}_j) \in \mathbb{R}^D$. Both $p(\cdot)$ and $q(\cdot)$ are implemented by computing the mean along the time axis, meaning that both functions do not have any learnable parameters.

For each key $\boldsymbol{k}_i$, we select a subset of prompts from $\mathcal{P}$ according to the similarity of their encoded vectors $\boldsymbol{p}_j$'s to the key. We define $\{s_t\}_{t=1}^N$ as a set of N indices from $[1, M]$. Given $\{s_t\}_{t=1}^N$, we define $\mathcal{P}_s = \{\boldsymbol{P}_{s_1}, \boldsymbol{P}_{s_2}, \ldots, \boldsymbol{P}_{s_N}\}$ as the set of top-N prompts chosen from $\mathcal{P}$. For an input $\boldsymbol{X}_i$, we use $\boldsymbol{k}_i = q(\boldsymbol{X}_i)$ as a key to select the top-N prompts by solving the following objective:

$$\begin{aligned} \{s_t^i\}_{t=1}^N &= \underset{\{s_r\}_{r=1}^N \subset [1,M]}{\operatorname{argmax}} \sum_{u=1}^{N} Sim(q(\boldsymbol{X}_i), p(\boldsymbol{P}_{s_u})) \\ \mathcal{P}_{s^i} &= \{\boldsymbol{P}_{s_1^i}, \boldsymbol{P}_{s_2^i}, \ldots, \boldsymbol{P}_{s_N^i}\} \end{aligned} \tag{4.2}$$

where $Sim: \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$ is a similarity function such as cosine.

4.2.3 Speaker Prompt Tuning

Speaker prompt tuning introduces learnable parameters into the Transformer's input space while freezing the PTM's parameters during downstream training or PTM adaptation.

We introduce a set of prompt embeddings for the i-th layer of a PTM, $\mathcal{P}_{s^i} = \{\boldsymbol{P}_{s_t^i} \in \mathbb{R}^{D \times T'}; 1 \leq t \leq N\}$, where N is the number of selected prompts. As illustrated in Fig 4.1, prompts are inserted into each Transformer layer's input space as learnable D-dimensional vectors. With prompting, the Transformer encoder's output at Layer i is:

$$\boldsymbol{Z}_i = \text{Encoder}\left(\left[\boldsymbol{P}_{s_1^i}; \boldsymbol{P}_{s_2^i}; , \ldots, \boldsymbol{P}_{s_N^i}; \boldsymbol{X}_{i-1}\right]\right), i = 1, 2, \ldots, L \tag{4.3}$$

where $\boldsymbol{Z}_i \in \mathbb{R}^{D \times (NT' + T)}$. Then, the first NT' frames of $\boldsymbol{Z}_i$ are dropped, and the remaining T frames are assigned to $\boldsymbol{X}_i$. This process is repeated for Layer $i+1$, with a new prompt subset $\mathcal{P}_{s^{i+1}}$ prepended to $\boldsymbol{X}_i$. In Eq. 4.3, L is the number of encoder layers, the colors • and • indicate learnable and frozen parameters, respectively. and the symbol ";" denotes concatenation along the time dimension.

4.2.4 Optimizing the Prompts

The frame-level speaker embeddings $\boldsymbol{Z}_i$'s at all encoding layers are linearly combined to produce a frame-level speaker feature matrix $\boldsymbol{H}^*$. The matrix is then passed to the speaker encoder g_ϕ to give an utterance-level speaker embedding vector. For each training utterance, the speaker encoder's parameters (ϕ), the selected prompts $\{\mathcal{P}_{s^i}\}_{i=1}^L$, and the combination weights $\{w_i\}_{i=1}^L$ are updated by backpropagation through minimizing the AAM-Softmax loss [108]. In Eq. 4.2, each prompt will be updated by the utterances of some similar speakers in a mini-batch.

4.3 Experiments and Results

4.3.1 Implementation Details

Pre-trained Model and Speaker Encoder. We chose HuBERT Large [136] and WavLM Large [102] as the PTMs and ECAPA-TDNN [3] as the speaker encoder.

Datasets. We used VoxCeleb1-dev [193], CN-Celeb1 [195], and CU-MARVEL [197] to fine-tune the PTMs and train the ECAPA-TDNN. CU-MARVEL, a Cantonese dementia data comprised of 280 speakers, was repurposed for speaker verification experiments. To create a challenging scenario, we trained the models on VoxCeleb1-dev and tested them on the VOiCES Challenge 2019 evaluation set (Voices19c) [222] due to the drastic difference in their acoustic conditions.

Settings. We truncated each training utterance's waveform to 2 seconds and used mini-batches of 128 utterances for fine-tuning and training. We employed AAM-Softmax [108], setting the margin to 0.2 and the scaling factor to 30. The learning rate was reduced by 3% after each epoch. For HuBERT Large and WavLM Large, the settings were $L = 24$ and $D = 1024$. We set T' to 5, N to 3 and M to 15.

4.3.2 Results and Analysis

Table 4.1 shows that using a pre-trained model for frame-level feature extraction can improve SV performance, particularly when fine-tuning the PTM is applied. Our prompt pool performs well, utilizing fewer parameters than other parameter-efficient methods. The performance improvement is attributed to our prompt pool, which dynamically learns speaker-aware prompts with significantly fewer tunable parameters.

We observed that full fine-tuning performs badly on the CU-MARVEL dataset, even worse than the performance of the fixed model (without fine-tuning). We speculate that this underperformance may arise from a language mismatch between the pre-

Table 4.1: Results on the test sets of VoxCeleb1, CN-Celeb1, and CU-MARVEL. Using HuBERT Large or WavLM Large as PTM and ECAPA-TDNN as the speaker encoder. In the column "#Parames," the first and second values are the number of adaptation parameters in a single tuning architecture for fine-tuning the PTM and the number of parameters in the ECAPA-TDNN, respectively.

PTM	Fine-tuning Method	#Params	VoxCeleb1-O		CN-Celeb1		CU-MARVEL	
			EER(%)	minDCF	EER(%)	minDCF	EER(%)	minDCF
-	-	14.7M	2.96	0.30	12.49	0.67	7.20	0.77
HuBERT Large	Fixed	0.0M+14.7M	2.76	0.30	12.05	0.61	10.40	0.93
	Full fine-tuning	316M+14.7M	1.98	0.22	10.51	0.60	11.65	0.98
	Adapter	0.5M+14.7M	2.13	0.24	10.89	0.62	8.10	0.95
	LoRA	0.5M+14.7M	2.38	0.23	10.48	0.60	9.11	0.92
	Static prompt	0.6M+14.7M	2.26	0.23	10.69	0.59	8.31	0.88
	Dynamic prompts (Ours)	0.3M+14.7M	2.17	0.21	10.61	0.58	8.20	0.86
WavLM Large	Fixed	0.0M+14.7M	1.94	0.22	11.17	0.59	6.66	0.88
	Full fine-tuning	316M+14.7M	1.39	0.16	10.47	0.56	9.09	0.94
	Adapter	0.5M+14.7M	1.68	0.19	10.83	0.63	5.58	0.81
	LoRA	0.5M+14.7M	1.88	0.21	10.89	0.63	6.83	0.88
	Static prompt	0.6M+14.7M	1.65	0.18	10.57	0.58	6.42	0.88
	Dynamic prompts (Ours)	0.3M+14.7M	1.51	0.17	10.38	0.59	6.62	0.83

trained model and the dataset and the limited number of speakers in CU-MARVEL. This could negatively affect the pre-trained model's parameters during full tuning. In contrast, the larger speaker count of CN-Celeb facilitates the training of a more effective speaker encoder. Thus, this issue is less pronounced in the CN-Celeb dataset. While LoRA is effective for natural language processing, its efficacy in speaker verification is inferior, as shown in Table 4.1. The performance gap may be due to the focus on capturing the phonetic properties of utterances during the pre-training phase [142]. In contrast, speaker verification demands discrimination between speakers, which is not achievable by merely modifying the attention weights.

Table 4.2: Ablation studies on VoxCeleb1. The train and test data are VoxCeleb1-dev and VoxCeleb1-eval, respectively.

Ablated component	EER(%)	minDCF
w/o prompt pool	1.65	0.18
w/o key-value pairs	1.71	0.18
Dynamic prompts (Ours)	1.51	0.17

4.3.3 Ablation Study

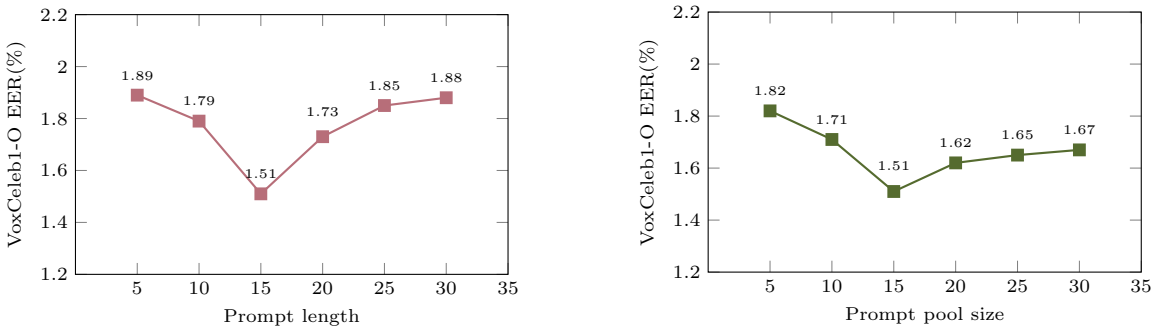
Table 4.2 (row 1) shows that removing the prompt pool but using a set of static prompts for each Transformer encoder layer leads to a significant drop in performance. This performance drop indicates severe catastrophic forgetting and knowledge interference among speakers when using static prompts. Conversely, our prompt pool can effectively encode speaker-specific knowledge.

Table 4.2 (row 2) demonstrates that randomly selecting the prompts from the pool adversely affects performance. This result underscores the critical role played by the key-prompt search to ensure that each prompt is adapted by a group of relevant speakers whose speeches, after transformation by the Transformer encoders, are close to the prompt.

4.3.4 Effect of Hyperparameters on Dynamic Prompts

Our prompt tuning has three key hyperparameters: prompt pool size M, single prompt T', and the prompt selection size N. Intuitively, M determines the capacity of learnable prompts, T' represents the capacity of a single prompt to encode knowledge, and NT' represents the capacity of the layerwise prompts in adapting the corresponding Transformer layer.

We fixed T' to 5 and M to 15 and then continuously increased N to identify the



(a) Length (b) Size

Figure 4.2: Results on Voxceleb1-O. The training dataset is VoxCeleb1-dev, and the PTM is WavLM Large. The total length of the prompt is NT'.

optimal prompt length. Results in Fig. 4.2 (upper panel) show that a too-small T' negatively impacts performance, whereas an oversized prompt can lead to knowledge overfitting. We hypothesize that optimal capacity for prompts is essential for encoding specific aspects of shared knowledge.

We also set T' to 5 and N to 3 and progressively increased M. Results in Fig. 4.2 (lower panel) suggest that enlarging the prompt pool size enhances performance, demonstrating the necessity of a sufficiently large pool to encode diverse speaker-specific knowledge. However, excessively increasing the prompt pool size does not significantly enhance performance.

4.3.5 Generalization Analysis

We trained the model on VoxCeleb1 and tested them on the evaluation set of Voices19c (v19-eval), acknowledging the acoustic differences between VoxCeleb and VOiCES. Speech files in v19-eval-wpe were subject to weighted prediction error (WPE) processing. Table 4.3 shows that adapters and static prompts yield similar results, whereas dynamic prompts exhibit improvement. This result suggests that dynamic prompts are better generalized to unseen speakers in different acoustic environments.

Table 4.3: The performance of dynamic prompts and conventional fine-tuning methods on Voices19c. The train data is VoxCeleb1-dev.

Fine-tuning Method	v19-eval		v19-eval-wpe	
	EER(%)	minDCF	EER(%)	minDCF
Adapter	20.02	0.97	18.62	0.97
Static prompts	20.22	0.96	17.75	0.87
Dynamic prompts (Ours)	19.06	0.93	15.99	0.86

4.4 Conclusions

This paper introduces a dynamic prompt-tuning method for speaker verification. Specifically, our dynamic prompts approach uses speaker representations as conditions to generate speaker-aware prompts, avoiding implicit correlations with previously seen speakers. Furthermore, we employ a prompt pool to minimize the number of tunable parameters without sacrificing the effectiveness of prompt embeddings. Our experiments in various settings demonstrate that our method surpasses current parameter-efficient baselines in speaker verification.

Chapter 5

Spectral-Aware Low-Rank Adaptation for Speaker Verification

5.1 Introduction

The primary goal of parameter-efficient fine-tuning (PEFT) is to reduce the number of tunable parameters compared to full fine-tuning. This approach conserves computational resources and enables easy sharing of lightweight, fine-tuned models [1, 142, 223]. Among these methods, the low-rank adaptation (LoRA) model [224] stands out for its simplicity and effectiveness. LoRA tunes an additional, trainable low-rank matrix, resulting in zero inference latency after integrating the adapter into the pre-trained model. Since its introduction, several LoRA variants have emerged. For instance, AdaLoRA [225], IncreLoRA [226], and DyLoRA [227] dynamically adjust the rank of the LoRA adaptation matrices to enhance tuning efficiency. A more recent variant, DoRA [228], decomposes a pre-trained weight matrix into a magnitude vector and a series of direction vectors.

