Decomposition and Reorganization of Phonetic Information for Speaker Embedding Learning

Qian-Bei Hong, Chung-Hsien Wu, *Senior Member, IEEE*, and Hsin-Min Wang, *Senior Member, IEEE*

***Abstract*—Speech content is closely related to the stability of speaker embeddings in speaker verification tasks. In this paper, we propose a novel architecture based on self-constraint learning (SCL) and reconstruction task (RT) to remove the influence of phonetic information on speaker embedding generation. First, SCL is used to reduce the divergence of frame-level features, which can avoid ambiguity between the resulting embeddings of the two utterances being compared. Second, RT is used to further remove phonetic information in frame-level layers, focusing on speaker-discriminative feature transformation. In our experiments, the speaker embedding models were trained on the VoxCeleb2 dataset and evaluated on the VoxCeleb1, Librispeech, SITW and VoxMovies datasets. Experimental results on VoxCeleb1 show that the proposed DROP-TDNN system reduced the EER by 7.5%, compared to the state-of-the-art ECAPA-TDNN system. Furthermore, the proposed DROP-TDNN system also outperformed the ECAPA-TDNN system in the experiments on SITW, Librispeech and VoxMovies under cross-dataset conditions. In the experiments on SITW, the proposed system reduced the EER by 3.4% compared to the ECAPA-TDNN system. In the experiments on Librispeech, the proposed system demonstrated the advantage of removing phonetic information under the clean speech condition, with a significant reduction of 25.5% in EER compared to the ECAPA-TDNN system. In the experiments on VoxMovies, the proposed system reduced the EER by up to 7.9% compared to the ECAPA-TDNN system under different pronunciation and background conditions.**

***Index Terms*—Speaker verification, self-constraint learning, reconstruction learning, phonetic information**

I. Introduction

SPEAKER verification (SV) aims to distinguish whether the observed speech is from a claimed identity and has been widely applied in real-life application scenarios [1]-[3]. Traditional text-independent SV systems usually use the Gaussian mixture model-universal background model (GMM-UBM) to construct speaker models and i-vectors to represent the short-term spectral characteristics of speakers [4]-[8]. Since a few years, thanks to the development of deep learning, the performance of speaker representation based on a deep neural network (DNN) is significantly better than that of traditional SV systems [9]-[11]. Recently, many SV systems were based on the x-vector model [12]-[13], whose architecture consists of frame-level and segment-level feature transformations. Frame-level feature transformation is based on the time delay neural network (TDNN) structure to extract short-term features [14], and the shift-invariant frame-wise features extracted from multi-frame signals through TDNN have been proved to be more effective than single-frame features [15]. Segment-level feature transformation typically applies temporal pooling to aggregate variable-length frame-level features to obtain a fixed-dimensional vector.

For the development of SV, many studies have focused on the effectiveness of pooling structures to project frame-level features to segment-level features. A simply way is to average all the features of the last DNN hidden layer corresponding to the sequential input of an utterance into a d-vector [9]. After that, Snyder et al. [16] aggregated the outputs of the preceding hidden layer over time, and their mean and standard deviation were concatenated and processed by the last hidden layer to form a fixed-dimensional vector. Later, it became the underlying structure for x-vector-based models, such as extended-TDNN (E-TDNN) [17]. Zhu et al. [18] introduced a self-attention mechanism for DNN-based embedding and computed the speaker embedding as a weighted average of frame-level features. Desplanques et al. [19] proposed an ECAPA-TDNN architecture in which channel-dependent frame attention is used for pooling. Furthermore, transformer-based pre-trained models, such as wav2vec 2.0 [59], HuBERT [60], and WavLM [61], have greatly contributed to automatic speech recognition (ASR) by providing robust speech representations learned through masked speech prediction from large amounts of unlabeled speech. Later, pre-trained models were applied to SV and also achieved significant performance improvements [62]-[63].

In general, natural speech contains different factors, such as speech content [20], speaker identity, pitch [21] and emotion [22], which have a significant impact on the stability of speaker representations extracted from it. Adding additional features to improve the performance of speaker embedding learning has been extensively analyzed, especially phonetic features [23]-

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Contract No. MOST 108-2221-E-006-103-MY3.

Q.-B. Hong is with the Graduate Program of Multimedia Systems and Intelligent Computing, National Cheng Kung University and Academia Sinica, Taiwan (e-mail: qbhong75@gmail.com).

C.-H. Wu is with the Department of Computer Science and Information Engineering, National Cheng Kung University (e-mail: chunghsienwu@gmail.com).

H.-M. Wang is with the Institute of Information Science, Academia Sinica, Taiwan (e-mail: whm@iis.sinica.edu.tw).

[1]The code of the proposed DROP-TDNN is available at https://github.com/NCKU-MHMC/DROP-TDNN.

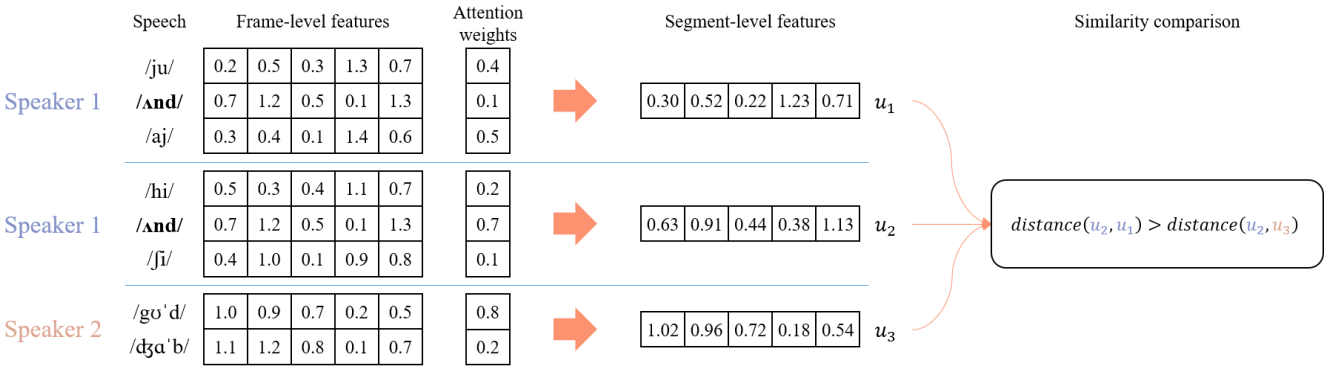


Fig. 1. Ambiguity in the statistics pooling process. Attention weights determine the importance of all frames of an utterance, but do not explicitly distinguish the similarity of speech features of different speakers. In this example, the frame-level feature distribution of Speaker 1's "/ʌnd/" is close to the distribution of Speaker 2's "/gʊˈd/" and "/ʤɑˈb/" frame-level features, but not the distribution of Speaker 1's other frame-level features. Inappropriate attention weights can produce poor segment-level features, leading to false similarity comparison results.

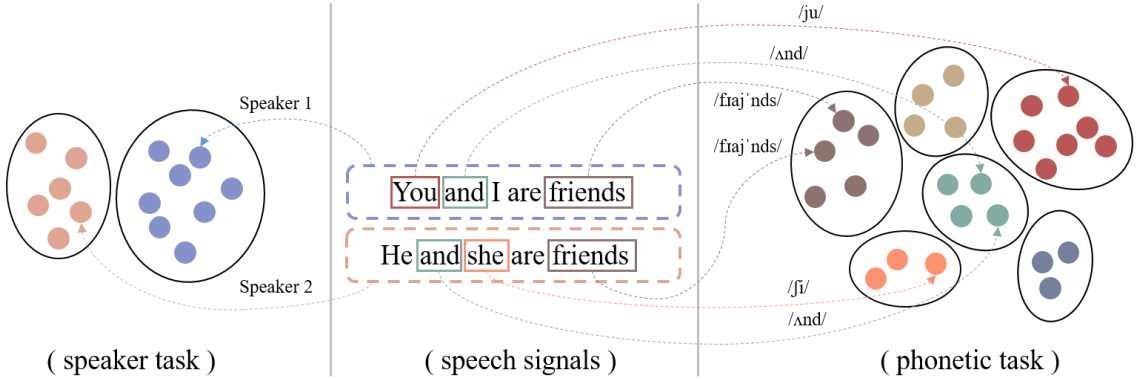


Fig. 2. Ambiguity in phonetic information learning in speaker embedding models. In MTL, the mutually exclusive tasks in some shared hidden layers will increase the training complexity. In this example, the features corresponding to "friends" of different speakers are distributed in different clusters in the speaker task, while they are distributed in the same cluster in the phonetic task.

[25]. Many studies directly combined frame-wise acoustic features with phonetic features as input to increase speaker specific information [26]-[27]. In [28], Stoll et al. compared speaker-discriminative features and phonetic-discriminative features for speaker recognition and achieved improved performance by combining phonetic features. However, the phonetic-discriminative features were extracted by a pre-trained network, while the back-end speaker recognition model was trained based on these features. The two-stage training procedure did not optimize phonetic-discriminative features for speaker recognition. Tang et al. [29] proposed a collaborative training approach based on a multi-task recurrent model, and obtained better performance on speech and speaker recognition tasks, compared to single-task models. Liu et al. [30] proposed two multi-task learning (MTL) methods using phonetic information to extract speaker embeddings. Experimental results show that the phonetic vectors effectively provide auxiliary information to adapt the speaker embedding extraction model, and the shared structure of MTL makes the network more robust. Chen et al. [31] proposed a phoneme-unit-specific TDNN (PUSTDNN) model, which uses a separate TDNN to model each phoneme unit to alleviate the impact of speech content on SV. Under the condition of high phoneme recognition accuracy, the PUSTDNN model outperformed the MTL systems with phonetic vectors. Furthermore, experimental results on different datasets show that the PUSTDNN model is more robust than the MTL systems.

