Robust Speaker Verification using Deep Weight Space Ensemble

Weiwei Lin and Man-Wai Mak, *IEEE, Senior Member*

***Abstract*—Domain shift is one of the most challenging problems in speaker verification. Although numerous methods have been proposed to address domain shift, most approaches optimize the performance of one domain at the sacrifice of the other. As a result, to obtain the best performance, each domain requires a dedicated model. However, deploying multiple models is resource-demanding and impractical, particularly when the deployment domains are not known in advance. Recent studies in deep neural networks (DNNs) suggest that near the low error surface of the DNN's weight space, there exists a linear path connecting a base model and a fine-tuned model. This finding inspires us to combine the strength of the fine-tuned models and the base models to solve challenging SV problems. Specifically, we aim to develop models that can handle (1) mixed text-dependent (TD) and text-independent (TI) speaker verification where the speech content can be either unconstrained or constrained, (2) cross-channel speaker verification where the recording can be 16kHz high-fidelity microphone speech or 8kHz telephone speech, and (3) bi-lingual speaker verification where the enrollment and test speech can be one of the two languages. With weight space ensemble, we show that we can substantially improve the tasks mentioned above, with a 39.6% improvement in mixing TD and TI SV, a 17.4% improvement in bi-lingual SV, and an 18.4% improvement in cross-channel SV. Moreover, we show that the weight space ensemble can also enhance the performance in the target domain, thanks to the regularization effect of the interpolation.**

***Index Terms*—robust speaker recognition, domain adaptation, domain shift, weight space ensemble**

I. Introduction

Robustness is crucial for speaker verification (SV) systems when they are used for biometric authentication. However, the performance of speaker verification systems is often compromised by domain shift [1]–[4]. For example, an SV system trained on English corpora would perform poorly on Chinese speech. Even if the languages are the same, the differences in channels and recording devices can also degrade speaker verification performance. This change of data characteristics is referred to as domain shift in the literature [5]. Mathematically speaking, domain shift happens when the training and test data are sampled from two different distributions. The training and test data are said to come from the source and target domains, respectively.

Domain adaptation (DA) is performed to mitigate the effect of domain shift using a small amount of data from the target domain. In speaker verification, there are several popular DA methods. For instance, In Kaldi's SRE16 recipe,[1] adaptation is carried out by interpolating the Probabilistic Linear Discriminant Analysis (PLDA) model's mean and covariance matrices. The interpolation can be done either in a supervised way or unsupervised way [6]. Another popular DA method for the PLDA backend is correlation alignment (CORAL) [7], [8]. CORAL whitens the source-domain data and then recolors the whitened data by a transformation matrix estimated from the target-domain data. If labeled data are enough, we can directly fine-tune the neural networks to optimize their performance on the target domain. For a very deep neural network, the weights of the bottom layers are typically fixed, and only the upper layers' weights are fine-tuned [9].

However, the above methods optimize the target domain's performance in the sacrifice of the source domain. In practice, we want the model to be robust to domain shifts, i.e., performing well on both domains. For example, there are a lot of bilingual speakers in the world. It would be more desirable if speaker verification systems are not language-specific. Besides, in some situations, the target domain is not known in advance. Take another example, as the voice over internet protocol (VoIP) is getting more and more popular, it is expected that a call center should deal with both VoIP and traditional telephone calls. If we optimize the speaker verification system on either VoIP or telephone calls, the overall performance will suffer.

Another type of domain shift involves applying a model trained with unconstrained speech content to speech with constraint content. For example, text-independent speaker verification (TI-SV) does not constrain the content of the enrollment and test speech, while text-dependent speaker verification (TD-SV) requires the content of enrollment speech and test speech to be the same and pre-defined. Because constrained speech is hard to collect in large quantities, models are typically trained on unconstrained speech and then fine-tuned on constrained speech [10]. However, because the fine-tuned TD system is optimized for constrained speech, it would not perform well on the TI task. In such a case, two models would have to be deployed, one for TI-SV and one for TD-SV. It would be more desirable to leverage the broad applicability of TI models and the high performance of TD models on constrained speech. However, to the best of our knowledge, there is no such work in the literature.

The loss landscapes of DNNs are notorious for having multiple local minima. It is suggested in [11] that some local minima lead to better performance than the others. The stochastic nature of stochastic gradient descent (SGD) leads to

This work was in part supported by National Natural Science Foundation of China (NSFC), Grant No. 61971371 and Huawei Technologies Co., Ltd, Project No. TC20210903021. Weiwei Lin and M.W. Mak are with Department of Electronic and Information Engineering, The Hong Kong Polytechnic University.