Although LoRA is simple and effective, its low-rank constraint may be suboptimal for tasks that demand high representation capacity. In particular, for a rank r approximation of a matrix $\boldsymbol{W}$, the optimal solution corresponds to the largest r singular values and their corresponding singular vectors—components that LoRA does not explicitly leverage. This limitation implies that potentially valuable directions in the parameter space, captured by these singular vectors, remain underutilized.

Previous research, such as [229, 230, 231, 232], explored incorporating the spectral information from the pre-trained model's weight matrices into PEFT by introducing a spectral adaptation mechanism that updates the top singular vectors of the pre-trained weight matrices. Other studies [233, 234, 235, 236, 237] further exploited the spectral space of pre-trained weight matrices, adjusting both singular values and singular vectors during fine-tuning. These approaches focus on the spectral components' magnitude and directions, aiming for a more refined and effective adaptation. Collectively, these works contribute to a deeper understanding of the relationship between the spectral information of weight matrices and model performance. In this work, we leverage the spectral information of the pre-trained weight matrices during fine-tuning to enhance the model's performance.

This chapter introduces a spectral fine-tuning (SpectralFT) method based on low-rank adaptation to adapt a pre-trained Transformer-based speech model for speaker verification. Specifically, we decompose a weight matrix $\boldsymbol{W}$ using singular value decomposition (SVD). Based on the magnitude of the singular values, $\boldsymbol{W}$ is divided into two components: a principal matrix $\boldsymbol{W}_p$, associated with the larger singular values, and a minor matrix $\boldsymbol{W}_m$, associated with the smaller singular values. The principal matrix encapsulates the core of the pre-trained knowledge, and we approximate the original parameter matrix $\boldsymbol{W}$ using this low-rank matrix $\boldsymbol{W}_p$. The principal matrix $\boldsymbol{W}_p$ is frozen, and low-rank adaptation is applied to adapt the singular vectors of $\boldsymbol{W}_p$ during fine-tuning. SpectralFT aims to effectively capture task-specific knowledge during fine-tuning while preserving and leveraging the pre-trained information.

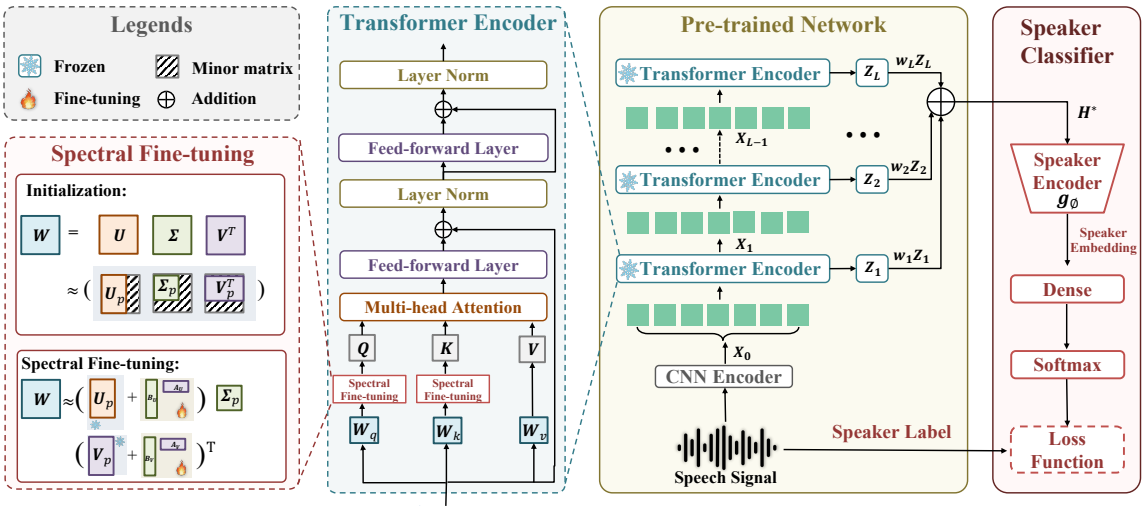


Figure 5.1: The architecture of the proposed SpectralFT. The principal singular components ($\boldsymbol{U}_p, \boldsymbol{V}_p, \boldsymbol{\Sigma}_p$) are retained to form a low-rank approximation of the original weight matrix $\boldsymbol{W}$, which is then fine-tuned using the principle of LoRA. During fine-tuning, only the low-rank matrices $\boldsymbol{B}_U$, $\boldsymbol{A}_U$, $\boldsymbol{B}_V$, and $\boldsymbol{A}_V$ are updated, while the principal matrices $\boldsymbol{U}_p$ and $\boldsymbol{V}_p$ remain frozen. For the operations and principles of the Transformer Encoder, Pre-trained Network, and Speaker Classifier, readers are referred to [1, 2].

5.2 Methodology

As shown in Fig. 5.1, we utilize SVD to decompose the pre-trained weight matrices, exploring the mechanisms of LoRA within the SVD framework. Our method strikes a good balance between preserving the generalization capacity of the pre-trained parameters and enabling task-specific adaptation.

5.2.1 Low-Rank Adaptation

LoRA [224] assumes that the updates to a pre-trained weight matrix $\boldsymbol{W}_0 \in \mathbb{R}^{m\times n}$ are low-rank, thereby allowing the changes to be represented by two trainable low-rank matrices: $\boldsymbol{B} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{A} \in \mathbb{R}^{r\times n}$. Specifically, the updated weight matrix is

expressed as:

$$\boldsymbol{W} = \boldsymbol{W}_0 + \Delta\boldsymbol{W} = \boldsymbol{W}_0 + \frac{\alpha}{r}\boldsymbol{B}\boldsymbol{A}, \tag{5.1}$$

where $\Delta\boldsymbol{W}$ represents the weight updates. Here, α and r are hyperparameters controlling the scale and the LoRA rank, respectively, with $r \ll \min(m, n)$.

The pre-trained matrix $\boldsymbol{W}_0$ remains fixed during fine-tuning, which significantly reduces the number of trainable parameters, as both $\boldsymbol{A}$ and $\boldsymbol{B}$ are low-rank matrices. The $\boldsymbol{B}$ matrix is initialized to zero, while the $\boldsymbol{A}$ matrix is initialized using a Gaussian distribution with zero mean and unit variance. This initialization strategy ensures that $\Delta\boldsymbol{W} = \boldsymbol{0}$ at the start of fine-tuning. Because LoRA only modifies the linear matrices in the Transformer model, the low-rank matrices $\boldsymbol{B}\boldsymbol{A}$'s can be seamlessly merged into the pre-trained linear matrices. This property results in no additional computation or GPU memory during inferencing.

However, the vanilla LoRA method, which constrains updates to a fixed low-rank subspace, presents a significant limitation. Specifically, the low-rank nature of LoRA restricts the difference between the fine-tuned weight matrix $\boldsymbol{W}_0 + \frac{\alpha}{r}\boldsymbol{B}\boldsymbol{A}$ and the pre-trained weights $\boldsymbol{W}_0$ to a low-rank matrix. This constraint severely limits LoRA's ability to fine-tune a model to arbitrary target tasks.

5.2.2 Singular Value Decomposition

Given a matrix $\boldsymbol{W} \in \mathbb{R}^{m\times n}$, its SVD is denoted as $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}}$, where $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m] \in \mathbb{R}^{m\times m}$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n] \in \mathbb{R}^{n\times n}$. The columns of $\boldsymbol{U}$ are the left singular vectors, and the columns of $\boldsymbol{V}$ are the right singular vectors. The diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}$ contains the singular values of $\boldsymbol{W}$ in descending order.

This decomposition can also be reformulated in matrix form. The matrix $\boldsymbol{U}$ can be column-wise partitioned into a *p*rincipal matrix and a *m*inor matrix: $\boldsymbol{U} = [\boldsymbol{U}_p, \boldsymbol{U}_m]$, where $\boldsymbol{U}_p = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k]$ and $\boldsymbol{U}_m = [\boldsymbol{u}_{k+1}, \boldsymbol{u}_{k+2}, \ldots, \boldsymbol{u}_m]$ are the left singular

vectors corresponding to the principal and minor singular values, respectively.[1] The matrices $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ are partitioned similarly. Thus, the SVD of $\boldsymbol{W}$ can be expressed as:

$$\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}} = \boldsymbol{U}_p\boldsymbol{\Sigma}_p\boldsymbol{V}_p^{\mathsf{T}} + \boldsymbol{U}_m\boldsymbol{\Sigma}_m\boldsymbol{V}_m^{\mathsf{T}} = \boldsymbol{W}_p + \boldsymbol{W}_m. \tag{5.2}$$

5.2.3 Spectral Fine-tuning

Inspired by the parameter efficiency of LoRA and the close connection between matrix rank and spectral representation, we explore a spectral fine-tuning mechanism. The idea is to apply SVD to a pre-trained model's weight matrix, followed by fine-tuning the principal columns of the singular vector matrices. To this end, we approximate the SVD of a weight matrix $\boldsymbol{W}$ by the spectral representation of $\boldsymbol{W}_p$ in Eq. 5.2, i.e., $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}} \approx \boldsymbol{U}_p\boldsymbol{\Sigma}_p\boldsymbol{V}_p^{\mathsf{T}}$. We define the additive spectral adapter as

$$\text{SpectralFT}(\boldsymbol{W}) := [\boldsymbol{U}_p + \boldsymbol{\Delta}_U]\boldsymbol{\Sigma}_p[\boldsymbol{V}_p + \boldsymbol{\Delta}_V]^{\mathsf{T}}, \tag{5.3}$$

where $\boldsymbol{U}_p \in \mathbb{R}^{m\times k}$ and $\boldsymbol{V}_p \in \mathbb{R}^{n\times k}$ represent the top-k columns of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. The adaptation set $\boldsymbol{\Delta} = \{\boldsymbol{\Delta}_U, \boldsymbol{\Delta}_V\}$ consists of trainable matrices with the same dimensions as $\boldsymbol{U}_p$ and $\boldsymbol{V}_p$, respectively. As observed in LASER [238], the minor singular components of a weight matrix often contain noisy information, whereas the principal singular components capture important features across tasks. Therefore, we discard $\boldsymbol{U}_m$ and $\boldsymbol{V}_m$ in Eq. 5.2.

To leverage the advantage of LoRA, we define $\boldsymbol{\Delta}_U \equiv \frac{\alpha}{r}\boldsymbol{B}_U\boldsymbol{A}_U$, where $\boldsymbol{B}_U \in \mathbb{R}^{m\times r}$ and $\boldsymbol{A}_U \in \mathbb{R}^{r\times k}$, such that $r \ll k$. The matrix $\boldsymbol{B}_U$ is initialized to zero, while $\boldsymbol{A}_U$ is initialized using a Gaussian distribution. The adapter weights $\boldsymbol{B}_U$ and $\boldsymbol{A}_U$ are initialized such that $\boldsymbol{B}_U\boldsymbol{A}_U = \boldsymbol{0}$. The same strategy is applied to $\boldsymbol{\Delta}_V \equiv \frac{\alpha}{r}\boldsymbol{B}_V\boldsymbol{A}_V$, where $\boldsymbol{B}_V \in \mathbb{R}^{n\times r}$ and $\boldsymbol{A}_V \in \mathbb{R}^{r\times k}$. During training, only the elements of $\boldsymbol{B}_U$, $\boldsymbol{A}_U$, $\boldsymbol{B}_V$, and $\boldsymbol{A}_V$ are updated.

[1] The subscript of a matrix (e.g., p and m in $\boldsymbol{U}_p$ and $\boldsymbol{U}_m$) is used for naming the matrix, whereas the subscript of a vector (e.g., k in $\boldsymbol{u}_k$) represents the vector's position in a matrix.

5.2.4 Computation Considerations

We propose incorporating spectral information into the fine-tuning process for the $\boldsymbol{W}_q$ and $\boldsymbol{W}_k$ matrices in the attention mechanism of the Transformer model. Our method allows for flexible parameter budgets by adjusting the values of r and k. Specifically, we fine-tune the top-k columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ using additive tuning, which requires storing only $\boldsymbol{B}_U$, $\boldsymbol{A}_U$, $\boldsymbol{B}_V$, and $\boldsymbol{A}_V$.

The only overhead is the runtime and GPU storage during training. Because our method involves only matrix multiplication during the forward pass, it should run as efficiently as LoRA. While the SVD process may introduce some runtime overhead, it is a one-time operation per model and can be reused for subsequent fine-tuning on different downstream tasks.

5.3 Experiments and Results

5.3.1 Implementation Details

We selected HuBERT-Large [136] and WavLM-Large [102] as the pre-trained models (PTMs) and ECAPA-TDNN [3] as the speaker encoder. VoxCeleb1-dev [193] and CN-Celeb1 [195] were used to fine-tune the PTMs and train the ECAPA-TDNN. We truncated each training utterance to 2 seconds and used mini-batches of 256 utterances for fine-tuning and training. AAM-Softmax [108] was employed with a margin of 0.2 and a scaling factor of 30. The rank r was set to 16, and the number of top singular vectors k was 256.