Despite the excellent performance of the TDNN-based architecture with phonetic information in SV tasks, its implicit process deserves attention and further analysis, including statistics pooling and phonetic information learning in the speaker embedding model. First, as shown in Fig. 1, the attention pooling mechanism flexibly determines the importance of different frames for speaker embedding generation, but it cannot obtain the distance between two speaker embeddings. More precisely, the attention pooling mechanism cannot handle differences in speaker representations between the two utterances being compared. Second, although the speaker and phonetic factors in speech are different, they share some common information in the MTL structure. However, as shown in Fig. 2, MTL applied to speaker and phonetic information prediction tasks may face the mutually exclusive learning problem, and the training complexity increases accordingly. Furthermore, the quality of MTL-based speaker embeddings will inevitably be affected by the performance of the phonetic information prediction task. As reported in [31], MTL performs worse on the VoxCeleb dataset than the counterpart single-task baseline due to poor phoneme

recognition results.

In this paper, we propose a Decomposition and Reorganization of Phoneme-TDNN (DROP-TDNN) model to improve speaker embedding learning. The main purpose is to remove the influence of phonetic information in low-level features on speaker discrimination. The proposed architecture contains the following new ideas and contributions.

1) **Constraint of Frame-Level Feature Divergence:** We train speaker embedding models using self-constraint learning (SCL) to reduce the divergence of frame-level features. Additionally, this approach reduces the decision cost of the attention pooling mechanism and avoids ambiguity in the embeddings generated from the compared utterances. Experimental results show that SCL increases the stability of speaker embeddings to reduce divergence and ambiguity, thereby improving SV performance.
2) **Indirect Phonetic Information Decomposition and Reorganization:** We use frame-level features and phonetic features for reconstruction of acoustic features. Different from MTL, the phonetic information in frame-level features is indirectly reduced by phoneme prediction and data reconstruction in a neural network, which avoids the ambiguity caused by phonetic information in speaker embedding learning. Experimental results show that the reconstruction task (RT) can make the frame-level transformation pay more attention to speaker-discriminative features and improve the discriminative ability of speaker embeddings.

The remainder of this paper is organized as follows. A state-of-the-art (SOTA) system for SV is described in Section II. The proposed DROP-TDNN speaker embedding learning is presented in Section III. In Section IV, we introduce the datasets and provide the experimental results in detail. Finally, conclusions are given in Section V.

II. Speaker Embedding Learning with Phonetic Information

One of the most popular speaker embedding learning systems is the x-vector model. First, each utterance is fed to the frame-level layers to obtain low-level features, where TDNN is commonly used to encode time-varying signals by operating with multiple interconnected layers. Second, since each utterance is recorded with a variable temporal length, a pooling (e.g., statistics pooling) layer is applied to aggregate variable-length frame-level features into a fixed-dimensional segment-level feature vector. Then, the feature vector is passed through one or two fully-connected (FC) layers to obtain the non-linear features as the speaker embedding. Generally, the speaker embedding model is obtained by training a speaker classifier with a large amount of data.

A. SOTA Speaker Embedding Model

In the VoxCeleb Speaker Recognition Challenge (VoxSRC) 2019, the BUT system [32] and JHU-HLTCOE system [33] utilized the E-TDNN x-vector architecture for speaker embedding extraction, and both won the top three in the challenge. As is well known, the x-vector model is composed of frame-level and segment-level feature transformations. E-TDNN is applied to frame-level transformation, in which an FC layer is added between two 1D convolutional layers for additional feature transformation. In VoxSRC 2020, the ECAPA-TDNN model was proposed and won the first place in the challenge. First, instead of standard statistics pooling (global mean and standard deviation), ECAPA-TDNN utilizes channel- and context-dependent statistics pooling to aggregate all frame-level features to form the fixed-dimensional vector. Second, the Res2Net module [34] and the squeeze-excitation (SE) block [35] are combined into SE-Res2Block, which is beneficial to rescaling frame-level features, processing multi-scale features through hierarchical residual connections, and reducing the number of network parameters. Third, multi-layer feature aggregation (MFA) is introduced to combine the output features of all the SE-Res2Blocks for each frame. This architecture has become a standard configuration for speaker embedding extraction in many speaker-related tasks. In VoxSRC 2021, the IDLab team introduced the ECAPA CNN-TDNN [36] and ResNet-based model, which ranked third in the challenge. Recently, several studies used pre-trained models to generate strong speech representations as input features for downstream SV models [62]-[63]. Experimental results in [62] show that combining the pre-trained model with the ECAPA-TDNN architecture outperformed the VoxSRC 2021 winner system. It is worth noting that ECAPA-TDNN remains popular as a downstream SV model and provides beneficial speaker embedding generation capabilities. Therefore, the ECAPA-TDNN architecture remains one of the most popular baseline systems for SV comparison.

B. Phonetic Information in Speaker Embedding

Although speaker embedding models learned from acoustic features have superior performance, the influence of speech content on speaker embedding generation is ignored. Liu et al. [25] pointed out that speaker features are influenced by phonetic information; therefore, in order to improve the effectiveness of pooling, phonetic information should be considered in the frame-level layers. From this point of view, an intuitive approach is to use phonetic vectors as additional features to extract more discriminative speaker representations. This approach assumes that phonetic information is complementary to acoustic features and provides additional speaker-specific information for speaker embedding learning. A common method to phonetic information extraction is to use ASR models to predict phonetic content [26]. It has been proposed to generate phonetic vectors from the bottleneck layer [37] or the posterior output [38] of a neural network-based ASR model. There are also studies on jointly learning speaker information and phonetic information using MTL [30], which learns some shared components between acoustic and phonetic features to improve the quality and generalization ability of speaker embeddings, but the performance of the phonetic information prediction task must be good enough.

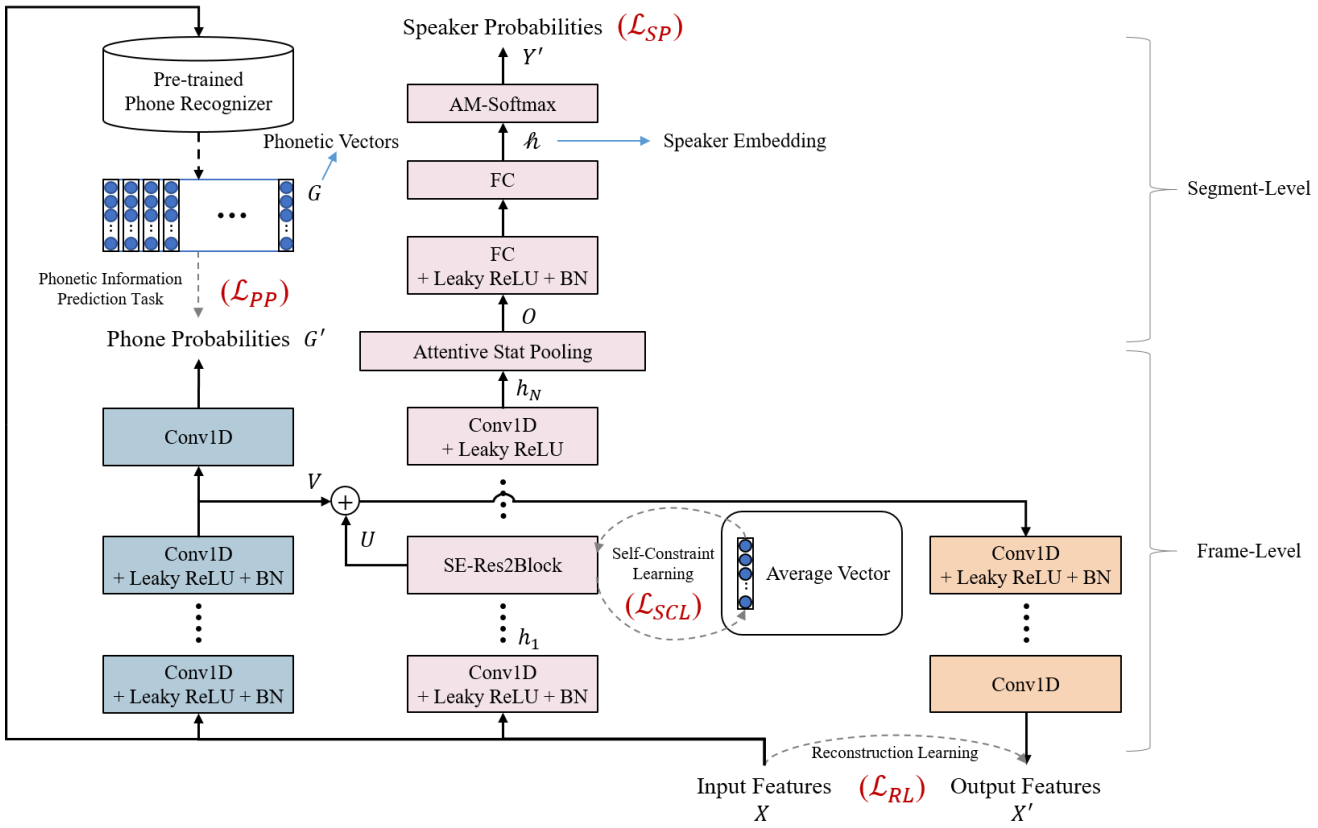


Fig. 3. The architecture of the proposed DROP-TDNN x-vector system. DROP-TDNN consists of three training procedures, including phonetic information prediction, reconstruction and speaker recognition.