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2

a different minimum (model) for each training run. Although these minima have similar values, the weights of their corresponding models are different. Recently, it was found that with an appropriate learning schedule, a model can traverse from one minimum to another minimum with just several epochs [12]. In the literature, it is referred to as linear-model connectivity [13]. This finding inspires researchers to explore an ensemble of models comprising a pre-trained model and several fine-tuned models in the weight space for robust image classification [14]. Besides the weight space ensemble (WSE), several influential papers use the "simple" model averaging trick. For example, Povey *et al.* [15] proposed using parameter averaging combined with natural gradient for efficient parallel training of DNNs. In [16], the authors proposed using a cyclical learning rate scheduler to find multiple points along the trajectory of SGD and then perform weight averaging to obtain a better optimum. It is worth noting that the methods in [15] and [16] will not work with naive weight averaging. Instead, they require natural gradient solutions or a specific learning rate scheduler to traverse the loss landscape. Our WSE approach has a similar property, i.e., naive weight averaging will not work in our case.

We proposed using weight space ensemble to build domain robust speaker embedding networks. Our contributions are two-fold. First, we identify the weakness of state-of-the-art SV systems in mixed domain speaker verification, which includes mixed text-dependent and text-independent SV, cross-channel SV, and bi-lingual SV. We showed that both pre-trained models and fine-tuned models have inherent weaknesses in these mixed domain scenarios. But with WSE and a well-designed fine-tuning procedure, we can dramatically improve performance. Second, we developed an effective fine-tuning procedure for multiple datasets, which utilizes a non-parametric loss function with multiple negative samples to make the fine-tuning effective. We show that applying two learning rate schedulers to the pre-trained layers and the classification head separately helps mitigate the learning disparity under the margin-based classification loss.

II. Speaker Embedding Networks and PLDA Backend

A. X-vector Architecture

The x-vector network consists of three parts: frame-level time-delay neural networks (TDNNs), utterance-level fully-connected (FC) layers, and a statistics pooling layer that bridges the frame-level layers and utterance-level layers [17], [18]. A TDNN is a special form of convolutional neural network (CNN). It skips the computation at chosen temporal positions while maintaining the same receptive field size as a CNN. A statistics pooling layer concatenates the mean and standard deviation of the activations from the last convolutional layer. The concatenated means and standard deviations are passed to two FC layers. The network is trained to minimize the standard cross-entropy loss using small chunks of acoustic sequences derived from the original utterances. The typical chunk length ranges from 2 seconds to 4 seconds. After the network is trained, the embedding of each utterance is extracted from the first affine layer after the statistics pooling layer. A backend consisting of LDA and PLDA models is trained using the embeddings as input [19].

TABLE I. Architecture of our x-vector network. The 3-tuples in the column "kernel" specify the kernel size, stride, and dilation, respectively. N is the number of training speakers

Layer	Kernel	Channel_in × Channel_out
Conv1	5,1,1	40 × 512
Conv2	3,1,2	512 × 512
Conv3	3,1,3	512 × 512
Conv4	1,1,1	512 × 512
Conv5	1,1,1	512 × 1536
Statistics pooling	–	1536 × 3072
FC6	–	3072 × 512
FC7	–	512 × 512
AM-Softmax	–	512 × N

B. DenseNet Architecture for Speaker Embedding

Although the x-vector network has shown remarkable improvement over the traditional factor analysis framework, it was found that deep speaker embedding networks, such as ResNet, Res2Net, and DenseNet, can further improve the performance [20]–[22].

DenseNets are proposed in [23] for computer vision. It has also been successfully used in SV [21]. A DenseNet comprises two block types, namely, dense block and transition block. In a dense block, each layer is connected by all the output from the previous layers. To prevent the number of feature maps from growing excessively, a transition block is introduced to reduce the feature map size. Suppose each convolutional layer produces k feature maps, then the l-th layer inside the block has $k_0 + k \times (l-1)$ feature maps, where k_0 is the number of channels in the input layer. The parameter k is referred to as the growth rate. In this work, we used a dense network composed of 1-dimensional convolution instead of 2D convolution. We used the same statistics pooling layer as that of the x-vector network. Because max-pooling and average pooling do not work well in speaker recognition, we replaced max-pooling with a stride-2 convolution layer. Table II shows our network architecture.