5.3.2 Results and Analysis

Table 5.1 shows that utilizing a pre-trained model for frame-level feature extraction enhances SV performance (compare Rows 1, 2, and 3), especially after fine-tuning the pre-trained models. We compare our approach with three widely used parameter-efficient fine-tuning methods: Adapter [207] (results extracted from [2]), static prompt tuning [1] (results extracted from [2]), and LoRA (results extracted from [2]) which was used to fine-tune the $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, and $\boldsymbol{W}_v$ matrices in the attention mechanism, with the scaling factor ($\frac{\alpha}{r}$ in Eq. 5.1) set to 0.1. The results demonstrate that our proposed method outperforms all others on both datasets, with the improvement being particularly pronounced compared to traditional LoRA. This advantage arises from the SVD being able to preserve the most critical features relevant to speaker characteristics while ignoring the unimportant factors that may negatively affect speaker verification. Therefore, the SVD provides a top spectral space that is more relevant to speakers for LoRA-style fine-tuning. With $k \gg r$, SpectralFT can maintain sufficient spectral contents without overparameterizing the LoRA adaptation matrix, an important advantage of SpectralFT over conventional LoRA.

5.3.3 Investigating Different Rank Settings

We examined the impact of varying the rank r on the fine-tuned WavLM-Large model. As shown in Fig. 5.2, SpectralFT with a rank of 16 yielded the best performance. The results indicate that selecting an appropriate rank is crucial for good performance when fine-tuning with SpectralFT. Insufficient rank means the subspace for fine-tuning the weight matrices is too restrictive, causing the fine-tuned model to fail to adapt to the downstream task. Conversely, while a higher rank allows the model to capture more details about the downstream task, it may also result in overfitting by learning noise from the adaptation data. Our results show that a rank of 16 strikes a good balance, suggesting that a moderate model capacity is sufficient to capture key

Table 5.1: Performance on the test sets of VoxCeleb1 and CN-Celeb1, using HuBERT-Large or WavLM-Large as PTM and ECAPA-TDNN as the speaker encoder. Row 1 uses Filterbank features as input to the ECAPA-TDNN. Results based on full fine-tuning are in italics. They are expected to be the best. The best results based on other fine-tuning methods are in bold.

PTM	Row	Fine-tuning Method	VoxCeleb1-O		CN-Celeb1	
			EER(%)	minDCF	EER(%)	minDCF
None	1	None	2.96	0.30	12.49	0.67
HuBERT-Large	2	None	2.76	0.30	12.05	0.61
	3	Full fine-tuning	*1.98*	*0.22*	*10.51*	*0.60*
	4	Adapter [2]	**2.13**	0.24	10.89	0.62
	5	Static prompt tuning [2]	2.26	0.23	10.69	0.59
	6	LoRA (r=16, $\frac{\alpha}{r}$=0.1) [2]	2.38	0.23	10.48	0.60
	7	**SpectralFT (Ours)**	2.31	**0.22**	**10.45**	**0.58**
WavLM-Large	8	None	1.94	0.22	11.17	0.59
	9	Full fine-tuning	*1.39*	*0.16*	*10.47*	*0.56*
	10	Adapter [2]	1.68	0.19	10.83	0.63
	11	Static prompt tuning [2]	1.65	0.18	10.57	0.58
	12	LoRA (r=16, $\frac{\alpha}{r}$=0.1) [2]	1.88	0.21	10.89	0.63
	13	**SpectralFT (Ours)**	**1.47**	**0.16**	**10.69**	**0.56**

features while maintaining strong generalization ability.

5.3.4 Analysis of Principle Columns

We conduct experiments to investigate the influence of the number of singular components on fine-tuning performance. We set the dimensions of the retained primary singular value components (k) to 64, 128, 256, 512, and 1024. Table 5.2 shows that the best results are achieved when retaining 256 components. A spectral space with 256 dimensions is enough because beyond which the singular values are too small for

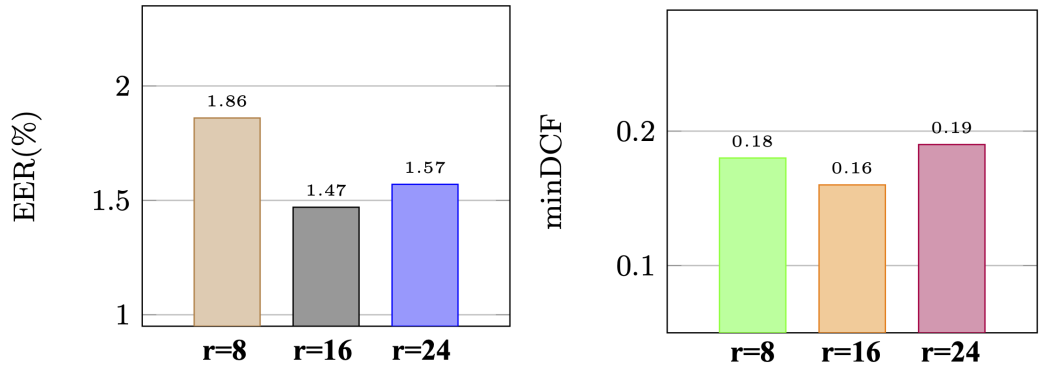


Figure 5.2: Results on VoxCeleb1-O for different ranks, using WavLM-Large as the PTM.

Table 5.2: Results on VoxCeleb1-eval using different number of principal columns (k) in $\boldsymbol{U}$.

No. of Principal columns k	VoxCeleb1-O	
	EER(%)	minDCF
64	1.83	0.22
128	1.59	0.21
256	1.47	0.16
512	1.51	0.16
1024	1.58	0.18

the spectral space to focus on the speaker features. The variation in the low spectral space contains more noise, which could interfere with the speaker verification task.

5.3.5 Analysis of the Effect of Singular Vectors

To explore the effect of different singular value settings, we conducted experiments in which only the principal singular components were retained, and we denoted the subspace as "$\boldsymbol{U}_p\boldsymbol{\Sigma}_p\boldsymbol{V}_p^{\mathsf{T}}$". We explored the effect of having $\boldsymbol{\Delta}_U$ and $\boldsymbol{\Delta}_V$ in Eq. 5.3. In the third row of Table 5.3, we used both the principal and minor singular components,

Table 5.3: Results of different subspace fine-tuning strategies on VoxCeleb1-eval, using WavLM-Large as the PTM.

Subspace		$\boldsymbol{\Delta}_U$ and $\boldsymbol{\Delta}_V$	VoxCeleb1-O	
Principal	Minor	(in Principal Subspace)	EER(%)	minDCF
$\boldsymbol{U}_p\boldsymbol{\Sigma}_p\boldsymbol{V}_p^\mathsf{T}$	None	✓	1.47	0.16
$\boldsymbol{U}_p\boldsymbol{\Sigma}_p\boldsymbol{V}_p^\mathsf{T}$	None	✗	1.60	0.17
$\boldsymbol{U}_p\boldsymbol{\Sigma}_p\boldsymbol{V}_p^\mathsf{T}$	$\boldsymbol{U}_m\boldsymbol{\Sigma}_m\boldsymbol{V}_m^\mathsf{T}$	✓	1.65	0.17
$\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\mathsf{T}$		✗	1.68	0.20

fine-tuning the primary singular value components $\boldsymbol{U}_p$ and $\boldsymbol{V}_p$, while keeping the minor components $\boldsymbol{U}_m$ and $\boldsymbol{V}_m$ frozen. In the fourth row of Table 5.3, we considered performing SVD on the weight matrices as the baseline and denoted it as "$\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\mathsf{T}$".

The results presented in Table 5.3, comparing the first and second rows, illustrate the effectiveness of applying our SpectralFT method. Comparing the first and third rows indicates that incorporating $\boldsymbol{U}_m$ and $\boldsymbol{V}_m$ led to a decline in performance, as $\boldsymbol{U}_m$ and $\boldsymbol{V}_m$ introduced more speaker verification-unfavorable noise. Comparing the first and fourth rows demonstrates that retaining the principal singular components, discarding minor singular components, and applying SpectralFT can significantly improve performance.

5.3.6 Analyze the Fine-tuning Positions

To identify the most effective weight matrices for spectral fine-tuning, we apply SpectralFT progressively to $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, and $\boldsymbol{W}_v$ in the Transformer attention mechanism. We also compared the results with other low-rank approximation fine-tuning methods, specifically LoRA and DoRA. In Table 5.4, r represents the rank, and α represents different scaling factors in Eq. 5.1. The experimental results indicate that the best performance is achieved when fine-tuning the $\boldsymbol{W}_q$ and $\boldsymbol{W}_k$ matrices. In

Table 5.4: Results on the test sets of VoxCeleb1 with fine-tuning different weight matrices.

Methods	Weight Type			VoxCeleb1-O	
	$\boldsymbol{W}_q$	$\boldsymbol{W}_k$	$\boldsymbol{W}_v$	EER(%)	minDCF
LoRA (r=16, $\frac{\alpha}{r}$=1)	✓	✗	✗	1.59	0.19
	✓	✓	✗	1.58	0.18
	✓	✓	✓	1.88	0.21
DoRA (r=16)	✓	✗	✗	1.67	0.19
	✓	✓	✗	1.54	0.17
	✓	✓	✓	1.65	0.18
SpectralFT (r=16, $\frac{\alpha}{r}$=1)	✓	✗	✗	1.60	0.18
	✓	✓	✗	1.47	0.16
	✓	✓	✓	1.64	0.19

Transformer-based models, the $\boldsymbol{W}_q$ and $\boldsymbol{W}_k$ matrices are responsible for computing attention scores, which determine how the model selects information from the input data. By adjusting the $\boldsymbol{W}_q$ and $\boldsymbol{W}_k$ matrices, SpectralFT can more precisely control the attention without altering the value matrix $\boldsymbol{W}_v$.

5.4 Conclusions

In this work, we explore integrating spectral information from the pre-trained model weight matrices into existing PEFT by introducing a spectral adaptation mechanism that updates only the top singular vectors of the pre-trained weight matrices. Empirically, we demonstrate the superiority of our proposed spectral adaptation method over various recent PEFT approaches through extensive experiments.

Chapter 6

Disentangling Speaker and Content Using Latent Diffusion

6.1 Introduction

A speech signal is represented as a one-dimensional waveform. Despite its apparent simplicity, a speech waveform encodes a wealth of high-level information such as phonemes, tone, emotion, gender, and speaker identity. However, attributes like speaking style, prosody, recording conditions, and noise levels are challenging to annotate [165, 239, 240]. To extract accurate speaker representations, existing methods employ phonetic content representations as a reference for speaker embeddings. Specifically, these methods include: (1) leveraging pre-trained automatic speech recognition (ASR) models [143, 241, 242] and (2) utilizing jointly trained multi-task models with additional modules for content representation [243, 244]. These approaches demonstrate that incorporating content representations enhances speaker recognition performance.

However, both strategies face limitations in practical applications. Pre-trained ASR models significantly increase model size and computational complexity during infer-

ence. Meanwhile, joint training with additional modules requires either a separate dataset with both text labels and speaker identities or a unified dataset containing both, which is often costly and challenging to obtain.

Disentangled representations have garnered considerable attention in recent research due to their ability to capture distinct variations in data generation. These variations often carry semantic meaning, facilitating the removal of irrelevant factors and reducing sample complexity for downstream learning tasks. In speaker verification, an ideal disentangled representation can isolate time-invariant features (e.g., speaker characteristics) from dynamic information (e.g., speech content). Moreover, downstream tasks such as speech recognition and speaker classification can benefit significantly from these representations by utilizing the separated components to improve representation learning.

Recent studies have investigated disentangled representation learning through variational autoencoders (VAEs) [9, 245] and generative adversarial networks (GANs). Models such as SpeechTripleNet [246], AnnealVAE [247], and JointVAE [248] set channel capacity for distinct latent variables to promote disentanglement. InfoGAN [249] divided the latent space and incorporated a mutual information regularization term into the standard GAN loss to enhance disentanglement. Similarly, Mathieu *et al.* [250] partitioned the encoding space into style and content components, employing adversarial training to encourage data points within the same class to share content representations while maintaining diverse style features.

Denoising diffusion models have recently demonstrated superior performance in disentangled representation learning, offering more stable training and higher representation fidelity compared to GANs and VAE-based models. Diffusion models address the challenge of representing complex, high-dimensional probability distributions by decomposing the problem into T incremental steps. At each step, the model transforms the noise data from a simpler distribution (e.g., the simplest Gaussian prior at $t = T$) to a more complex one (e.g., the real data distribution at $t = 0$). This

iterative inference and denoising paradigm enables the model to map a simple distribution to a complex one through gradual refinement over many steps. However, the latent variables produced by diffusion models often lack high-level semantics and other desirable properties, such as speaker features.

To overcome the aforementioned limitations, we condition a denoising diffusion implicit model (DDIM) [251] on speaker features and propose a disentangled sequential model that leverages the capabilities of the DDIM to learn multi-level representations. Specifically, we employ a learnable speaker encoder to capture utterance-level speaker characteristics while the DDIM decodes and models the Gaussian variations in data. A latent vector represents the speaker features, and additional frame-level latent vectors capture dynamic information such as speech content. The DDIM's forward and generative processes are conducted within the joint latent space of speaker features and speech content. We applied our proposed disentanglement method to the speech representations generated by the pre-trained models WavLM [102] and HuBERT [136]. Experiments conducted on the speaker verification datasets VoxCeleb demonstrate that our method can effectively extract accurate speaker embeddings.