III. Decomposition and Reorganization of Phoneme-TDNN

As shown in Fig. 3, the proposed DROP-TDNN x-vector model aims to reduce the impact of phonetic information on frame-level transformation.

A. Constraint of Frame-Level Feature Divergence

As mentioned above, the x-vector model can be divided into frame-level and segment-level processes. In the frame-level layers, the components generated by the initial layers appear cluttered due to differences in speech content, and the discriminative components are gradually generated through the non-linear transformation of N layers:

$$h_i = \mathcal{F}_i(h_{i-1}; \theta_{\mathcal{F}_i}), \qquad i = 1, \dots, N, \tag{1}$$

where h_i denotes the output of the i-th frame-level layer with parameters $\theta_{\mathcal{F}_i}$, and $\mathcal{F}_i(\cdot)$ denotes the i-th frame-level layer. The E-TDNN x-vector model uses the output of the N-th frame-level layer as the input to the pooling layer. In the ECAPA-TDNN x-vector model, multi-layer feature aggregation (MFA) is used in the N-th frame-level layer, and its output h_N, which is computed by

$$h_N = \mathcal{F}_N([h_2, \dots, h_{N-1}]; \theta_{\mathcal{F}_N}), \tag{2}$$

is used as the input to the pooling layer.

Since speech content causes differences in frame-level features at different time frames, attentive statistics pooling is used to determine the different contributions of frame-level features at different time frames to speaker representation generation. In SV tasks, the similarity function calculates the distance between two embeddings to distinguish identities. However, the attention weights determine the importance of individual frames in an utterance, but it does not explicitly distinguish the similarity of features between different speakers' voices. In order to improve the discriminative ability between embeddings from different speakers, we propose a self-constraint learning (SCL) mechanism to reduce the influence of phonetic features on speaker embedding generation. Let $U = h_i \in \mathbb{R}^{T \times C_U}$ represent the temporal features of the i-th frame-level layer, where T is the time length of an utterance, and C_U is the feature dimension of U. $u_{t,c} \in \mathbb{R}$ is the c-th component of the t-th frame-level feature vector in U. The SCL loss is computed by

$$\mathcal{D}_t = \left(\sum_{c=1}^{C_U} \left(u_{t,c} - \frac{1}{T}\sum_{\tau=1}^{T} u_{\tau,c}\right)^2\right)^{1/2}, \tag{3}$$

$$\mathcal{L}_{SCL} = \gamma \max([\mathcal{D}_1, \dots, \mathcal{D}_T]), \tag{4}$$

where $\mathcal{D}_t$ is the distance between the feature vector of the t-th frame of U and the average vector of all frames of U, and γ is the scale value of loss (γ was set to 0.01 in all experiments). By

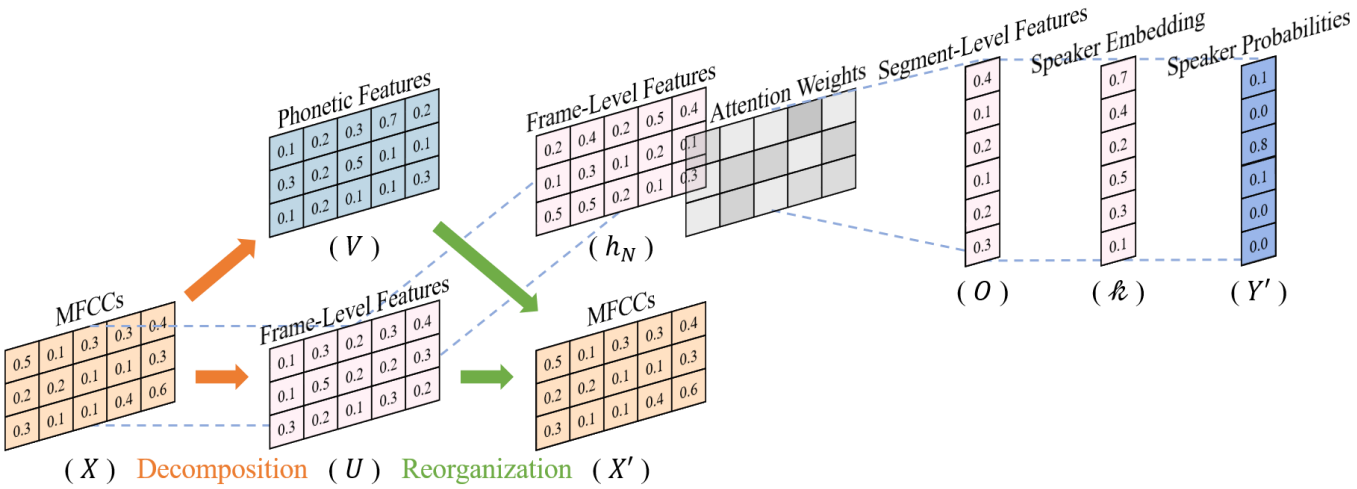


Fig. 4. The two stages of decomposition and reorganization in frame-level transformation in DROP-TDNN.

minimizing the maximal distance, we hope to reduce the variance of frame-level features in an utterance, and thus reduce the influence of phonetic features on speaker embedding generation.

B. Indirect Phonetic Information Decomposition and Reorganization

The analysis-reconstruction procedure has been successfully applied to audio-related tasks [39]-[40], benefiting from carrying specific information and model constraints. In this study, the reconstruction task with SCL is designed to remove the phonetic information from the frame-level features. Since speech consists of speaker characteristics and speech content, reconstruction can be regarded as a combination of speaker features and phonetic features over time.

In this study, RT reconstructs speech by reorganizing phonetic information and speaker features, and is trained with phonetic information prediction loss and reconstruction loss. First, a pre-trained phone recognizer is used to obtain the phone posteriorgram (PPG) G from the input frame-level acoustic features X (i.e., Mel-spectrogram). We adopt the ResNet structure [41]-[42] as the backbone of the phone recognizer. Second, a phonetic information prediction network is constructed to extract the phonetic features V from the input frame-level acoustic features X. Through the reconstruction of X based on speaker features U and phonetic features V, the phonetic information is indirectly removed from the speaker features in the frame-level layers.

In the phonetic information prediction network (see the dark blue part in Fig. 3), five stacked non-linear layers are used to map the acoustic features to the phonetic discriminant features. The input to the first layer at time step t is the concatenated features of the contextual window $[\tau - M/2, \ldots, \tau + M/2]$ of length M. The remaining four layers only take the output of the current time step of the previous layer as input. A simple Conv1D layer is used for PPG prediction. In the training stage, the PPG given by the pre-trained phone recognizer is used as the soft label, and the loss of phonetic information prediction is defined as

$$\mathcal{L}_{PP} = \sum_{t=1}^{T}\sum_{d=1}^{D}(g'_{t,d} - g_{t,d})^2 , \tag{5}$$

where $g_{t,d} \in G$ is the d-th component of the phonetic vector of the t-th frame given by the pre-trained phone recognizer, and $g'_{t,d}$ is the predicted result corresponding to $g_{t,d}$.

In the reconstruction learning process, X is fed to the phonetic information prediction network and mapped to the sequence output $V \in \mathbb{R}^{T\times C_V}$ of the last non-linear layer as the phonetic-discriminative features. V and U are concatenated on the feature dimension and passed through three Conv1D layers to obtain the reconstructed features X',

$$X' = \mathcal{R}_3(\mathcal{R}_2(\mathcal{R}_1([V,U];\theta_{\mathcal{R}_1});\theta_{\mathcal{R}_2});\theta_{\mathcal{R}_3}) , \tag{6}$$

where $\mathcal{R}_i(\cdot)$ denotes the i-th Conv1D layer in the reconstruction module. The reconstruction loss is formulated as

$$\mathcal{L}_{RL} = \sum_{t=1}^{T}\sum_{b=1}^{B}(x'_{t,b} - x_{t,b})^2 , \tag{7}$$

where $x_{t,b} \in X$ is the b-th Mel-energy of frame t, and $x'_{t,b}$ is the reconstructed energy corresponding to $x_{t,b}$. While SCL can reduce the temporal variation of the extracted feature vectors in U, RT may further indirectly make U pay more attention to speaker-discriminative features.

C. Speaker Classification

In frame-level transformation, RT is integrated with SCL to remove the phonetic information in the frame-level layers. Subsequently, the attention layer computes the segment-level features based on speaker-discriminative features.

The channel- and context-dependent statistics pooling mechanism [19] is used in the DROP-TDNN x-vector architecture to model the speaker-discriminative parts. Let $Q = h_N \in \mathbb{R}^{T\times C_s}$, where C_s is the feature dimension of h_N. The

attention weight is computed by

$$e_{t,c} = g_c^{\mathrm{T}} f(W \begin{bmatrix} q_{t,1} \\ \vdots \\ q_{t,C_s} \end{bmatrix} + b) + \ell_c \,, \tag{8}$$

$$e'_{t,c} = \frac{\exp(e_{t,c})}{\sum_{\tau=1}^{T} \exp(e_{\tau,c})}, \tag{9}$$

where $q_{t,c}$ denotes the c-th component of Q at time t. The parameters $W \in \mathbb{R}^{L \times C_s}$ and bias b project the input features to L-dimensional features, and then an activation function $f(\cdot)$ is used for non-linearity. After that, an activation weight $e_{t,c}$ corresponding to the t-th frame and the c-th feature dimension is computed through a weight vector $g_c \in \mathbb{R}^{L \times 1}$ and bias ℓ_c, and $e'_{t,c}$ is the normalized weight of $e_{t,c}$ computed by softmax.

Next, the weighted mean $\hat{\mu}_c$ and weighted standard deviation $\hat{\sigma}_c$ are obtained by

$$\hat{\mu}_c = \sum_{t=1}^{T} e'_{t,c} q_{t,c} \,, \tag{10}$$

$$\hat{\sigma}_c = (\sum_{t=1}^{T} e'_{t,c} (q_{t,c} - \hat{\mu}_c)^2)^{1/2} \,. \tag{11}$$

$$O = [\hat{\mu}_1, \dots, \hat{\mu}_{C_s}, \hat{\sigma}_1, \dots, \hat{\sigma}_{C_s}] \,, \tag{12}$$

where $O \in \mathbb{R}^{2C_s \times 1}$ is the attention output of speaker-discriminative features.

By pooling frame-level features to segment-level features, the speaker discrimination probabilities in DROP-TDNN are predicted by two consecutive hidden layers and a final softmax layer:

$$\hbar = \mathcal{H}_2(\mathcal{H}_1(O; \theta_{\mathcal{H}_1}); \theta_{\mathcal{H}_2}) \,, \tag{13}$$

$$Y' = Softmax(W^{\mathrm{T}} \hbar) \,, \tag{14}$$

where $\hbar \in \mathbb{R}^{C_D \times 1}$ denotes the speaker embedding obtained by two consecutive hidden layers with parameters $\theta_{\mathcal{H}_1}$ and $\theta_{\mathcal{H}_2}$, $\mathcal{H}_j(\cdot)$ denotes the j-th hidden layer, Y' denotes the predicted speaker probabilities, $W \in \mathbb{R}^{C_D \times C_{spk}}$ denotes the model parameters for predicting the probabilities, C_D is the feature dimension of $\hbar$ and C_{spk} is the number of speakers in the model training phase.

D. Summary

As shown in Fig. 4, DROP-TDNN uses two stages of decomposition and reorganization to reduce the influence of phonetic information in the frame-level transformation, and computes the speaker embedding of an utterance based on similar frame-level features in the utterance, thereby reducing the computational burden of attention pooling and improving the stability of speaker embeddings compared to ECAPA-TDNN.

The additive margin softmax (AM-Softmax) loss [43] is used to train the speaker classifier as

$$\mathcal{L}_{SP} = -\frac{1}{N} \sum_{n=1}^{N} log \frac{e^{s \cdot (W_{\Psi_n}^{\mathrm{T}} \hbar_n - m)}}{e^{s \cdot (W_{\Psi_n}^{\mathrm{T}} \hbar_n - m)} + \sum_{k=1, k \neq \Psi_n}^{K} e^{s W_k^{\mathrm{T}} \hbar_n}} \,, \tag{15}$$

where $W_{\Psi_n} \in \mathbb{R}^{C_D \times 1}$ is the weight vector for predicting the probability of the target speaker Ψ_n of the n-th training sample, $\hbar_n \in \mathbb{R}^{C_D \times 1}$ is the speaker embedding, K is the number of training speakers, and N is the number of training samples in a mini-batch. In order to follow the cosine definition, L2 normalization is applied to the features and weights. For all classes, a scale value s is used to avoid the large loss caused by the smaller feature norms. For the target class, a margin value m is used for angular margin adjustment in the training stage.

Moreover, L2-regularization with a Lagrange multiplier λ for weight decay is applied to all the model parameters in the DROP-TDNN x-vector architecture as

$$\mathcal{L}_{REG} = \lambda \sum_{j=1}^{J} w_j^2 \,, \tag{16}$$

where J denote the total number of model parameters.

The five losses (i.e., $\mathcal{L}_{SP}$, $\mathcal{L}_{SCL}$, $\mathcal{L}_{RL}$, $\mathcal{L}_{PP}$, and $\mathcal{L}_{REG}$) for DROP-TDNN training are optimized simultaneously, and the total loss is defined as

$$\mathcal{L}_{TOTAL} = \mathcal{L}_{SP} + \alpha \mathcal{L}_{SCL} + \beta(\mathcal{L}_{RL} + \mathcal{L}_{PP}) + \mathcal{L}_{REG} \,, \tag{17}$$

where α denotes the weight of SCL, and β denotes the weight of RT. Currently, we only use binary values (i.e., 0 or 1) for α and β to control whether the proposed SCL and RT are applied to the model training process. That is, both α and β are set to 1 in the proposed model indicating the consideration of both SCL and RT, $\alpha = 0$ indicates that SCL is not applied, while $\beta = 0$ indicates that RT is not applied.

IV. Experiments

The experiments were conducted on the Librispeech [44], VoxCeleb [45]-[46], SITW [47], VoxMovies [58] and HI-MIA [69] datasets. Because all SV models were trained using a cosine-based AM-Softmax loss for network adaptation, all experiments were compared by the cosine similarity backend, except for the experiments trained on the VoxCeleb1 dataset, which used the PLDA backend [48] for scoring.

A. Datasets

1) Librispeech

This dataset contains approximately 1,000 hours read English speech. We trained a phone recognizer with 39 output nodes corresponding to 39 phone units using the train-clean-100 subset of 28,539 audio files from 251 speakers. In DROP-TDNN, the pre-trained phone

TABLE I
THE PHONE SET USED IN THIS STUDY

Categories	Symbols
vowel	AA, AE, AH, AO, AW, AY, EH, ER, EY, IH, IY, OW, OY, UH, UW
semivowel	W, Y
affricate	CH, JH
fricative	DH, F, S, SH, TH, V, Z, ZH
nasal	M, N, NG
aspirate	HH
liquid	L, R
stop	B, D, G, K, P, T

recognizer's softmax output of 39 phonetic probabilities (PPG) is used as the phonetic vector (i.e., $C_p = 39$). Table I shows the categories of these 39 English phones [49].

2) VoxCeleb2

This dataset contains over 1 million utterances collected from 6,112 speakers on YouTube, and is divided into two subsets for SV: *DEV* and *TEST*. The *DEV* set contains 1,092,009 utterances from 5,994 speakers and has no overlap with the speakers in the VoxCeleb1 dataset. In this study, the *DEV* set of the VoxCeleb2 dataset was used to train the embedding extraction models (i.e., the baseline ECAPA-TDNN model and the proposed DROP-TDNN model). We followed the popular data augmentation strategy of Kaldi [50] recipe to generate extra samples for model training. In the Kaldi /egs/voxceleb/v2/ recipe, the MUSAN dataset (babble, noise, music) [51] and the RIR dataset (reverb) [52] were combined with the original utterances to generate 4 extra samples for each utterance. Note that in the Kaldi recipe, only 1,000,000 samples were randomly selected from the 4,368,036 extra samples to train the model with the original 1,092,009 utterances.

3) VoxCeleb1

This dataset contains 153,516 utterances collected from 1,251 speakers on YouTube, and is divided into two subsets for SV: *DEV* and *TEST*. The *DEV* set contains 148,642 utterances from 1,211 speakers, and the *TEST* set contains 4,874 utterances from 40 speakers. Three trial lists of this dataset formed under different conditions [53] were used for SV evaluation experiments. Furthermore, the *DEV* set was used to train the E-TDNN models to evaluate the effectiveness of SCL and RT on different frame-level layers.

4) SITW

This dataset contains samples of about 300 people in different scenarios, and there are often multiple speakers in a recording. The *EVAL* subset contains 2,883 audio files from 180 speakers, and is divided into four trial lists to evaluate SV.

5) VoxMovies

This dataset contains approximately 9,000 utterances from 856 speakers. Five trial lists formed from this dataset under different emotions, accents, and background noises are used for cross-domain SV evaluation experiments.

6) HI-MIA

This dataset consists of wake-up words "Hi, Mia" for text-dependent SV. The test subset contains 165,120 short utterances from 44 speakers, and is divided into two trial lists to distribute the microphone type and recording distance.

TABLE II
THE CONFIGURATION OF THE DROP-TDNN MODEL

Network	Layer	Structure	Input Dimension	Output Dimension
Backbone (ECAPA-TDNN)	Input			$T \times 80$
	Conv1D + Reaky ReLU + BN	kernel size=5 dilation rate=1	$T \times 80$	$T \times C_U$
	SE-Res2Block	kernel size=3 dilation rate=2	$T \times C_U$	$T \times C_U$
	SE-Res2Block	kernel size=3 dilation rate=3	$T \times C_U$	$T \times C_U$
	SE-Res2Block	kernel size=3 dilation rate=4	$T \times C_U$	$T \times C_U$
	Conv1D + Reaky ReLU	kernel size=1 dilation rate=1	$T \times 3C_U$	$T \times C_s$
	Attentive Statistics Pooling		$T \times C_s$	$2C_s$
	FC + Reaky ReLU + BN		$2C_s$	C_D
	FC		C_D	C_D
	AM-Softmax (Output)		C_D	5,994
Phonetic Information Prediction Task	Input			$T \times 80$
	Conv1D + Reaky ReLU + BN	kernel size=9 dilation rate=1	$T \times 80$	$T \times C_V$
	Conv1D + Reaky ReLU + BN	kernel size=1 dilation rate=1	$T \times C_V$	$T \times C_V$
	Conv1D + Reaky ReLU + BN	kernel size=1 dilation rate=1	$T \times C_V$	$T \times C_V$
	Conv1D + Reaky ReLU + BN	kernel size=1 dilation rate=1	$T \times C_V$	$T \times C_V$
	Conv1D + Reaky ReLU + BN	kernel size=1 dilation rate=1	$T \times C_V$	$T \times C_V$
	Conv1D (Output)	kernel size=1 dilation rate=1	$T \times C_V$	$T \times 39$
Reconstruction	Input			$T \times (C_V + C_U)$
	Conv1D + Reaky ReLU + BN	kernel size=1 dilation rate=1	$T \times (C_V + C_U)$	$T \times 1{,}024$
	Conv1D + Reaky ReLU + BN	kernel size=1 dilation rate=1	$T \times 1{,}024$	$T \times 1{,}024$
	Conv1D (Output)	kernel size=1 dilation rate=1	$T \times 1{,}024$	$T \times 80$

B. Experimental Setup

In the experiments, all audio files were labeled with speaker identities, and silent intervals were removed using energy-based voice activity detection. The input acoustic feature vector contained 80-dimentional Mel-frequency cepstral coefficients (MFCCs) extracted based on a 25ms window with a stride of 10ms. The training data consisted of randomly sampled 2 to 4 second segments from the training utterances. All models were trained with a cyclical learning rate [54] between 0.01 and 0.1. The duration was set to 130k steps for each cycle, and each model was trained for 10 cycles. The learning rate decayed by a factor of 0.75 per cycle. Early stopping was used to avoid overfitting. All models were trained by AM-Softmax with a margin value of 0.2 and a scale value of 30. Stochastic gradient descent (SGD) with a momentum of 0.9 was applied for model optimization. All weights used L2-regularization with a Lagrange multiplier of 2e-5 for weight decay (i.e., $\lambda = 2\text{e-}5$). The mini-batch size was set to 128.