6.2 Methodology

We denote the input speech sequence as $\boldsymbol{x}_{1:N} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$, where $\boldsymbol{x}_i$ is the filter-bank feature vector corresponding to the i-th frame, and N is the number of frames in the sequence. To facilitate an informative global latent representation $\boldsymbol{z}_T$ for the decoding process, we introduce a conditional DDIM decoder, represented by $p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{f}_s)$. This decoder is conditioned on an auxiliary latent variable $\boldsymbol{f}_s$ obtained through a speaker encoder $\boldsymbol{f}_s = \text{Enc}_\phi(\boldsymbol{x}_{1:N})$, which maps the entire input sequence $\boldsymbol{x}_{1:N}$ to speaker representation $\boldsymbol{f}_s$. $\boldsymbol{f}_s$ is fed into two linear heads to produce the mean vector $\boldsymbol{\mu}_s$ and standard deviation vector $\boldsymbol{\sigma}_s$. Then, the speaker vector $\boldsymbol{s}$ is obtained by sampling the Gaussian distribution defined by these mean and standard deviation

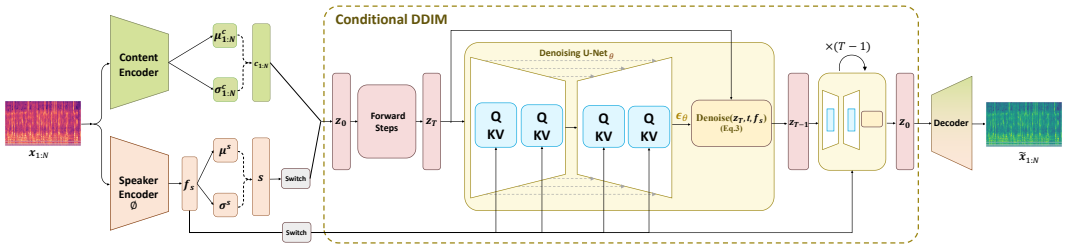


Figure 6.1: The autoencoder comprises a speaker encoder, a content encoder, a conditional DDIM, and a speech decoder. The speaker encoder utilizes an ECAPA-TDNN [3] to transform the input speech $\boldsymbol{x}_{1:N}$ into a speaker representation $\boldsymbol{f}_s$, which is further transformed to $\boldsymbol{\mu}^s$ and $\boldsymbol{\sigma}^s$ through two linear heads. Similarly, $\boldsymbol{\mu}^c_{1:N}$ and $\boldsymbol{\sigma}^c_{1:N}$ can be obtained from a long short-term memory (LSTM) network with two linear heads. The "Switch" module changes the dimension of input vectors. For notational simplicity, we use the same symbols before and after the change of dimension. The dotted brace represents Gaussian sampling, which is performed by a reparameterization trick [4]. A conditional DDIM that serves as both a stochastic encoder $\boldsymbol{z}_0 \rightarrow \boldsymbol{z}_T$ and a deterministic decoder $\boldsymbol{z}_{t-1} = \text{Denoise}(\boldsymbol{z}_t, \boldsymbol{f}_s, t)$. $\boldsymbol{z}_0 \in \mathbb{R}^{2D \times N}$, where D is the dimension of $\boldsymbol{c}_i$ and $\boldsymbol{s}$. Similarly, $\boldsymbol{z}_T, \boldsymbol{z}_{T-1}, \ldots, \boldsymbol{z}_1$ have the same dimensions as $\boldsymbol{z}_0$.

vectors. The module "Switch" in Fig. 6.1 changes the dimension of $\boldsymbol{s}$ by repeating it N times so that the resulting matrix can be concatenated with $\boldsymbol{c}_{1:N}$. We refer to the network in Fig. 6.1 as **D**isentengled **L**atent **D**iffusion–**A**uto**E**ncoder (DLD–AE).

6.2.1 Speaker Encoder

We utilize an ECAPA-TDNN model [3] to transform the input speech sequence $\boldsymbol{x}_{1:N}$ to a representative vector $\boldsymbol{f}_s$. This vector captures crucial speaker information for the DDIM decoder (the yellow boxes in Fig. 6.1), expressed as $p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{f}_s)$, to perform the denoising process and predict the output latent vector $\tilde{\boldsymbol{z}}_0 \equiv (\boldsymbol{z}_0, \boldsymbol{f}_s)$. By conditioning DDIM on an enriched information vector $\boldsymbol{f}_s$, we enhance the efficiency and accuracy of the denoising operation, ultimately leading to a more reliable generation

of latent representations.

6.2.2 Content Encoder

We employ an LSTM with two linear heads as the content encoder to transform $\boldsymbol{x}_{1:N}$ into $\boldsymbol{c}_{1:N}$, where $\boldsymbol{c}_i$ denotes the dynamic state learned at frame i. We assume that each $\boldsymbol{c}_i$ depends on the preceding dynamic variables, denoted as $\boldsymbol{c}_{<i} \equiv \{\boldsymbol{c}_0, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_{i-1}\}$, with $\boldsymbol{c}_0 = \mathbf{0}$.

6.2.3 Reverse Diffusion Process

Our proposed conditional DDIM's reverse process utilizes the input $\tilde{\boldsymbol{z}}_{t-1} \equiv (\boldsymbol{z}_t, \boldsymbol{f}_s)$, which comprises the DDIM encoder's output and speaker representation, to generate an output latent vector. Using a denoising U-Net, each block of the conditional DDIM decoder models the probability distribution $p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{f}_s)$:

$$p_\theta(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{f}_s) = \begin{cases} \mathcal{N}(\boldsymbol{z}_{t-1}; f_\theta(\boldsymbol{z}_1, 1, \boldsymbol{f}_s), \sigma_1^2 \boldsymbol{I}) & \text{if } t = 1, \\ q_\sigma(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, f_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s)) & \text{otherwise} \end{cases}, \tag{6.1}$$

where σ_1 is set to 0. Following the approach in Song *et al.* [251], the inference distribution q_σ in Eq. 6.1 is defined as follows:

$$\begin{aligned} q_\sigma = \mathcal{N}\Bigg(& \boldsymbol{z}_{t-1}; \sqrt{\alpha_{t-1}} f_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s) \\ & + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{\boldsymbol{z}_t - \sqrt{\alpha_t} f_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s)}{\sqrt{1 - \alpha_t}}, \sigma_t^2 \boldsymbol{I} \Bigg) \end{aligned} \tag{6.2}$$

where σ_t is set to 0. We implement f_θ in Eqs. 6.1 and 6.2 using a noise prediction network $\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s)$:

$$f_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s) \equiv \text{Denoise}(\boldsymbol{z}_t, t, \boldsymbol{f}_s) = \frac{\boldsymbol{z}_t - \sqrt{1 - \alpha_t} \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s)}{\sqrt{\alpha_t}}, \tag{6.3}$$

where $\boldsymbol{\epsilon}_\theta$ is implemented by a U-Net as shown in Fig 6.1.

The training process involves optimizing the $\mathcal{L}_{DDIM}$ loss with respect to parameters θ and ϕ:

$$\mathcal{L}_{DDIM} = \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{z}_0, \boldsymbol{\epsilon}_t} \left[\parallel \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, \boldsymbol{f}_s) - \boldsymbol{\epsilon}_t \parallel_2^2 \right], \tag{6.4}$$

where $\boldsymbol{f}_s = \text{Enc}_\phi(\boldsymbol{x}_{1:N})$, $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $\boldsymbol{z}_t = \sqrt{\alpha_t}\boldsymbol{z}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t$, and T is an integer, e.g., 100. Note that this simplified loss function optimizes the DDIM but does not optimize the actual variational lower bound.

6.2.4 Disentangled Sequential Variational Autoencoder

We define the global latent representation as $\boldsymbol{z}_0 \equiv (\boldsymbol{s}, \boldsymbol{c}_{1:N})$. Our formulation is based on the intuition that sequence variations can be decomposed into time-dependent dynamic components $\{\boldsymbol{c}_i\}$ and a static component $\boldsymbol{s}$. We assume independence between the static variable $\boldsymbol{s}$ and the dynamic variables $\boldsymbol{c}_{1:N}$, implying $p(\boldsymbol{z}_0) = p(\boldsymbol{s}, \boldsymbol{c}_{1:N}) = p(\boldsymbol{s})p(\boldsymbol{c}_{1:N})$. The static component remains constant across all frames within a given utterance but differs across different utterances.

In a speech signal, the phonetic transcription governs the movement of the vocal tract and the produced sounds over time, while the speaker's identity remains fixed throughout an utterance. Based on these assumptions, we derive the following complete likelihood [164]:

$$\begin{aligned} p(\boldsymbol{x}_{1:N}, \boldsymbol{z}_0) &= p(\boldsymbol{x}_{1:N}, \boldsymbol{s}, \boldsymbol{c}_{1:N}) \\ &= p(\boldsymbol{s}, \boldsymbol{c}_{1:N}) p(\boldsymbol{x}_{1:N} | \boldsymbol{s}, \boldsymbol{c}_{1:N}) \\ &= p(\boldsymbol{s}) \left[\prod_{i=1}^{N} p(\boldsymbol{c}_i | \boldsymbol{c}_{<i}) p(\boldsymbol{x}_i | \boldsymbol{s}, \boldsymbol{c}_i) \right]. \end{aligned} \tag{6.5}$$

We define $p(\boldsymbol{s})$ in Eq. 6.5 as a standard Gaussian $\mathcal{N}(\boldsymbol{s}; \mathbf{0}, \boldsymbol{I})$. We assume that $p(\boldsymbol{c}_i | \boldsymbol{c}_{<i})$ follows a Gaussian distribution:

$$p(\boldsymbol{c}_i | \boldsymbol{c}_{<i}) = \mathcal{N}(\boldsymbol{c}_i; \boldsymbol{\mu}_i(\boldsymbol{c}_{<i}), \text{diag}((\boldsymbol{\sigma}_i(\boldsymbol{c}_{<i}))^2)), \tag{6.6}$$

where $\boldsymbol{\mu}_i(\cdot)$ and $\boldsymbol{\sigma}_i(\cdot)$ can be modeled by an LSTM followed by two linear heads. Since both $\boldsymbol{\mu}_i(\cdot)$ and $\boldsymbol{\sigma}_i(\cdot)$ are conditioned on the temporal context $\boldsymbol{c}_{<i}$, their derivation at frame i requires access to the history $\boldsymbol{c}_{<i}$. To sample $\boldsymbol{c}_i$ from $p(\boldsymbol{c}_i|\boldsymbol{c}_{<i})$, $\boldsymbol{c}_{i-1}$ is first passed through the LSTM cells to forward one step, generating $\boldsymbol{\mu}_i(\cdot)$ and $\boldsymbol{\sigma}_i(\cdot)$ via linear transformation layers. The reparameterization trick is then applied to draw a sample from the resulting distribution [252, 253].

To derive latent representations solely from the observed data $\boldsymbol{x}_{1:N}$, where speaker characteristics and content are entangled, we aim to learn a posterior distribution $q(\boldsymbol{z}_0|\boldsymbol{x}_{1:N})$ that disentangles these two components. Specifically, we use variational inference:

$$\begin{aligned} q(\boldsymbol{z}_0|\boldsymbol{x}_{1:N}) &= q(\boldsymbol{c}_{1:N}, \boldsymbol{s}|\boldsymbol{x}_{1:N}) \\ &= q(\boldsymbol{c}_{1:N}|\boldsymbol{x}_{1:N})q(\boldsymbol{s}|\boldsymbol{x}_{1:N}) \\ &= q(\boldsymbol{s}|\boldsymbol{x}_{1:N}) \prod_{i=1}^{N} q(\boldsymbol{c}_i|\boldsymbol{c}_{<i}, \boldsymbol{x}_{1:N}). \end{aligned} \tag{6.7}$$

The speaker latent posterior follows a Gaussian distribution:

$$q\left(\boldsymbol{s}|\boldsymbol{x}_{1:N}\right) = \mathcal{N}\left(\boldsymbol{s}; \boldsymbol{\mu}^s(\boldsymbol{x}_{1:N}), \mathrm{diag}\{(\boldsymbol{\sigma}^s(\boldsymbol{x}_{1:N}))^2\}\right), \tag{6.8}$$

where the mean and standard deviation vectors, $\boldsymbol{\mu}^s(\cdot)$ and $\boldsymbol{\sigma}^s(\cdot)$, are modeled by an ECAPA-TDNN [3] with two linear layers. Similarly, we define:

$$q\left(\boldsymbol{c}_i|\boldsymbol{c}_{<i}, \boldsymbol{x}_{1:N}\right) = \mathcal{N}\left(\boldsymbol{c}_i; \boldsymbol{\mu}_i^c(\boldsymbol{x}_{1:N}, \boldsymbol{c}_{<i}), \mathrm{diag}\{(\boldsymbol{\sigma}_i^c(\boldsymbol{x}_{1:N}, \boldsymbol{c}_{<i}))^2\}\right), \tag{6.9}$$

where $\boldsymbol{\mu}_i^c(\cdot)$ and $\boldsymbol{\sigma}_i^c(\cdot)$ are obtained by passing the inputs $\boldsymbol{c}_{<i}$ and $\boldsymbol{x}_{1:N}$ through bidirectional LSTMs, followed by an RNN and two linear layers. The reparameterization trick is applied to sample $\boldsymbol{s}$ and $\{\boldsymbol{c}_i\}_{i=1}^N$.