The ResNet-34 architecture [41] was replicated as the backbone of the phone recognizer for phonetic vector extraction. The baseline ECAPA-TDNN x-vector architecture was copied from [19]. Each SE-Res2Block had 1,024 output channels (i.e., $C_U = 1{,}024$). The scale dimension in Res2Block was set to 8. The bottleneck dimension in SE-Block was set to 128. The last frame-level layer had 3,072 output channels (i.e., $C_s = 3{,}072$). The two consecutive embeddings had 192 channels (i.e., $C_D = 192$), which were also trained by predicting the probabilities of

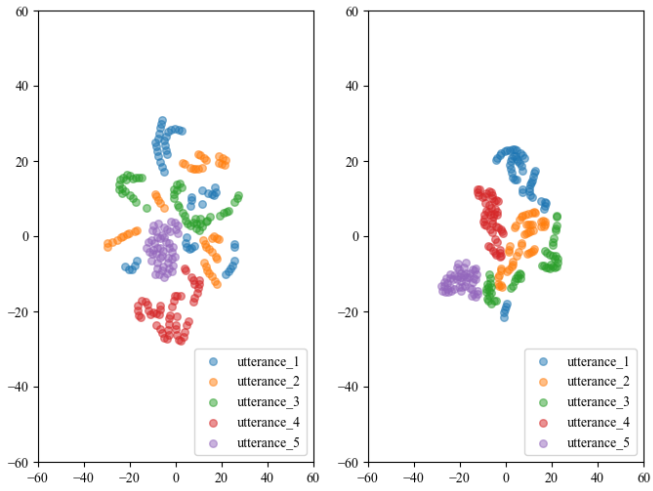


Fig. 5. Feature distribution of the last frame-level layer for 5 utterances of different speakers from the VoxCeleb1 dataset using t-SNE. (a) ECAPA-TDNN. (b) ECAPA-TDNN + SCL.

5,994 speakers. The leaky rectified linear unit (Leaky ReLU) [55] activation function with a negative ratio of 0.2 and batch normalization (BN) were applied to each transformation layer for non-linear mapping. In the proposed DROP-TDNN-based model, ECAPA-TDNN was used as the backbone, and additional five 1D convolutional non-linear layers and one 1D convolutional linear layer were used to map the acoustic features to the phonetic probabilities (PPG), and each layer had 512 output channels (i.e., $C_V = 512$). The kernel size of the first layer was set to 9 (i.e., $M = 9$), and the kernel size of the other layers was set to 1. This phonetic information prediction task is to regress the sequence phonetic vectors corresponding to the input features X. Besides, the reconstructed features X' were generated using two 1D convolutional non-linear layers with 1,024 output channels and one 1D convolutional linear layers with 80 output channels. The kernel size of these layers was set to 1. The detailed configuration for DROP-TDNN is shown in Table II. For score normalization, adaptive score normalization (AS-Norm) [56] was used. The impostor cohorts were generated by averaging the speaker embeddings for each speaker in the VoxCeleb2 *DEV* set (i.e., the number of impostor cohorts was 5,994), and the scores of the top 200 most similar cohorts were used for score normalization. The training and testing data were prepared by the Kaldi toolkit, and all neural network models were implemented by TensorFlow platform.

We reported SV results in term of equal error rate (EER) and minimum detection cost function (minDCF) with $p_{target} = 0.01$.

C. Effect of Frame-Level Feature Divergence

As stated in the introduction, phonetic information is one of the factors that affects the stability of speaker embeddings. Here, we intended to investigate the relationship between the divergence of frame-level features and the stability of speaker embeddings. Firstly, we evaluated SCL in model training. SCL

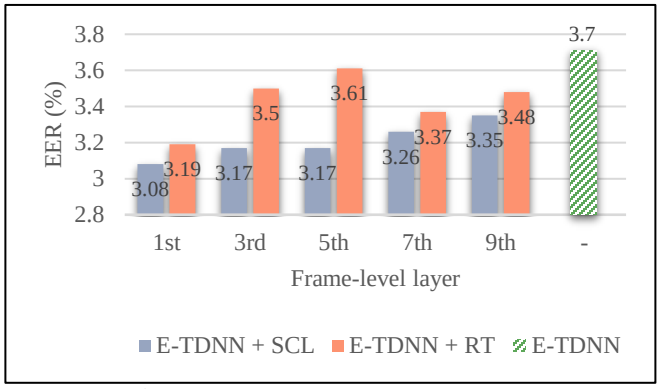


Fig. 6. EERs of E-TDNN embedding models with SCL and RT applied at different frame-level layers on the VoxCeleb1 *TEST* set.

TABLE III
PERFORMANCE OF DROP-TDNN-BASED AND ECAPA-TDNN-BASED SYSTEMS ON THE VOXCELEB1-E AND VOXCELEB1-H TRIAL LISTS

System	VoxCeleb1-E (cleaned)		VoxCeleb1-H (cleaned)	
	EER (%)	minDCF	EER (%)	minDCF
ECAPA-TDNN	1.33	0.1398	2.25	0.2005
DROP-TDNN ($\alpha = 1, \beta = 1$)	**1.24**	**0.1322**	**2.11**	**0.1973**

was applied to the last frame-level layer of the ECAPA-TDNN architecture. Fig. 5 shows the feature distribution of the last frame-level layer for five utterances of different speakers randomly selected from the VoxCeleb1 dataset using t-SNE [57]. The results reveal that the frame-level features without SCL were more sparsely distributed than the frame-level features with SCL.

D. SCL and RT Applied at Different Frame-Level Layers

In this experiment, we analyzed the effect of SCL and RT on different frame-level layers. To reduce the experiment costs, the E-TDNN x-vector architecture (implemented with a standard DNN structure as in [32]) were used as the backbone. We also reduced the training duration to 50k steps for each cycle, and each model was trained for 10 cycles. As mentioned in Section IV.A, the VoxCeleb1 *DEV* set was used for model training, and the VoxCeleb1 *TEST* set with a trial list (i.e., the VoxCeleb1 (cleaned) trial list) was used for SV evaluation. Fig. 6 shows the performance of SCL and RT applied at different frame-level layers. E-TDNN consisted of 9 frame-level layers, and the comparison was performed on odd-numbered frame-level layers. For SCL, we can see that EER continued to drop as SCL was applied to lower layers. The results show that SCL could effectively deal with the details of speech characteristics in the lower layers. On the other hand, compared to other layers, RT achieved the best performance when applied to the first frame-level layer. Since RT indirectly removes phonetic information from features at frame-level layers, the results seem to imply that applying RT at the lowest layer can remove phonetic information early and facilitate subsequent layers to effectively extract speaker features layer by layer.

E. Performance of the DROP-TDNN Model

In the above experiments, SCL and RT have respectively

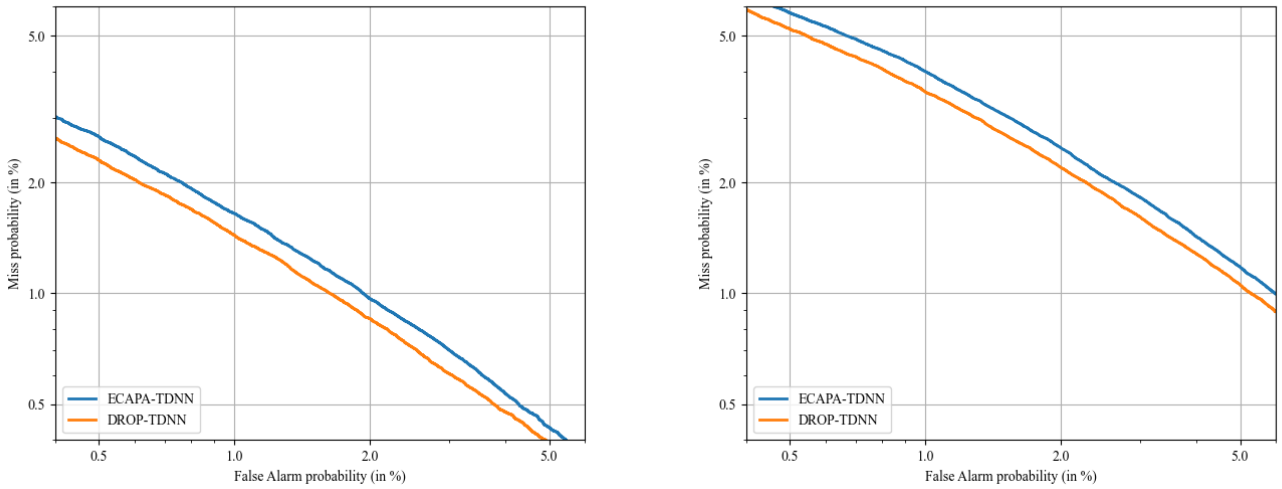


Fig. 7. DET curves of different speaker embeddings evaluated on the VoxCeleb1-E (cleaned) trial list (left) and VoxCeleb1-H (cleaned) trial list (right).

TABLE IV
ABLATION STUDY OF DROP-TDNN WITH MULTI-LAYER SCL ON THE VOXCELEB1-E AND VOXCELEB1-H TRIAL LISTS

System	Weight		VoxCeleb1-E (cleaned)		VoxCeleb1-H (cleaned)	
	α	β	EER (%)	minDCF	EER (%)	minDCF
DROP-TDNN (multi-layer SCL)	1	1	**1.23**	**0.1312**	**2.11**	**0.1929**
	1	0	1.27	0.1347	2.16	0.1930
	0	1	1.32	0.1365	2.21	0.1935
DROP-TDNN (single-layer SCL)	1	1	1.24	0.1322	2.11	0.1973

demonstrated the ability to improve the performance of SV. In this experiment, we integrated SCL and RT into the ECAPA-TDNN x-vector system to evaluate the proposed DROP-TDNN x-vector system. Based on the above experimental results, SCL and RT was applied to the first frame-level layer. Table III shows the performance of SV experiments on the VoxCeleb1 dataset. We compared the proposed DROP-TDNN x-vector systems with the baseline ECAPA-TDNN x-vector systems. From Table III, we can see that the DROP-TDNN x-vector system outperformed the baseline ECAPA-TDNN x-vector system on all trial lists. Evaluated on the VoxCeleb1-E (cleaned) trial list, DROP-TDNN reduced the EER by 6.8% compared with ECAPA-TDNN (1.24% vs 1.33%). Evaluated on the VoxCeleb1-H (cleaned) trial list, DROP-TDNN reduced the EER by 6.2% compared with ECAPA-TDNN (2.11% vs 2.25%). Fig. 7 shows the DET curves of different systems. The DET curves of the DROP-TDNN-based systems were obviously closer to the lower left corner than those of the ECAPA-TDNN-based systems on the VoxCeleb1-E and VoxCeleb1-H trial lists. Therefore, the results confirm that DROP-TDNN-based speaker embedding learning using SCL and RT can effectively improve the discriminative ability of speaker embeddings and outperforms the state-of-the-art ECAPA-TDNN-based speaker embedding learning.

F. SCL at Multiple Frame-Level Layers of DROP-TDNN

We also investigated the performance of applying SCL to multiple frame-level layers. Since the input of the last frame-level layer of DROP-TDNN is the integration of the outputs of all previous SE-Res2Blocks by MFA, the SCL applied to the last frame-level layer of DROP-TDNN can be regarded as simultaneously applying this constraint to different frame-level layers. Table IV shows the results of an ablation study on DROP-TDNN using multi-layer SCL on the VoxCeleb1 dataset. In the table, $\alpha = 0$ means that SCL was not applied, while $\beta = 0$ means that RT was not applied. Evaluated on RT, we can see that DROP-TDNN ($\alpha = 1$ and $\beta = 1$) with multi-layer SCL reduced the EER by 3.1% on the VoxCeleb1-E (cleaned) trial list (1.23% vs 1.27%) and 2.3% on the VoxCeleb1-H (cleaned) trial list (2.11% vs 2.16%), respectively, compared with DROP-TDNN ($\alpha = 1$ and $\beta = 0$) with multi-layer SCL. Evaluated on multi-layer SCL, DROP-TDNN ($\alpha = 1$ and $\beta = 1$) with multi-layer SCL reduced the EER by 6.8% and 4.5% (VoxCeleb1-E (cleaned): 1.23% vs 1.32% and VoxCeleb1-H (cleaned): 2.11% vs 2.21%), respectively, compared with DROP-TDNN ($\alpha = 0$ and $\beta = 1$) without SCL. Furthermore, DROP-TDNN ($\alpha = 1$ and $\beta = 1$) with multi-layer SCL outperformed DROP-TDNN ($\alpha = 1$ and $\beta = 1$) with single-layer SCL. On the other hand, DROP-TDNN ($\alpha = 1$ and $\beta = 0$) with multi-layer SCL reduced the EER by 3.8% on the VoxCeleb1-E (cleaned) trial list (1.27% vs 1.32%) and 2.3% on the VoxCeleb1-H (cleaned) trial list (2.16% vs 2.21%), respectively, compared with DROP-TDNN ($\alpha = 0$ and $\beta = 1$). DROP-TDNN ($\alpha = 0$ and $\beta = 1$) multi-layer SCL performed worse than DROP-TDNN ($\alpha = 1$ and $\beta = 0$) with multi-layer SCL because it did not utilize SCL so that its frame-level features were under the influence of phonetic information. Obviously, SCL and RT provide

TABLE V
COMPARISON BETWEEN LIBRISPEECH AND VOXCELEB TRIAL LISTS

Name	# Speakers	# Utterances	# Positive Pairs	# Negative Pairs
VoxCeleb1 (cleaned)	40	4,708	18,802	18,809
VoxCeleb1-E (cleaned)	1,251	145,160	289,921	289,897
VoxCeleb1-H (cleaned)	1,190	137,924	275,488	275,406
Librispeech	251	2,925	35,500	8,517,200

TABLE VI
PERFORMANCE OF DROP-TDNN AND ECAPA-TDNN EMBEDDING SYSTEMS ON THE SITW, LIBRISPEECH AND VOXMOVIES TRIAL LISTS

System	SITW *EVAL* core-core		Librispeech		VoxMovies E-1		E-2		E-3		E-4		E-5	
	EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF	EER (%)	minDCF
ECAPA-TDNN	1.74	**0.1515**	0.55	0.0379	8.02	0.5351	8.70	**0.5904**	9.49	0.7712	9.80	0.7738	12.88	0.7615
DROP-TDNN (multi-layer SCL) ($\alpha = 1, \beta = 1$)	**1.68**	0.1534	**0.41**	**0.0336**	**7.44**	**0.5281**	**8.25**	0.5921	**8.74**	**0.7300**	**9.17**	**0.7506**	**12.19**	**0.7568**

TABLE VII
PERFORMANCE OF DROP-TDNN AND ECAPA-TDNN EMBEDDING SYSTEMS ON THE HI-MIA TRIAL LIST

System	HI-MIA_1m	
	EER (%)	minDCF
ECAPA-TDNN	8.29	0.5257
DROP-TDNN (multi-layer SCL) ($\alpha = 1, \beta = 1$)	**8.24**	**0.5119**

complementary contributions to reduce the impact of phonetic information on speaker embedding generation.

G. Comparison of Systems on Other Datasets

Although the above experimental results have demonstrated that the proposed DROP-TDNN embedding systems can achieve excellent performance on the down-stream SV task on the VoxCeleb1 dataset, the performance of generalization across datasets must be examined, which is related to the stability of speaker embeddings. In this experiment, the SITW dataset was used to evaluate cross-dataset generalization, and the *EVAL* core-core trial list was used for SV. Moreover, the Librispeech dataset was used to evaluate SV performance under the clean condition, and the VoxMovies dataset was used to evaluate SV performance under different pronunciation and background conditions. Since there is no generic trial list for SV evaluation in Librispeech, we selected audio files with IDs from 0001 to 0005 in audio Chapters to generate a simulated trial list by exhaustive pairing. Without pairing the same audio files, the selected 2,925 audio files yielded 8,552,700 trial pairs (i.e., $\#\text{pairs} = 2925^2 - 2925$). Table V summarizes the main differences in trials between VoxCeleb and Librispeech. As shown in Table VI, DROP-TDNN outperformed ECAPA-TDNN and reduced the EER by 3.4% (1.68% vs 1.74%) on the *EVAL* core-core trial list, but the minDCF was slightly increased. Evaluated on the Librispeech simulated trial list, DROP-TDNN significantly reduced the EER by 25.5% compared with ECAPA-TDNN (0.41% vs 0.55%). Clean speech provides explicit phonetic information, which is beneficial to reduce the influence of phonetic information in DROP-TDNN. This may be the reason for the large improvement on the Librispeech trial. Evaluated on VoxMovies E-1, E-2, E-3, E-4, and E-5 trial lists, DROP-TDNN reduced the EER by 7.2%, 5.2%, 7.9%, 6.4% and 5.4% (E-1: 7.44% vs 8.02%, E-2: 8.25% vs 8.70%, E-3: 8.74% vs 9.49%, E-4: 9.17% vs 9.80%, and E-5: 12.19% vs 12.88%), respectively, compared with ECAPA-TDNN. Therefore, the results confirm that the proposed DROP-TDNN has robust speaker embedding stability under cross-dataset conditions. The results on Librispeech further demonstrate the advantage of DROP-TDNN in removing phonetic information under the clean condition. Moreover, the HI-MIA dataset was used to evaluate the performance of text-dependent SV. Different to the other text-independent SV datasets, HI-MIA is a dataset of wake-up words with short and consistent pronunciation. Table VII shows the results of text-dependent SV on the HI-MIA trial list. Evaluated on the 1m (i.e., one-meter distance) trial list, even though the utterances provide insufficient length to generate stable speaker embeddings, DROP-TDNN still outperformed ECAPA-TDNN and reduced the EER by 0.6% (8.24% vs 8.29%). Therefore, the DROP-TDNN demonstrated the stability of speaker embedding extraction.