Previous studies have introduced similar parameterizations of dynamic variables through recurrent networks [252, 253]. The standard approach for learning latent representa-

tions is to maximize the evidence lower bound (ELBO) [9, 254]:

$$\max_{p,q} \mathbb{E}_{\boldsymbol{x}_{1:N}\sim p_{\mathrm{D}}(\boldsymbol{x}_{1:N})} \left[\underbrace{\mathbb{E}_{q(\boldsymbol{z}_0|\boldsymbol{x}_{1:N})} \log p(\boldsymbol{x}_{1:N}|\boldsymbol{z}_0)}_{\text{reconstruction term}} - \underbrace{\mathrm{KL}[q(\boldsymbol{z}_0|\boldsymbol{x}_{1:N}) \parallel p(\boldsymbol{z}_0)]}_{\text{prior matching term}} \right], \tag{6.10}$$

where $p_{\mathrm{D}}(\boldsymbol{x}_{1:N})$ represents the empirical data distribution and $\mathrm{KL}[\cdot||\cdot]$ denotes Kullback-Leibler (KL) divergence. Under the assumption that $\boldsymbol{s}$ and $\boldsymbol{c}_{1:N}$ are mutually independent in the posterior, the KL-divergence term is simplified as

$$\begin{aligned} &\mathrm{KL}[q(\boldsymbol{z}_0|\boldsymbol{x}_{1:N}) \parallel p(\boldsymbol{z}_0)] = \\ &\mathrm{KL}[q(\boldsymbol{s}|\boldsymbol{x}_{1:N}) \parallel p(\boldsymbol{s})] + \mathrm{KL}[q(\boldsymbol{c}_{1:N}|\boldsymbol{x}_{1:N}) \parallel p(\boldsymbol{c}_{1:N})], \end{aligned} \tag{6.11}$$

where the second term is approximated using sampled trajectories of the dynamic variables $\boldsymbol{c}_{1:N}$.

We define the **d**isentangled **s**equential **v**ariational **a**uto**e**ncoder (**DSVAE**) loss as the negative ELBO of the log-likelihood:

$$\begin{aligned} \mathcal{L}_{DSVAE} = &- \mathbb{E}_{p_{\mathrm{D}}(\boldsymbol{x}_{1:N})} \mathbb{E}_{q(\boldsymbol{z}_0|\boldsymbol{x}_{1:N})} \Big[\log p(\boldsymbol{x}_{1:N}|\boldsymbol{z}_0) \Big] \\ &+ \mathrm{KL}\big[q(\boldsymbol{s}|\boldsymbol{x}_{1:N}) \parallel p(\boldsymbol{s})\big] \\ &+ \mathrm{KL}[q(\boldsymbol{c}_{1:N}|\boldsymbol{x}_{1:N}) \parallel p(\boldsymbol{c}_{1:N})\big]. \end{aligned} \tag{6.12}$$

The first term in Eq. 6.12 corresponds to the reconstruction loss, while the next two terms correspond to the KL divergence between the posterior and prior distributions of the time-variant content embeddings $\{\boldsymbol{c}_i\}_{i=1}^N$ (Eq. 6.7 and 6.9) and the time-invariant speaker embeddings $\boldsymbol{s}$ (Eq. 6.8), respectively. Specifically, the reconstruction loss is computed through the mean squared error (MSE) between the decoder outputs and inputs. The KL divergence terms can be computed analytically as both the priors and posteriors of $\boldsymbol{s}$ and $\{\boldsymbol{c}_i\}_{i=1}^N$ are assumed to be Gaussian distributed [255].

6.2.5 Model Training

To ensure a meaningful condition for the speaker embedding $\boldsymbol{s}$, we optimize the speaker encoder using AAM-Softmax [108]. To train the network, we define the total loss: the AAM-Softmax [108], DDIM loss (Eq. 6.4), and DSVAE loss (Eq. 6.12). The last one can be treated as regularization. The combination can be implemented as follows:

$$\mathcal{L}_{DLDAE} = \mathcal{L}_{AAM\text{-}Softmax} + \mathcal{L}_{DDIM} + \lambda \mathcal{L}_{DSVAE}, \tag{6.13}$$

where λ is a hyperparameter that regulates the impact of sequential disentanglement. During inference, only the speaker encoder is used to extract speaker embeddings.

6.3 Experiments and Results

6.3.1 Implementation Details

We trained our method on VoxCeleb2-dev and evaluated on the VoxCeleb1 [193, 194] datasets for speaker verification. Features were extracted using HuBERT-Large [136] and WavLM-Large [102], enhanced with SpecAugment [256]. The speaker encoder was ECAPA-TDNN [3], and we applied four augmentation types (room impulse, music, noise, babble) with a 0.6 probability each. Training used 3-second utterances and the batch size was 256. We employed AAM-Softmax [108] (margin=0.2, scale=30) and reduced the learning rate by 3% per epoch. Networks were optimized using Adam with CosineAnnealingWarmRestarts [257].

6.3.2 Comparing with Existing Methods

To evaluate the effectiveness of our proposed DLD–AE, we compare its performance with existing disentanglement techniques. As shown in Table 6.1, when using Fbank

Table 6.1: Performance of the baseline models and the proposed DLD–AE on VoxCeleb1-O, VoxCeleb1-E, and VoxCeleb1-H. All experiments used ECAPA-TDNN as the speaker encoder and were trained on VoxCeleb2-dev. Results were obtained without AS-Norm [5, 6] nor quality-aware score calibration [7]. For RecXi, the results are based on the setting RecXi($\tilde{\phi}, \tilde{\phi}_{\text{lin}}$) in [8].

Row	Input Feature	Disentanglement Method	VoxCeleb1-O		VoxCeleb1-E		VoxCeleb1-H	
			EER(%)	minDCF	EER(%)	minDCF	EER(%)	minDCF
1	Fbank	None	1.12	0.145	1.25	0.142	2.43	0.239
2		RecXi + $\mathcal{L}_{\text{ssp}}$ [8]	1.19	0.107	1.29	0.141	2.46	0.227
3		DLD–AE (Ours)	1.01	0.101	1.28	0.153	2.35	0.213
4	HuBERT	None	0.91	0.119	0.99	1.146	2.35	0.252
5		DLD–AE (Ours)	0.88	0.088	**0.91**	1.011	2.05	0.231
6	WavLM	None	0.85	0.113	1.12	0.091	2.06	0.197
7		DLD–AE (Ours)	**0.78**	**0.081**	**0.91**	**0.090**	**1.83**	0.191

features, our DLD–AE (Row 3) outperforms the baseline ECAPA-TDNN (Row 1) and achieves competitive results compared to RecXi (Row 2). This demonstrates the effectiveness of our disentanglement framework in improving speaker verification performance. The results also show that our disentanglement technique is particularly effective when applied to pre-trained features, including HuBERT and WavLM features. For example, with the WavLM features, DLD–AE (Row 7) reduces the EER to 0.78% on VoxCeleb1-O, compared to 0.85% without disentanglement (Row 6). A similar trend is observed for minDCF. This improvement is attributed to our framework's ability to disentangle static speaker components, enhancing speaker recognition effectively. The improvement highlights the importance of modeling the dynamic contents in speech and disentangling the speaker and content representations.

6.3.3 Ablation Study

We conducted ablation experiments to investigate the importance of different components in the proposed DLD–AE. We also conducted experiments using DSVAE

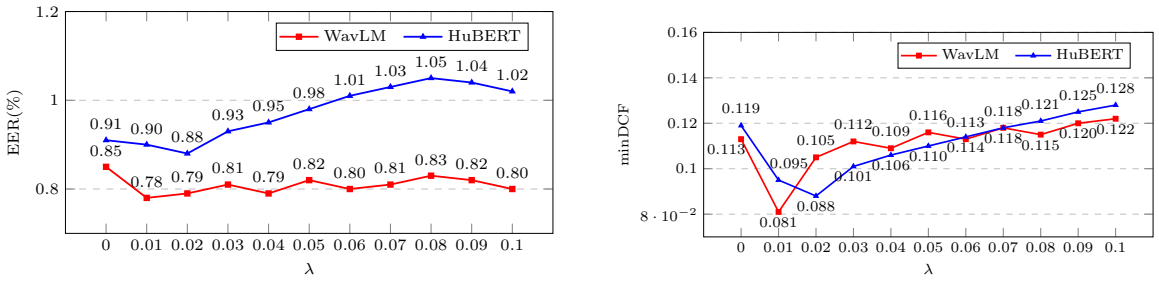
Table 6.2: Ablation study on VoxCeleb1-O. DSVAE [9] incorporates AAM-Softmax.

Row	Input Feature	Disentanglement Method	VoxCeleb1-O	
			EER(%)	minDCF
1	HuBERT	None	0.91	0.119
2		DSVAE [9]	0.90	0.093
3		DLD–AE (w/o condition)	0.89	0.090
4		DLD–AE (Ours)	0.88	0.088
5	WavLM	None	0.85	0.113
6		DSVAE [9]	0.83	0.094
7		DLD–AE (w/o condition)	0.81	0.084
8		DLD–AE (Ours)	0.78	0.081

to perform the disentanglement, which is essentially a VAE-based disentanglement without the diffusion process. Results are shown in Table 6.2. Comparing Row 6 with Row 5 in Table 6.2 reveals that adding the VAE can slightly improve performance. However, a significant performance gain is observed when integrating the diffusion processes into the VAE (Row 7). The best performance is achieved when the diffusion processes are conditioned on the speaker embeddings (Row 8). The same conclusions are obtained regardless of which pre-trained models were used.

6.3.4 Impact of λ

The hyperparameter λ in Eq. 6.13 controls the extent of DLD–AE's contribution in the proposed framework. We analyze the impact of varying λ on SV performance. We selected λ ranging from 0.01 to 0.1, incrementing by 0.01 at each step. The results, shown in Fig. 6.2, indicate that for both EER and minDCF, when WavLM is used as the pre-trained model, the best performance is achieved at $\lambda = 0.01$. For HuBERT, the optimal result is observed at $\lambda = 0.02$, with performance declining as λ increases.



(a) EER(%) (b) minDCF

Figure 6.2: Results on VoxCeleb1-O for different λ in Eq. 6.13, using WavLM Large and HuBERT Large as the PTMs.

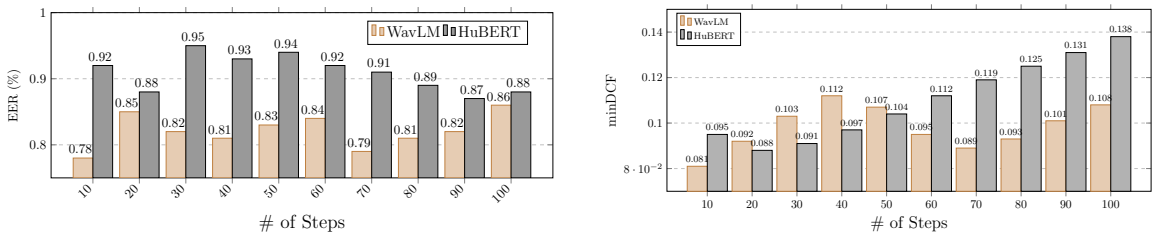
These findings suggest that placing excessive emphasis on sequence decoupling may negatively impact the model's ability to learn discriminative speaker embeddings.

6.3.5 Impact of Diffusion Steps

In our work, we employ DDIM for diffusion and denoising, which substantially reduces the number of steps. Unlike standard DDPM, which often requires hundreds or even thousands of iterative steps, DDIM can generate high-quality samples in just a few dozen steps. This efficiency is achieved through an explicit inference process that reduces the random noise term, making each step more efficient and accurate. As illustrated in Fig. 6.3, the optimal performance is achieved with 10 steps, while for HuBERT, the best results are obtained using 20 steps.

6.4 Conclusions

This paper proposes a sequential disentanglement framework based on a latent diffusion model (DLD–AE) to decouple speaker traits from content factors, leveraging only speaker traits for speaker verification. Using WavLM and HuBERT as pre-trained models to extract frame-level features and the latent diffusion model for speaker-



(a) EER(%) (b) minDCF

Figure 6.3: Impact of diffusion steps on VoxCeleb1-O using WavLM-Large and HuBERT as pre-trained models.

content disentanglement, our method achieves the best performance on the VoxCeleb1 test set. Experimental results demonstrate the effectiveness of incorporating sequential disentanglement with pre-trained models for extracting discriminative speaker embeddings.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

In this thesis, we addressed several critical challenges in speaker representation learning and proposed innovative solutions to improve the robustness, efficiency, and generalizability of speaker embedding networks for verification. Our contributions include:

- A supervised contrastive learning framework that integrates an additive angular margin and maximizes the mutual information between acoustic features and speaker representations. This approach effectively reduces intra-class variation and enhances inter-speaker separability.

- Parameter-efficient fine-tuning strategies for pre-trained Transformer models. By using dynamic prompt tuning and incorporating spectral information into a LoRA-based adaptation process, our approach efficiently extracts task-relevant features while reducing computational and storage overhead.

- A diffusion-based method within a variational autoencoder framework was designed to disentangle speaker timbre from spoken content. By leveraging a

conditional diffusion model in the latent space, our method produces content-invariant speaker embeddings that are resilient to language mismatches.

Extensive experiments on VoxCeleb, CN-Celeb, and CU-MARVEL benchmarks validate that these contributions significantly improve speaker verification performance by addressing the limitations of softmax-based classification and the challenges associated with directly applying pre-trained speech models.

7.2 Future Works in Speaker Representation Learning

Speaker-specific features are speech characteristics inherently tied to an individual's physiology and speaking habits. Such features often provide speaker-discriminative cues, including formant distributions, spectral–temporal patterns, and utterance-level statistics (e.g., average pitch, pitch range, and temporal energy profiles). These features are relatively stable across sessions and content. Speaker-specific features are those that not only reflect personal traits but also maximize inter-speaker variability while minimizing intra-speaker variation. Deep neural networks can capture speaker-specific features through frame-level embedding that preserve fine-grained timbre and articulation cues and utterance-level embeddings that aggregate prosodic and stylistic patterns.