H. Comparison Among Different SOTA Systems

The proposed systems are also compared with other SOTA systems in Table VIII. Since the amount of data augmented in [19] was approximately six times the amount of data used in this study, ECAPA-TDNN (Re-implemented) performed worse than ECAPA-TDNN (C=1024) [19]. Kwon et al. [64] showed that data augmentation helps to significantly improve performance and reduce overfitting. In Table VIII, DROP-TDNN was also combined with different methods to confirm whether the SCL and RT lead to the expected performance improvement. First, when phonetic vectors were added as additional features to the last frame-level layer of DROP-TDNN (similar to the work in [30]), the performance degraded on the VoxCeleb1-E (cleaned) trial list (1.25% vs 1.23%) and the VoxCeleb1-H (cleaned) trial list (2.12% vs 2.11%), compared with DROP-TDNN. The reason for the result may be that the frame-level layers of DROP-TDNN are dedicated to

TABLE VIII
COMPARISON DROP-TDNN AND OTHER EMBEDDING SYSTEMS ON THE VOXCELEB1-E AND VOXCELEB1-H TRIAL LISTS

System	Input Feature (Dim.-Type)	#Aug.	Samples	Loss Function	VoxCeleb1-E (cleaned)		VoxCeleb1-H (cleaned)	
					EER (%)	minDCF	EER (%)	minDCF
ECAPA-TDNN (C=1024) [19]	80-MFCCs	✓	6,552,054	AAM-Softmax	1.12	0.1318	2.12	0.2101
MFA-TDNN (Standard) [65]	80-FBanks	✓	5,460,045	AAM-Softmax	1.08	0.1175	2.05	0.1897
ARET-25 [66]	161-Spec.	×	0	AAM-Softmax	1.52	-	2.61	-
ECAPA-TDNN-ResPEL [67]	80-MFCCs	✓	1,000,000	AM-Softmax	1.17	0.1223	2.03	0.1895
ResNet18 [19]	80-MFCCs	✓	6,552,054	AAM-Softmax	1.60	0.1789	2.88	0.2672
ResNet34 [19]	80-MFCCs	✓	6,552,054	AAM-Softmax	1.33	0.1560	2.46	0.2288
ResNet160 [32]	40-FBanks	✓	5,000,000	AAM-Softmax	1.38	-	2.50	-
ResNet256 [32]	40-FBanks	✓	5,000,000	AAM-Softmax	1.35	-	2.48	-
Res2Net-26w8s [68]	80-FBanks	✓	-	Softmax	1.47	0.1692	2.72	0.2717
H / ASP [64]	64-FBanks	×	0	AP+Softmax	1.42	-	2.77	-
H / ASP [64]	64-FBanks	✓	-	AP+Softmax	**1.07**	-	2.21	-
ECAPA-TDNN (Re-implemented)	80-MFCCs	✓	1,000,000	AM-Softmax	1.33	0.1398	2.25	0.2005
A1: DROP-TDNN (multi-layer SCL) ($\alpha = 1, \beta = 1$)	80-MFCCs	✓	1,000,000	AM-Softmax	1.23	0.1312	2.11	0.1929
A1 + phonetic vectors (G)	80-MFCCs	✓	1,000,000	AM-Softmax	1.25	0.1294	2.14	0.2004
A1 + phonetic vectors (V)	80-MFCCs	✓	1,000,000	AM-Softmax	1.25	0.1279	2.12	0.1868
A1 + ResPEL	80-MFCCs	✓	1,000,000	AM-Softmax	1.10	**0.1173**	**1.94**	**0.1756**

TABLE IX
COMPARISON DROP-TDNN AND OTHER PRE-TRAINED SYSTEMS ON THE VOXCELEB1 TRIAL LIST

System	Input Feature	Aug.	VoxCeleb1 (cleaned)
			EER (%)
ECAPA-TDNN	Fbank [62]	✓	2.37
	HuBERT Base (Last) [62]	✓	3.08
	HuBERT Base (Hidden) [62]	✓	**1.86**
ECAPA-TDNN	WavLM (Hidden)	✓	2.47
DROP-TDNN (multi-layer SCL) ($\alpha = 1, \beta = 1$)	WavLM (Hidden)	✓	**2.15**

Here, the models were trained by VoxCeleb1 *Dev* set, and the pre-trained models were fixed in training process for speech representations extraction.

reducing the influence of phonetic information on speaker embedding generation, while adding phonetic vectors to the last frame-level layer may cause ambiguity in attention pooling. Therefore, using phonetic vectors as additional features in DROP-TDNN was not always effective, and there was no significant difference in SV performance whether phonetic vectors were used or not. Second, the combination of DROP-TDNN and parent embedding learning (PEL) [67] (i.e., DROP-TDNN + ResPEL) reduced the EER by 10.6% on the VoxCeleb1-E (cleaned) trial list (1.10% vs 1.23%) and 8.1% on the VoxCeleb1-H (cleaned) trial list (1.94% vs 2.11%), respectively, compared with DROP-TDNN. Furthermore, DROP-TDNN + ResPEL achieved the best performance on the VoxCeleb1-H (cleaned) trial list.

Table IX shows the results of DROP-TDNN and ECAPA-TDNN using speech representations extracted by different self-supervised learning models. Clearly, DROP-TDNN using WavLM features outperformed the ECAPA-TDNN on the VoxCeleb1 (cleaned) trial list. Therefore, the results confirm that DROP-TDNN-based speaker embedding learning using SCL and RT can be combined with other SOTA systems to further improve SV performance.

V. CONCLUSION

Speech content is closely related to the stability of speaker embeddings. In this paper, we have proposed a DROP-TDNN architecture for training the speaker embedding extractor, which is different from common studies focusing on extracting speaker features from phonetic information. The proposed DROP-TDNN applies self-constraint learning and reconstruction task to reduce the impact of phonetic information on speaker embedding generation. The backbone of DROP-TDNN architecture is suitable for different x-vector-based models, such as E-TDNN and SOTA ECAPA-TDNN. Since DROP-TDNN focuses on using learning methods to optimize the quality of speaker representations, the proposed self-constraint learning and reconstruction task do not require additional memory to setup the corresponding sub-networks in the inference stage. From the experimental results, the proposed DROP-TDNN x-vector system has the potential to improve the applicability of text-independent speaker verification under different conditions.

REFERENCES

[1] V A. Prasad, and R Ranjith, "Intelligent chatbot for lab security and automation," in *Proc. Int. Conf. Comput. Commun. Netw. Technol.*, West Bengal, India, 2020, pp. 1–4.

[2] I. Bisio, C. Garibotto, A. Grattarola, F. Lavagetto, and A. Sciarrone, "Smart and robust speaker recognition for context-aware in-vehicle applications," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8808–8821, Sep. 2018.

[3] Y. Dong, and Y.-D. Yao, "Secure mmWave-radar-based speaker verification for IoT smart home," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3500–3511, Mar. 2020.

[4] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Toulouse, France, 2006, pp. I-97–I-100.

[5] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[6] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in GMM-based speaker verification," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 4, pp. 1448–1460, May 2007.

[7] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason, "I-vector based speaker recognition on short utterances," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Florence, FI, Italy, 2011, pp. 2341–2344.

[8] A. Larcher, P.-M. Bousquet, K. A. Lee, D. Matrouf, H. Li, and J.-F. Bonastre, "I-vectors in the context of phonetically-constrained short utterances for speaker verification," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Kyoto, Japan, 2012, pp. 4773–4776.

[9] 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. Int. Conf. Acoust. Speech Signal Process.*, Florence, FI, Italy, 2014, pp. 4052–4056.

[10] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Shanghai, China, 2016, pp. 5115–5119.

[11] H. Muckenhirn, M. M. Doss, and S. Marcell, "Towards directly modeling raw speech signal for speaker verification using CNNs," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Calgary, AB, Canada, 2018, pp. 4884–4888.

[12] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Stockholm, AB, Sweden, 2017, pp. 999–1003.

[13] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Calgary, AB, Canada, 2018, pp. 5329–5333.

[14] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Audio Speech Lang. Process.*, vol. 37, no. 3, pp. 328–339, Mar. 1989.

[15] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *IEEE Autom. Speech Recognit. Underst. Workshop*, Scottsdale, AZ, USA, 2015, pp. 92–97.

[16] 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 *IEEE Workshop Spok. Lang. Technol.*, San Diego, CA, USA, 2016, pp. 165–170.

[17] 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. Int. Conf. Acoust. Speech Signal Process.*, Brighton, UK, 2019, pp. 5796–5800.

[18] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Hyderabad, India, 2018, pp. 3573–3577.

[19] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Shanghai, China, 2020, pp. 3830–3834.

[20] S. Wang, J. Rohdin, L. Burget, O. Plchot, Y. Qian, K. Yu, and J. Cernocky, "On the usage of phonetic information for text-independent speaker embedding extraction," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Graz, Austria, 2019, pp. 1148–1152.

[21] M. Ravanelli, and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *IEEE Workshop Spok. Lang. Technol.*, Athens, Greece, 2018, pp. 1021–1028.

[22] R. Pappagari, T. Wang, J. Villalba, N. Chen, and N. Dehak, "X-vectors meet emotions: A study on dependencies between emotion and speaker recognition," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Barcelona, Spain, 2020, pp. 7169–7173.