While existing representation-learning models can reliably capture many robust acoustic–spectral cues (e.g., spectral envelope, formant structure, harmonic patterns), they often fail to preserve more subtle, context-dependent features. For example, fine-grained voice-quality measures such as jitter, shimmer, and breathiness are frequently underrepresented [258]. Similarly, speaker-specific prosodic tendencies—such as characteristic intonation patterns or microl-level articulation habits—are often entangled with linguistic content in speech and may be suppressed by content-dominant repre-

sentations [164]. Existing models also struggle to disentangle speaker identity from language- or phoneme-dependent variations, leading to reduced robustness in cross-lingual or cross-content scenarios [259, 260].

Despite the advances presented in this work, several challenges remain that open promising avenues for future research. They are briefly outlined below.

Explainability. While deep neural network-based speaker representation models achieve strong performance, their decision processes remain largely opaque. Future work should focus on enhancing the interpretability of learned embeddings by investigating what speaker-specific cues are actually captured in the latent space. For instance, experiments could vary the jitter and shimmer in the input speech and observe how such perturbations influence inter- and intra-speaker variability in the latent space. Such studies will provide empirical evidence on which types of acoustic-prosodic traits are robustly encoded and which types could be easily lost.

Leveraging Large-Scale Pre-training. Pre-trained speech models offer robust feature representations, yet their direct application to speaker verification is limited by the dual nature of speech signals—carrying both speaker identity and linguistic content. Future research should develop self-supervised representation learning methods specifically tailored to speaker verification. Moreover, while the model is developed for speaker verification, an interesting future direction would be to investigate whether disentangling speaker timbre from spoken content also benefits ASR tasks. We leave such evaluations for future work.

Cross-Modality Learning. Recent advances in large language models have opened up opportunities for cross-modal approaches. Investigating methods that translate speaker representations into textual descriptions and vice versa can enhance interpretability and adaptability. This direction promises to bridge the gap between speaker characteristics and natural languages, potentially leading to more explainable systems.

Privacy Protection and Ethical Issues. Since speaker data inherently includes sensitive personal information, safeguarding privacy is paramount. Future studies must refine techniques such as speaker anonymization to protect individual privacy without compromising the utility of the data for downstream tasks.

Security: Attacks and Defenses. The advent of advanced voice synthesis technologies raises significant security concerns for speaker verification systems. Developing robust methods to distinguish between genuine and synthesized voices and effective defenses against adversarial attacks are essential to maintaining system integrity.

References

[1] Z. Li, M.-W. Mak, and H. Meng, "Dual parameter-efficient fine-tuning for speaker representation via speaker prompt tuning and adapters," in *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 10 751–10 755.

[2] Z. Li, M.-W. Mak, H.-y. Lee, and H. Meng, "Parameter-efficient fine-tuning of speaker-aware dynamic prompts for speaker verification," in *Proc. Interspeech 2024*, 2024, pp. 2675–2679.

[3] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. of Interspeech*, 2020, pp. 3830–3834.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. of International Conference on Learning Representations (ICLR)*, 2014.

[5] Z. N. Karam, W. M. Campbell, and N. Dehak, "Towards reduced false-alarms using cohorts," in *Proc. of IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2011, pp. 4512–4515.

[6] S. Cumani, P. D. Batzu, D. Colibro, C. Vair, P. Laface, V. Vasilakakis *et al.*, "Comparison of speaker recognition approaches for real applications." in *Proc. of Interspeech*, 2011, pp. 2365–2368.

[7] J. Thienpondt, B. Desplanques, and K. Demuynck, "The IDLab VoxSRC-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification," in *Prof. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5814–5818.

[8] T. Liu, K. A. Lee, Q. Wang, and H. Li, "Disentangling voice and content with self-supervision for speaker recognition," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, pp. 50 221–50 236, 2023.

[9] L. Yingzhen and S. Mandt, "Disentangled sequential autoencoder," in *Proc. of International Conference on Machine Learning (ICML)*, 2018, pp. 5670–5679.

[10] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," *Neural Networks*, vol. 140, pp. 65–99, 2021.

[11] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[12] Z. Saquib, N. Salam, R. P. Nair, N. Pandey, and A. Joshi, "A survey on automatic speaker recognition systems," in *International Conference on Multimedia, Computer graphics, and Broadcasting*. Springer, 2010, pp. 134–145.

[13] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, 2010.

[14] S. Furui, "40 years of progress in automatic speaker recognition," in *Advances in Biometrics: Third International Conference, ICB 2009, Alghero, Italy, June 2-5, 2009. Proceedings 3*. Springer, 2009, pp. 1050–1059.

[15] N. Singh, A. Agrawal, and R. Khan, "Voice biometric: A technology for voice based authentication," *Advanced Science, Engineering and Medicine*, vol. 10, no. 7-8, pp. 754–759, 2018.

[16] E. Kiktova and J. Juhar, "Speaker recognition for surveillance application," *Journal of Electrical and Electronics Engineering*, vol. 8, no. 2, p. 19, 2015.

[17] I. McGraw, R. Prabhavalkar, R. Alvarez, M. G. Arenas, K. Rao, D. Rybach, O. Alsharif, H. Sak, A. Gruenstein, F. Beaufays *et al.*, "Personalized speech recognition on mobile devices," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5955–5959.

[18] J. P. Campbell, W. Shen, W. M. Campbell, R. Schwartz, J.-F. Bonastre, and D. Matrouf, "Forensic speaker recognition," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 95–103, 2009.

[19] M. Kotti, V. Moschou, and C. Kotropoulos, "Speaker segmentation and clustering," *Signal Processing*, vol. 88, no. 5, pp. 1091–1124, 2008.

[20] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization: Recent advances with deep learning," *Computer Speech & Language*, vol. 72, p. 101317, 2022.

[21] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[22] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. Lopez Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[23] Y. Liu, C. Yu, W. Shuai, Z. Yang, Y. Chao, and W. Zhang, "Non-parallel any-to-many voice conversion by replacing speaker statistics," *Proc. of Interspeech*, pp. 1369–1373, 2021.

[24] F. Fang, X. Wang, J. Yamagishi, I. Echizen, M. Todisco, N. Evans, and J.-F. Bonastre, "Speaker anonymization using x-vector and neural waveform models," *arXiv preprint arXiv:1905.13561*, 2019.

[25] N. Tomashenko, X. Miao, P. Champion, S. Meyer, X. Wang, E. Vincent, M. Panariello, N. Evans, J. Yamagishi, and M. Todisco, "The voiceprivacy 2024 challenge evaluation plan," *arXiv preprint arXiv:2404.02677*, 2024.

[26] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[27] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052–4056.

[28] N. Chen, Y. Qian, and K. Yu, "Multi-task learning for text-dependent speaker verification," in *16th Annual Conference of the International Speech Communication Association*, 2015.

[29] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 165–170.

[30] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[31] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5115–5119.

[32] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[33] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification." in *Proc. of Interspeech*, 2017, pp. 999–1003.

[34] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.

[35] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "But system description to voxceleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[36] H. Wang, S. Zheng, Y. Chen, L. Cheng, and Q. Chen, "CAM++: A fast and efficient network for speaker verification using context-aware masking," in *Proc. of Interspeech*, 2023, pp. 5301–5305.

[37] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Proc. of Odyssey: The Speaker and Language Recognition Workshop*, 2018, pp. 74–81.

[38] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," in *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2019, pp. 1652–1656.

[39] J. S. Chung, J. Huh, S. Mun, M. Lee, H.-S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," *Proc. of Interspeech*, pp. 2977–2981, 2020.

[40] S. Wang, Z. Huang, Y. Qian, and K. Yu, "Discriminative neural embedding learning for short-duration text-independent speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1686–1696, 2019.

[41] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification." in *Proc. of Interspeech*, 2018, pp. 3573–3577.

[42] S. Wang, Y. Yang, Y. Qian, and K. Yu, "Revisiting the statistics pooling layer in deep speaker embedding learning," in *Proc. of International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2021.

[43] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5791–5795.

[44] W. Lin, M.-W. Mak, N. Li, D. Su, and D. Yu, "A framework for adapting dnn speaker embedding across languages," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2810–2822, 2020.

[45] W. Lin and M.-W. Mak, "Robust speaker verification using deep weight space ensemble," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 802–812, 2023.

[46] ——, "Model-agnostic meta-learning for fast text-dependent speaker embedding adaptation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1866–1876, 2023.

[47] Y. Zhang, Z. Lv, H. Wu, S. Zhang, P. Hu, Z. Wu, H.-y. Lee, and H. Meng, "MFA-Conformer: Multi-scale feature aggregation conformer for automatic speaker verification," *arXiv preprint arXiv:2203.15249*, 2022.

[48] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[49] Y. Liu, L. He, W. Liu, and J. Liu, "Exploring a unified attention-based pooling framework for speaker verification," in *Proc. of International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2018, pp. 200–204.

[50] Q. Wang, K. Okabe, K. A. Lee, H. Yamamoto, and T. Koshinaka, "Attention mechanism in speaker recognition: What does it learn in deep speaker embedding?" in *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 1052–1059.

[51] S. Wang, H. Dinkel, Y. Qian, and K. Yu, "Covariance based deep feature for text-dependent speaker verification," in *International Conference on Intelligent Science and Big Data Engineering*. Springer, 2018, pp. 231–242.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[53] N. Li, D. Tuo, D. Su, Z. Li, and D. Yu, "Deep discriminative embeddings for duration robust speaker verification," in *Proc. of Interspeech*, 2018, pp. 2262–2266.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Neural Information Processing Systems,Neural Information Processing Systems*, Jun 2017.

[55] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[56] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[57] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[58] N. J. M. S. Mary, S. Umesh, and S. V. Katta, "S-vectors and tesa: Speaker embeddings and a speaker authenticator based on transformer encoder," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 404–413, 2021.

[59] P. Safari, M. India, and J. Hernando, "Self-attention encoding and pooling for speaker recognition," *arXiv preprint arXiv:2008.01077*, 2020.

[60] B. Han, Z. Chen, and Y. Qian, "Local information modeling with self-attention for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6727–6731.

[61] R. Wang, J. Ao, L. Zhou, S. Liu, Z. Wei, T. Ko, Q. Li, and Y. Zhang, "Multi-view self-attention based transformer for speaker recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6732–6736.

[62] J.-H. Choi, J.-Y. Yang, Y.-R. Jeoung, and J.-H. Chang, "Improved cnn-transformer using broadcasted residual learning for text-independent speaker verification." in *Interspeech*, 2022, pp. 2223–2227.

[63] H. Wang, X. Lin, and J. Zhang, "A lightweight cnn-conformer model for automatic speaker verification," *IEEE Signal Processing Letters*, 2023.

[64] M. Sang, Y. Zhao, G. Liu, J. H. Hansen, and J. Wu, "Improving transformer-based networks with locality for automatic speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[65] J. Yao, C. Liang, Z. Peng, B. Zhang, and X.-L. Zhang, "Branch-ecapa-tdnn: A parallel branch architecture to capture local and global features for speaker verification," in *Proc. of Interspeech*, 2023, pp. 1943–1947.

[66] X. Wang, F. Wang, B. Xu, L. Xu, and J. Xiao, "P-vectors: A parallel-coupled tdnn/transformer network for speaker verification," *arXiv preprint arXiv:2305.14778*, 2023.

[67] Y. Sun, C. Li, and B. Li, "Branchformer-based tdnn for automatic speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 10 981–10 985.

[68] N. Tawara, A. Ogawa, T. Iwata, M. Delcroix, and T. Ogawa, "Frame-level phoneme-invariant speaker embedding for text-independent speaker recognition on extremely short utterances," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6799–6803.

[69] S. Wang and J. Rohdin, "On the usage of phonetic information for text-independent speaker embedding extraction." in *Proc. of Interspeech*, 2019.

[70] S. Maiti, Y. Ueda, S. Watanabe, C. Zhang, M. Yu, S.-X. Zhang, and Y. Xu, "Eend-ss: Joint end-to-end neural speaker diarization and speech separation for flexible number of speakers," in *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 480–487.

[71] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny, A. Laptev, and A. Romanenko, "Target-Speaker Voice Activity Detection: A Novel Approach for Multi-Speaker Diarization in a Dinner Party Scenario," in *Proc. of Interspeech*, 2020, pp. 274–278.

[72] J.-w. Jung, H.-S. Heo, J.-h. Kim, H.-j. Shim, and H.-J. Yu, "Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," *arXiv preprint arXiv:1904.08104*, 2019.

[73] J.-w. Jung, S.-b. Kim, H.-j. Shim, J.-h. Kim, and H.-J. Yu, "Improved rawnet with feature map scaling for text-independent speaker verification using raw waveforms," *Proc. of Interspeech*, pp. 3583–3587, 2020.

[74] J.-w. Jung, Y. J. Kim, H.-S. Heo, B.-J. Lee, Y. Kwon, and J. S. Chung, "Pushing the limits of raw waveform speaker recognition," *Proc. of Interspeech*, 2022.

[75] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 1021–1028.

[76] T. Zhou, Y. Zhao, and J. Wu, "Resnext and res2net structures for speaker verification," in *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 301–307.

[77] J. Thienpondt and K. Demuynck, "Ecapa2: A hybrid neural network architecture and training strategy for robust speaker embeddings," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–8.

[78] J. Thienpondt, B. Desplanques, and K. Demuynck, "Integrating frequency translational invariance in tdnns and frequency positional information in 2d resnets to enhance speaker verification," *arXiv preprint arXiv:2104.02370*, 2021.

[79] T. Liu, R. K. Das, K. A. Lee, and H. Li, "Mfa: Tdnn with multi-scale frequency-channel attention for text-independent speaker verification with short utterances," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 7517–7521.

[80] Y.-Q. Yu and W.-J. Li, "Densely connected time delay neural network for speaker verification." in *Interspeech*, 2020, pp. 921–925.

[81] G. Zhu, F. Jiang, and Z. Duan, "Y-vector: Multiscale waveform encoder for speaker embedding," *arXiv preprint arXiv:2010.12951*, 2020.

[82] S. H. Mun, J.-w. Jung, M. H. Han, and N. S. Kim, "Frequency and multi-scale selective kernel attention for speaker verification," in *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 548–554.

[83] M. K. Roy and U. Keshwala, "Res2net based text independent speaker recognition system," in *Proc. of 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2022, pp. 612–616.

[84] H.-J. Heo, U.-H. Shin, R. Lee, Y. Cheon, and H.-M. Park, "Next-tdnn: Modernizing multi-scale temporal convolution backbone for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 11 186–11 190.

[85] X. Liu, D. Chen, X. Wang, S. Xiang, and X. Zhou, "Rep-mca-former: An efficient multi-scale convolution attention encoder for text-independent speaker verification," *Computer Speech & Language*, vol. 85, p. 101600, 2024.

[86] Z. Li, C. Fang, R. Xiao, W. Wang, and Y. Yan, "Si-net: Multi-scale context-aware convolutional block for speaker verification," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 220–227.

[87] B. Han, Z. Chen, B. Liu, and Y. Qian, "Mlp-svnet: A multi-layer perceptrons based network for speaker verification," in *Proc. of IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 7522–7526.

[88] Y. Zi and S. Xiong, "Resformer: Local frame-level feature and global segment-level feature joint learning for speaker verification," *Circuits, Systems, and Signal Processing*, pp. 1–20, 2024.

[89] Y. Li, J. Gan, X. Lin, Y. Qiu, H. Zhan, and H. Tian, "Ds-tdnn: Dual-stream time-delay neural network with global-aware filter for speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[90] X. Lu and J. Dang, "An investigation of dependencies between frequency components and speaker characteristics for text-independent speaker identification," *Speech Communication*, vol. 50, no. 4, pp. 312–322, 2008.

[91] A. Deng, S. Wang, W. Kang, and F. Deng, "On the importance of different frequency bins for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7537–7541.

[92] B. Gu, J. Zhang, and W. Guo, "A dynamic convolution framework for session-independent speaker embedding learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[93] Y. Shi, Q. Huang, and T. Hain, "Robust speaker recognition using speech enhancement and attention model," *arXiv preprint arXiv:2001.05031*, 2020.

[94] S. Kataria, P. S. Nidadavolu, J. Villalba, N. Chen, P. Garcia-Perera, and N. Dehak, "Feature enhancement with deep feature losses for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7584–7588.

[95] A. Hajavi and A. Etemad, "Knowing what to listen to: Early attention for deep speech representation learning," *arXiv preprint arXiv:2009.01822*, vol. 1, 2020.

[96] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. International Conference on Machine Learning (ICML)*, 2020, pp. 1597–1607.

[97] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2020, pp. 9729–9738.

[98] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9650–9660.

[99] D. Cai, W. Wang, and M. Li, "An iterative framework for self-supervised deep speaker representation learning," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6728–6732.

[100] B. Han, Z. Chen, and Y. Qian, "Self-supervised speaker verification using dynamic loss-gate and label correction," in *Proc. of Interspeech*, 2022, pp. 4780–4784.

[101] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *ArXiv*, vol. abs/1910.05453, 2019.

[102] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[103] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. International Conference on Machine Learning (ICML)*, 2023, pp. 28 492–28 518.

[104] Y. Li, F. Gao, Z. Ou, and J. Sun, "Angular softmax loss for end-to-end speaker verification," in *Proc. of International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2018, pp. 190–194.

[105] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification." in *Interspeech*, 2018, pp. 3623–3627.

[106] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[107] Y.-Q. Yu, L. Fan, and W.-J. Li, "Ensemble additive margin softmax for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6046–6050.

[108] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4690–4699.

[109] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[110] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances." in *Interspeech*, 2017, pp. 1487–1491.

[111] C. Zhang, K. Koishida, and J. H. Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1633–1644, 2018.

[112] Z. Huang, S. Wang, and Y. Qian, "Joint i-vector with end-to-end system for short duration text-independent speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4869–4873.

[113] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: a deep quadruplet network for person re-identification," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 403–412.

[114] V. S. Narayanaswamy, J. J. Thiagarajan, H. Song, and A. Spanias, "Designing an effective metric learning pipeline for speaker diarization," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5806–5810.

[115] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. European Conference on Computer Vision (ECCV)*, 2016, pp. 499–515.

[116] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[117] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2536–2544.

[118] A. Jati and P. Georgiou, "Neural predictive coding using convolutional neural networks toward unsupervised learning of speaker characteristics," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 10, pp. 1577–1589, 2019.

[119] M. Ravanelli and Y. Bengio, "Learning speaker representations with mutual information," in *Proc. of Interspeech*, 2019, pp. 1153–1157.

[120] H. Zhang, Y. Zou, and H. Wang, "Contrastive self-supervised learning for text-independent speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 6713–6717.

[121] W. Xia, C. Zhang, C. Weng, M. Yu, and D. Yu, "Self-supervised text-independent speaker verification using prototypical momentum contrastive

learning," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 6723–6727.

[122] B. Han, Z. Chen, and Y. Qian, "Self-supervised learning with cluster-aware-dino for high-performance robust speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 529–541, 2023.

[123] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. of Interspeech*, 2018, pp. 1086–1090.

[124] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9650–9660.

[125] J. weon Jung, Y. Kim, H.-S. Heo, B.-J. Lee, Y. Kwon, and J. S. Chung, "Pushing the limits of raw waveform speaker recognition," in *Proc. of Interspeech*, 2022, pp. 2228–2232.

[126] H.-S. Heo, J.-w. Jung, J. Kang, Y. Kwon, Y. J. Kim, and B. abd JS Chung, "Self-supervised curriculum learning for speaker verification," *arXiv preprint arXiv:2203.14525*, 2022.

[127] Z. Chen, Y. Qian, B. Han, Y. Qian, and M. Zeng, "A comprehensive study on self-supervised distillation for speaker representation learning," in *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 599–604.

[128] S. Zheng, G. Liu, H. Suo, and Y. Lei, "Autoencoder-Based Semi-Supervised Curriculum Learning for Out-of-Domain Speaker Verification," in *Proc. of Interspeech*, 2019, pp. 4360–4364.

[129] X. Qin, M. Li, H. Bu, S. Narayanan, and H. Li, "The 2022 far-field speaker verification challenge: Exploring domain mismatch and semi-supervised learning under the far-field scenario," *arXiv preprint arXiv:2209.05273*, 2022.

[130] Z. Li, Y. Lin, N. Jiang, X. Qin, G. Zhao, H. Wu, and M. Li, "Multi-objective progressive clustering for semi-supervised domain adaptation in speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 12 236–12 240.

[131] N. Inoue and K. Goto, "Semi-supervised contrastive learning with generalized contrastive loss and its application to speaker recognition," in *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2020, pp. 1641–1646.

[132] J.-H. Choi, J. Kyung, J.-S. Seong, Y.-R. Jeoung, and J.-H. Chang, "Extending self-distilled self-supervised learning for semi-supervised speaker verification," in *Proc. of Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–8.

[133] L. Chen, V. Ravichandran, and A. Stolcke, "Graph-Based Label Propagation for Semi-Supervised Speaker Identification," in *Proc. of Interspeech*, 2021, pp. 4588–4592.

[134] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[135] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 12 449–12 460, 2020.

[136] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[137] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *International Conference on Machine Learning (ICML)*, 2022, pp. 1298–1312.

[138] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. of Interspeech*, 2021, pp. 1194–1198.

[139] Z. Fan, M. Li, S. Zhou, and B. Xu, "Exploring wav2vec 2.0 on Speaker Verification and Language Identification," in *Proc. of Interspeech*, 2021, pp. 1509–1513.

[140] N. Vaessen and D. A. Van Leeuwen, "Fine-tuning wav2vec2 for speaker recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7967–7971.

[141] Z. Chen, S. Chen, Y. Wu, Y. Qian, C. Wang, S. Liu, Y. Qian, and M. Zeng, "Large-scale self-supervised speech representation learning for automatic speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6147–6151.

[142] J. Peng, T. Stafylakis, R. Gu, O. Plchot, L. Mošner, L. Burget, and J. Černockỳ, "Parameter-efficient transfer learning of pre-trained transformer models for speaker verification using adapters," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[143] D. Cai, W. Wang, M. Li, R. Xia, and C. Huang, "Pretraining conformer with asr for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[144] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 4879–4883.

[145] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5265–5274.

[146] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech 2018*, 2018, pp. 2252–2256.

[147] M. Sang, W. Xia, and J. H. Hansen, "Open-set short utterance forensic speaker verification using teacher-student network with explicit inductive bias," *Proc. of Interspeech*, pp. 2262–2266, 2020.

[148] G. Bhattacharya, J. Monteiro, J. Alam, and P. Kenny, "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6226–6230.

[149] M. Sang, W. Xia, and J. H. Hansen, "DEAAN: Disentangled embedding and adversarial adaptation network for robust speaker representation learning," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6169–6173.

[150] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[151] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," in *Proc. 3rd International Conference on Learning Representations, (ICLR)*, 2015.

[152] G. Elsayed, D. Krishnan, H. Mobahi, K. Regan, and S. Bengio, "Large margin deep networks for classification," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[153] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. 33rd International Conference on Machine Learning (ICML)*, 2016, pp. 507–516.

[154] W.-W. Lin, M.-W. Mak, and J.-T. Chien, "Multisource i-vectors domain adaptation using maximum mean discrepancy based autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 12, pp. 2412–2422, 2018.

[155] L. Li, M.-W. Mak, and J.-T. Chien, "Contrastive adversarial domain adaptation networks for speaker recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, pp. 2236–2245, 2022.

[156] Y. Tu, M.-W. Mak, and J.-T. Chien, "Variational domain adversarial learning with mutual information maximization for speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2013–2024, 2020.

[157] ——, "Information maximized variational domain adversarial learning for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6449–6453.

[158] D. Cai, W. Cai, and M. Li, "Within-sample variability-invariant loss for robust speaker recognition under noisy environments," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6469–6473.

[159] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," *arXiv preprint arXiv:1703.09507*, 2017.

[160] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "NormFace: L2 hypersphere embedding for face verification," in *Proc. of 25th ACM International Conference on Multimedia*, 2017, pp. 1041–1049.

[161] Y. Liu, H. Li, and X. Wang, "Rethinking feature discrimination and polymerization for large-scale recognition," *arXiv preprint arXiv:1710.00870*, 2017.

[162] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 212–220.

[163] Y. Zhang, H. Zhu, Y. Wang, N. Xu, X. Li, and B. Zhao, "A contrastive framework for learning sentence representations from pairwise and triple-wise perspective in angular space," in *Proc. of 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022, pp. 4892–4903.

[164] Y. Tu, M.-W. Mak, and J.-T. Chien, "Contrastive self-supervised speaker embedding with sequential disentanglement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[165] C.-X. Gan, M.-W. Mak, W. Lin, and J.-T. Chien, "Asymmetric clean segments-guided self-supervised learning for robust speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 11 081–11 085.

[166] H. Wang, X. Guo, Z.-H. Deng, and Y. Lu, "Rethinking minimal sufficient representation in contrastive learning," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 041–16 050.

[167] Y. Tu and M.-W. Mak, "Mutual information enhanced training for speaker embedding." in *Prof. of Interspeech*, 2021, pp. 91–95.

[168] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, and W. Xu, "Consert: A contrastive framework for self-supervised sentence representation transfer," *arXiv preprint arXiv:2105.11741*, 2021.

[169] J. Giorgi, O. Nitski, B. Wang, and G. Bader, "DeCLUTR: Deep contrastive learning for unsupervised textual representations," in *Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021, pp. 879–895.

[170] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proc. Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 6894–6910.

[171] L. Li, R. Nai, and D. Wang, "Real additive margin softmax for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7527–7531.

[172] Z. Chen, S. Wang, and Y. Qian, "Self-supervised learning based domain adaptation for robust speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 5834–5838.

[173] M. Sang, H. Li, F. Liu, A. O. Arnold, and L. Wan, "Self-supervised speaker verification with simple siamese network and self-supervised regularization," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 6127–6131.

[174] Z. Li and M.-W. Mak, "Speaker representation learning via contrastive loss with maximal speaker separability," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2022, pp. 962–967.

[175] Z. Li, M.-W. Mak, and H. M.-L. Meng, "Discriminative speaker representation via contrastive learning with class-aware attention in angular space," in *Proc.*

of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023.

[176] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

[177] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[178] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[179] W. Wang, D. Cai, X. Qin, and M. Li, "The dku-dukeece systems for voxceleb speaker recognition challenge 2020," *arXiv preprint arXiv:2010.12731*, 2020.

[180] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 18 661–18 673, 2020.

[181] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proc. International Conference on Machine Learning (ICML)*, 2020, pp. 9929–9939.

[182] M. Federici, A. Dutta, P. Forré, N. Kushman, and Z. Akata, "Learning robust representations via multi-view information bottleneck," in *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[183] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[184] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker, "On variational bounds of mutual information," in *Proc. of International Conference on Machine Learning (ICML)*, 2019, pp. 5171–5180.

[185] T. Lepage and R. Dehak, "Label-efficient self-supervised speaker verification with information maximization and contrastive learning," in *Proc. of InterSpeech*, 2022, pp. 4018–4022.