[23] T. Zhou, Y. Zhao, J. Li, Y. Gong, and J. Wu, "CNN with phonetic attention for text-independent speaker verification," in *IEEE Autom. Speech Recognit. Underst. Workshop*, Sentosa, Singapore, 2019, pp. 718–725.

[24] N. Chen, Y. Qian, and K. Yu, "Multi-task learning for text-dependent speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Dresden, Germany, 2015, pp. 185–189.

[25] Y. Liu, L. He, J. Liu, and M. T. Johnson, "Introducing phonetic information to speaker embedding for speaker verification," *Eurasip J. Audio Speech Music Process.*, vol. 2019, no. 1, pp. 1–17, Dec. 2019.

[26] L. Li, D. Wang, Y. Chen, Y. Shi, Z. Tang, and T. F. Zheng, "Deep factorization for speech signal," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Calgary, AB, Canada, 2018, pp. 5094–5098.

[27] A. K. Sarkar, C.-T. Do, V.-B. Le, and C. Barras, "Combination of cepstral and phonetically discriminative features for speaker verification," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1040–1044, Sep. 2014.

[28] L. Stoll, J. Frankel, and N. Mirghafori, "Speaker recognition via nonlinear discriminant features," in *NOLISP*, Paris, France, 2007, pp. 27–30.

[29] Z. Tang, L. Li, D. Wang, and R. Vipperla, "Collaborative joint training with multitask recurrent model for speech and speaker recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 3, pp. 493–504, Mar. 2017.

[30] Y. Liu, L. He, J. Liu, and M. T. Johnson, "Speaker embedding extraction with phonetic information," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Hyderabad, India, 2018, pp. 2247–2251.

[31] X. H. Chen, and C. C. Bao, "Phoneme-unit-specific time-delay neural network for speaker verification," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 1243–1255, Mar. 2021.

[32] H. Zeinali, S. Wang, A. Silnova, P. Matejka, and O. Plchot, "BUT system description to VoxCeleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[33] D. Garcia-Romero, A. McCree, D. Snyder, and G. Sell, "JHU-HLTCOE system for the VoxSRC speaker recognition challenge," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Barcelona, Spain, 2020, pp. 7559–7563.

[34] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Aug 2019.

[35] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 7132–7141.

[36] J. Thienpondt, B. Desplanques, and K. Demuynck, "The IDLAB VoxCeleb speaker recognition challenge 2021 system description," *arXiv preprint arXiv:2109.04070*, 2021.

[37] P. Matejka, O. Glembek, O. Novotny, O. Plchot, F. Grezl, L. Burget, and J. Cernocky, "Analysis of DNN approaches to speaker identification," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Shanghai, China, 2016, pp. 5100–5104.

[38] D. Cai, Z. Cai, and M. Li, "Deep speaker embeddings with convolutional neural network on supervector for text-independent speaker recognition," in *Asia-Pac. Signal Inf. Process. Assoc. Annu. Summit Conf.*, Honolulu, HI, USA, 2018, pp. 1478–1482.

[39] K. W. Cheuk, Y.-J. Luo, E. Benetos, and D. Herremans, "The effect of spectrogram reconstruction on automatic music transcription: An alternative approach to improve transcription accuracy," in *Int. Conf. Pattern Recognit.*, Milan, Italy, 2020, pp. 9091–9098.

[40] H. Wang, Y. Qian, X. Wang, Y. Wang, C. Wang, S. Liu, T. Yoshioka, J. Li, and D. L. Wang, "Improving noise robustness of contrastive speech representation learning with speech reconstruction," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Singapore, 2022, pp. 6062–6066.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.

[42] M. Algabri, H. Mathkour, M. A. Bencherif, M. Alsulaiman, and M. A. Mekhtiche, "Towards deep object detection techniques for phoneme recognition," *IEEE Access*, vol. 8, pp. 54663–54680, Mar. 2020.

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

[44] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, South Brisbane, QLD, Australia, 2015, pp. 5206–5210.

[45] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Stockholm, AB, Sweden, 2017.

[46] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Hyderabad, India, 2018.
[47] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The speakers in the wild (SITW) speaker recognition database," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, San Francisco, CA, USA, 2016, pp. 818–822.
[48] S. J. Prince, and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE Int. Conf. Comput. Vision*, Rio de Janeiro, Brazil, 2007, pp. 1–8.
[49] J. Kominek, and A. W Black, "The CMU Arctic speech databases," in *ISCA Speech Synth. Workshop*, Pittsburgh, PA, USA, 2004, pp. 223–224.
[50] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *IEEE Autom. Speech Recognit. Underst. Workshop*, Big Island, HI, USA, 2011.
[51] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.
[52] 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. Int. Conf. Acoust. Speech Signal Process.*, New Orleans, LA, USA, 2017, pp. 5220–5224.
[53] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Comput. Speech Lang.*, vol. 60, 2020, Art. no. 101027.
[54] L. N. Smith, "Cyclical learning rates for training neural networks," in *IEEE Winter Conf. App. Comput. Vision*, Santa Rosa, CA, USA, 2017, pp. 464–472.
[55] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013.
[56] S. Cumani, P. D. Batzu, D. Colibro, C. Vair, P. Laface, and V. Vasilakakis, "Comparison of speaker recognition approaches for real applications," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Florence, FI, Italy, 2011, pp. 2365–2368.
[57] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[58] A. Brown, J. Huh, A. Nagrani, J. S. Chung, and A. Zisserman, "Playing a part: Speaker verification at the movies," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Toronto, ON, Canada, 2021, pp. 6174–6178.
[59] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Graz, Austria, 2019, pp. 3465–3469.
[60] 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 Trans. Audio Speech Lang. Process.*, vol. 29, pp. 3451–3460, Oct. 2021.
[61] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Jour. Select. Topics Signal Process.*, vol. 16, no. 6, pp. 1505–1518, Oct. 2022.
[62] 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. Int. Conf. Acoust. Speech Signal Process.*, Singapore, 2022, pp. 6147–6151.
[63] S. Chen, Y. Wu, C. Wang, S. Liu, Z. Chen, P. Wang, G. Liu, J. Li, J. Wu, X. Yu, and F. Wei, "Why does self-supervised learning for speech recognition benefit speaker recognition?" in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Incheon, Korea, 2022, pp. 3699–3703.
[64] Y. Kwon, H.-S. Heo, B.-J. Lee, and J. S. Chung, "The ins and outs of speaker recognition: lessons from VoxSRC 2020," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Toronto, ON, Canada, 2021, pp. 5809–5813.
[65] 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 *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Singapore, 2022, pp. 7517–7521.
[66] R. Zhang, J. Wei, W. Lu, L. Wang, M. Liu, L. Zhang, J. Jin, and J. Xu, "ARET: Aggregated residual extended time-delay neural networks for speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, Shanghai, China, 2020, pp. 946–950.
[67] Q.-B. Hong, C.-H. Wu and H.-M. Wang, "Generalization ability improvement of speaker representation and anti-Interference for speaker verification," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 486–499, 2023.
[68] T. Zhou, Y. Zhao, and J. Wu, "Resnext and res2net structures for speaker verification," in *Proc. IEEE Spoken Lang. Technol. Workshop*, Shenzhen, China, 2021, pp. 301–307.
[69] X. Qin, H. Bu, and M. Li, "HI-MIA: A far-field text-dependent speaker verification database and the baselines," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Barcelona, Spain, 2020, pp. 7609–7613.



Qian-Bei Hong received the B.S. and M.S. degrees in electrical engineering from Southern Taiwan University of Science and Technology, Tainan, Taiwan, in 2009 and 2011, respectively. He received the Ph.D. degree in Graduate Program of Multimedia Systems and Intelligent Computing, National Cheng Kung University and Academia Sinica, Tainan, Taiwan, in 2023. He is currently a principal engineer at Taiwan Mobile. His research interests include deep learning, speaker recognition, automatic speech recognition, and multimedia signal processing.



Chung-Hsien Wu (Senior Member, IEEE) received the B.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 1987 and 1991, respectively. Since 1991, he has been with the Department of Computer Science and Information Engineering, NCKU. He became the Chair Professor in 2017. He served as the Deputy Dean of the College of Electrical Engineering and Computer Science, NCKU, from 2009 to 2015. He also worked at Computer Science and Artificial Intelligence Laboratory of Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in summer 2003, as a Visiting Scientist. He was an Associate Editor of the *IEEE Transactions on Audio, Speech and Language Processing* (2010–2014) and the *IEEE Transactions on Affective Computing* (2010–2014). He is currently an Associate Editor of *ACM Transactions on Asian and Low-Resource Language Information Processing*. Currently, he is the APSIPA BoG Member (2019~2021).



Hsin-Min Wang (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1989 and 1995, respectively. In October 1995, he joined the Institute of Information Science, Academia Sinica, Taipei, Taiwan, where he is currently a Research Fellow. He also holds a joint appointment as a Professor with the Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan. His major research interests include spoken language processing, natural language processing, multimedia information retrieval, machine learning, and pattern recognition. From 2016 to 2020, he was an Associate Editor for *IEEE/ACM Transactions on Audio, Speech and Language Processing*. He is currently a Senior Editor of *APSIPA Transactions on Signal and Information Processing*. He is a General Co-Chair of ISCSLP2016, ISCSLP2018, and ASRU2023 and a Technical Co-Chair of ISCSLP2010, O-COCOSDA2011, APSIPAASC2013, ISMIR2014, ASRU2019, and APSIPAASC2023. He was the recipient of the Chinese Institute of Engineers Technical Paper Award in 1995 and the ACM Multimedia Grand Challenge First Prize in 2012. He was an APSIPA distinguished lecturer for 2014–2015. He is also a Member of the International Speech Communication Association and ACM.