[186] C. Zhang and D. Yu, "C3-dino: Joint contrastive and non-contrastive self-supervised learning for speaker verification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1273–1283, 2022.

[187] Z. Li, W. Guo, B. Gu, S. Peng, and J. Zhang, "Contrastive learning and inter-speaker distribution alignment based unsupervised domain adaptation for robust speaker verification," in *Proc. of InterSpeech*, 2024.

[188] S. H. Mun, M. H. Han, M. Kim, D. Lee, and N. S. Kim, "Disentangled speaker representation learning via mutual information minimization," in *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2022, pp. 89–96.

[189] W. H. Kang, J. Alam, and A. Fathan, "Domain generalized speaker embedding learning via mutual information minimization." in *Odyssey*, 2022, pp. 178–184.

[190] F. Zhang, W. Zhou, Y. Liu, W. Geng, Y. Shan, and C. Zhang, "Disentangling age and identity with a mutual information minimization for cross-age speaker verification," in *Proc. of InterSpeech*, 2024, pp. 3789–3793.

[191] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. of International Conference on Learning Representations (ICLR)*, 2014.

[192] Y. Chen, S. Zheng, H. Wang, L. Cheng, T. Zhu, R. Huang, C. Deng, Q. Chen, S. Zhang, W. Wang *et al.*, "3d-speaker-toolkit: An open-source toolkit for multimodal speaker verification and diarization," in *Proc. of IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), 2025, pp. 1–5.

[193] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. of Interspeech*, 2017, pp. 2616–2620.

[194] J. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *Proc. of InterSpeech*, 2018.

[195] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, "CN-Celeb: A challenging chinese speaker recognition dataset," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7604–7608.

[196] L. Li, R. Liu, J. Kang, Y. Fan, H. Cui, Y. Cai, R. Vipperla, T. F. Zheng, and D. Wang, "CN-Celeb: multi-genre speaker recognition," *Speech Communication*, vol. 137, pp. 77–91, 2022.

[197] H. Meng, B. Mak, M.-W. Mak, H. Fung, X. Gong, T. Kwok, X. Liu, V. Mok, P. Wong, J. Woo *et al.*, "Integrated and enhanced pipeline system to support spoken language analytics for screening neurocognitive disorders," in *Proc. of Interspeech*, 2023, pp. 1713–1717.

[198] Y. Chen, S. Zheng, H. Wang, L. Cheng, Q. Chen, S. Zhang, and J. Li, "ERes2NetV2: Boosting short-duration speaker verification performance with computational efficiency," in *Proc. of Interspeech*, 2024, pp. 3245–3249.

[199] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2019.

[200] Y. Chen, S. Zheng, H. Wang, L. Cheng, Q. Chen, and J. Qi, "An enhanced res2net with local and global feature fusion for speaker verification," in *Proc. of Interspeech*, 2023, pp. 2228–2232.

[201] H. Choi, A. Som, and P. Turaga, "AMC-Loss: Angular margin contrastive loss for improved explainability in image classification," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 838–839.

[202] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[203] Z. Chen, B. Han, X. Xiang, H. Huang, B. Liu, and Y. Qian, "Sjtu-aispeech system for voxceleb speaker recognition challenge 2022," *arXiv preprint arXiv:2209.09076*, 2022.

[204] Z. Bai, J. Wang, X.-L. Zhang, and J. Chen, "End-to-end speaker verification via curriculum bipartite ranking weighted binary cross-entropy," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1330–1344, 2022.

[205] B. Liu and Y. Qian, "Ecapa++: Fine-grained deep embedding learning for tdnn based speaker verification," in *Proc. of Interspeech*, 2023, pp. 3132–3136.

[206] S.-H. Liou, P.-C. Chan, C.-P. Chen, T.-C. Lin, C.-L. Lu, Y.-H. Cheng, H.-F. Chuang, and W.-Y. Chen, "Enhancing ecapa-tdnn with feature processing module and attention mechanism for speaker verification," in *Proc. of Interspeech*, 2024, pp. 2120–2124.

[207] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proc. International Conference on Machine Learning (ICML)*, 2019, pp. 2790–2799.

[208] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. of the 59th Annual Meeting of the Association for Compu-*

tational Linguistics and the 11th International Joint Conference on Natural Language Processing, Aug. 2021, pp. 4582–4597.

[209] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2021.

[210] K.-W. Chang, W.-C. Tseng, S.-W. Li, and H. yi Lee, "An exploration of prompt tuning on generative spoken language model for speech processing tasks," in *Proc. of Interspeech*, 2022, pp. 5005–5009.

[211] K.-W. Chang, Y.-K. Wang, H. Shen, I.-t. Kang, W.-C. Tseng, S.-W. Li, and H.-y. Lee, "SpeechPrompt v2: Prompt tuning for speech classification tasks," *arXiv preprint arXiv:2303.00733*, 2023.

[212] C.-H. H. Yang, B. Li, Y. Zhang, N. Chen, R. Prabhavalkar, T. N. Sainath, and T. Strohman, "From english to more languages: Parameter-efficient model reprogramming for cross-lingual speech recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[213] P. Peng, B. Yan, S. Watanabe, and D. Harwath, "Prompting the hidden talent of web-scale speech models for zero-shot task generalization," in *Proc. of Interspeech*, 2023, pp. 396–400.

[214] Z. Guo, Y. Leng, Y. Wu, S. Zhao, and X. Tan, "PromptTTS: Controllable text-to-speech with text descriptions," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[215] H. Gao, J. Ni, K. Qian, Y. Zhang, S. Chang, and M. Hasegawa-Johnson, "WAVPrompt: towards few-shot spoken language understanding with frozen language models," in *Proc. of Interspeech*, vol. 2022, 2022, pp. 2738–2742.

[216] H. Yang, J. Lin, A. Yang, P. Wang, C. Zhou, and H. Yang, "Prompt tuning for generative multimodal pretrained models," *CoRR*, vol. abs/2208.02532, 2022.

[217] X. Liu, K. Ji, Y. Fu, W. Tam, Z. Du, Z. Yang, and J. Tang, "P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks," in *Proc. of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022, pp. 61–68.

[218] B. Yuan, S. You, and B.-K. Bao, "Self-PT: Adaptive self-prompt tuning for low-resource visual question answering," in *Proc. of the 31st ACM International Conference on Multimedia*, 2023, pp. 5089–5098.

[219] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Conditional prompt learning for vision-language models," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 816–16 825.

[220] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy *et al.*, "DualPrompt: Complementary prompting for rehearsal-free continual learning," in *Proc. European Conference on Computer Vision*, 2022, pp. 631–648.

[221] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 139–149.

[222] M. K. Nandwana, J. Van Hout, M. McLaren, C. Richey, A. Lawson, and M. A. Barrios, "The voices from a distance challenge 2019 evaluation plan," *arXiv preprint arXiv:1902.10828*, 2019.

[223] M. Sang and J. H. Hansen, "Efficient adapter tuning of pre-trained speech models for automatic speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 12 131–12 135.

[224] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *Proc. of the International Conference on Learning Representations (ICML)*, 2023.

[225] Q. Zhang, M. Chen, A. Bukharin, P. He, Y. Cheng, W. Chen, and T. Zhao, "Adaptive budget allocation for parameter-efficient fine-tuning," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2023.

[226] F. Zhang, L. Li, J. Chen, Z. Jiang, B. Wang, and Y. Qian, "IncreLoRA: Incremental parameter allocation method for parameter-efficient fine-tuning," *arXiv preprint arXiv:2308.12043*, 2023.

[227] M. Valipour, M. Rezagholizadeh, I. Kobyzev, and A. Ghodsi, "DyLoRA: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation," in *Proc. of Conference of the European Chapter of the Association for Computational Linguistics*, 2023, pp. 3274–3287.

[228] S.-Y. Liu, C.-Y. Wang, H. Yin, P. Molchanov, Y.-C. F. Wang, K.-T. Cheng, and M.-H. Chen, "DoRA: Weight-decomposed low-rank adaptation," in *Proc. of International Conference on Machine Learning (ICML)*, 2024.

[229] F. Zhang and M. Pilanci, "Spectral adapter: Fine-tuning in spectral space," *arXiv preprint arXiv:2405.13952*, 2024.

[230] S. Gao, T. Hua, Y.-C. Hsu, Y. Shen, and H. Jin, "Adaptive rank selections for low-rank approximation of language models," in *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024, pp. 227–241.

[231] F. Meng, Z. Wang, and M. Zhang, "Pissa: Principal singular values and singular vectors adaptation of large language models," *arXiv preprint arXiv:2404.02948*, 2024.

[232] H. Wang, Z. Xiao, Y. Li, S. Wang, G. Chen, and Y. Chen, "MiLoRA: Harnessing minor singular components for parameter-efficient llm finetuning," *arXiv preprint arXiv:2406.09044*, 2024.

[233] X. Zhang, S. Wen, L. Han, F. Juefei-Xu, A. Srivastava, J. Huang, H. Wang, M. Tao, and D. N. Metaxas, "Spectrum-aware parameter efficient fine-tuning for diffusion models," *arXiv preprint arXiv:2405.21050*, 2024.

[234] G. Li, Y. Tang, and W. Zhang, "LoRAP: Transformer sub-layers deserve differentiated structured compression for large language models," in *Proc. of the International Conference on Machine Learning (ICML)*, 2024.

[235] Y. Yang, X. Li, Z. Zhou, S. L. Song, J. Wu, L. Nie, and B. Ghanem, "CorDA: Context-oriented decomposition adaptation of large language models," *arXiv preprint arXiv:2406.05223*, 2024.

[236] M. Nikdan, S. Tabesh, E. Crnčević, and D. Alistarh, "RoSA: Accurate parameter-efficient fine-tuning via robust adaptation," in *Proc. of International Conference on Machine Learning (ICML)*, 2024.

[237] M. G. A. Hameed, A. Milios, S. Reddy, and G. Rabusseau, "ROSA: Random subspace adaptation for efficient fine-tuning," *arXiv preprint arXiv:2407.07802*, 2024.

[238] P. Sharma, J. T. Ash, and D. Misra, "The truth is in there: Improving reasoning in language models with layer-selective rank reduction," in *Proc. of International Conference on Learning Representations (ICLR)*, 2023.

[239] A. Tjandra, R. Pang, Y. Zhang, and S. Karita, "Unsupervised learning of disentangled speech content and style representation," in *Proc. of Interspeech*, 2021, pp. 4089–4093.

[240] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen *et al.*, "Hierarchical generative modeling for controllable speech

synthesis," in *Proc. of International Conference on Learning Representations (ICLR)*, 2021.

[241] D. Liao, T. Jiang, F. Wang, L. Li, and Q. Hong, "Towards a unified conformer structure: from asr to asv task," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[242] Z. Song, L. He, P. Wang, Y. Hu, and H. Huang, "Introducing multilingual phonetic information to speaker embedding for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 10 091–10 095.

[243] T. Liu, R. K. Das, M. Madhavi, S. Shen, and H. Li, "Speaker-utterance dual attention for speaker and utterance verification," in *Proc. of Interspeech*, 2020, pp. 4293–4297.

[244] T. Liu, R. K. Das, K. A. Lee, and H. Li, "Neural acoustic-phonetic approach for speaker verification with phonetic attention mask," *IEEE Signal Processing Letters*, vol. 29, pp. 782–786, 2022.

[245] J. Bai, W. Wang, and C. P. Gomes, "Contrastively disentangled sequential variational autoencoder," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 10 105–10 118, 2021.

[246] H. Lu, X. Wu, Z. Wu, and H. Meng, "Speechtriplenet: End-to-end disentangled speech representation learning for content, timbre and prosody," in *Proc. of the 31st ACM International Conference on Multimedia*, 2023, pp. 2829–2837.

[247] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-vae," *arXiv e-prints*, p. arXiv:1804.03599, 2018.

[248] E. Dupont, "Learning disentangled joint continuous and discrete representations," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[249] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29, 2016.

[250] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun, "Disentangling factors of variation in deep representation using adversarial training," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29, 2016.

[251] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. of International Conference on Learning Representations (ICLR)*, 2020.

[252] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015.

[253] A. G. ALIAS PARTH GOYAL, A. Sordoni, M.-A. Côté, N. R. Ke, and Y. Bengio, "Z-forcing: Training stochastic recurrent networks," *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[254] Y. Zhu, M. R. Min, A. Kadav, and H. P. Graf, "S3VAE: Self-supervised sequential vae for representation disentanglement and data generation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6538–6547.

[255] J. Duchi, "Derivations for linear algebra and optimization," *Berkeley, California*, vol. 3, no. 1, pp. 2325–5870, 2007.

[256] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. of Interspeech*, 2019, pp. 2613–2617.

[257] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Prof. of International Conference on Learning Representations (ICLR)*, 2022.

[258] J. Zhang, J. Liss, S. Jayasuriya, and V. Berisha, "Robust vocal quality feature embeddings for dysphonic voice detection," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 31, pp. 1348–1359, 2023.

[259] K. Nam, Y. Kim, J. Huh, H.-S. Heo, J. weon Jung, and J. S. Chung, "Disentangled representation learning for multilingual speaker recognition," in *Proc. of InterSpeech*, 2023, pp. 5316–5320.

[260] A. Srinivas Menon, R. P. Gohil, K. Tripathi, and P. Wasnik, "LASPA: Language Agnostic Speaker Disentanglement with Prefix-Tuned Cross-Attention," in *Proc. of InterSpeech*, 2025, pp. 3623–3627.