C. Fine-tuning Speaker Recognition Models

After the model had been trained on a sizeable upstream dataset, we switched to employing the downstream target data to update (fine-tune) the model using a small learning rate for a few epochs. We refer to this process as fine-tuning. Although fine-tuning is very popular in computer vision and natural language processing, speaker verification researchers are more in favor of backend adaptation, which includes PLDA adaptation [6], inter dataset variability compensation (IDVC) [24], within-class correlation normalization (WCCN) [25], source normalization [26], and correlation alignment (CORAL) [8].

This paper advocates that fine-tuning the front end (i.e., speaker embedding networks) can be very effective. What's more, front-end fine-tuning opens the opportunity to leverage

TABLE II. THE DENSENET ARCHITECTURE FOR SPEAKER EMBEDDING. THE GROWTH RATE FOR THE NETWORKS IS 40. "CONV k" CORRESPONDS TO THE SEQUENCE BN (BATCH-NORMALIZATION)-RELU-CONV WITH KERNEL SIZE k. N IS THE NUMBER OF TRAINING SPEAKERS.

Layers	Output Size	Operation
Convolution	400×80	conv 3
Dense Block 1	400×320	$\begin{bmatrix} \text{conv 1} \\ \text{conv 3} \end{bmatrix} \times 6$
Transition Layer 1	200×160	conv 2 stride 2
Dense Block 2	200×640	$\begin{bmatrix} \text{conv 1} \\ \text{conv 3} \end{bmatrix} \times 12$
Transition Layer 2	100×320	conv 2 stride 2
Dense Block 3	100×1280	$\begin{bmatrix} \text{conv 1} \\ \text{conv 3} \end{bmatrix} \times 24$
Transition Layer 3	50×640	conv 2 stride 2
Dense Block 4	50×1280	$\begin{bmatrix} \text{conv 1} \\ \text{conv 3} \end{bmatrix} \times 16$
Stats-pooling Layer	50×2560	Pooling
FC	256	FeedForward
Classification Layer	N	AM-Softmax

the properties of the DNN loss landscape to make the speaker embedding network robust under multiple domains. To train a base model or source-domain model, we used additive margin softmax. Additive margin loss [27] enforces a minimum margin m between the target and non-target classes:

$$\begin{aligned}\mathcal{L}_{AM}(\mathbf{W}, \mathbf{z}_i, y_i) &= -\log \frac{e^{s\cdot\left(\cos\theta_{y_i}-m\right)}}{e^{s\cdot\left(\cos\theta_{y_i}-m\right)}+\sum_{j=1,j\neq y_i}^{N} e^{s\cdot\cos\theta_j}} \\ &= -\log \frac{e^{s\cdot\left(\mathbf{w}_{y_i}^{\mathsf{T}}\mathbf{z}_i-m\right)}}{e^{s\cdot\left(\mathbf{w}_{y_i}^{\mathsf{T}}\mathbf{z}_i-m\right)}+\sum_{j=1,j\neq y_i}^{N} e^{s\mathbf{w}_j^{\mathsf{T}}\mathbf{z}_i}},\end{aligned} \tag{1}$$

where ($\mathbf{w}_j$ is the j-th column of weight matrix $\mathbf{W}$ and $\mathbf{z}_i$ is an embedding vector, both of which are normalized to have unit length. s is a scaling constant. $y_i \in \{1, \ldots, N\}$ is a speaker label, and N is the number of training speakers.

Another very popular loss function is triplet loss [28]:

$$\mathcal{L}_{\text{triplet}}\left(\mathbf{z}_i^a, \mathbf{z}_i^p, \mathbf{z}_i^n\right) = \max\left\{\left\|\mathbf{z}_i^a - \mathbf{z}_i^p\right\|^2 - \left\|\mathbf{z}_i^a - \mathbf{z}_i^n\right\|^2 + m, 0\right\}, \tag{2}$$

where $\mathbf{z}_i^p$ is a positive-class embedding sharing the class label with the anchor embedding $\mathbf{z}_i^a$, and $\mathbf{z}_i^n$ is the negative-class embedding. Compared with Eq. 1, Eq. 2 has two distinctions. Firstly, Eq. 2 does not have any parameters, which makes it more suitable for fine-tuning the speaker embedding networks, because learning a new classification layer from scratch is more challenging. Secondly, the triplet-loss explicitly relies on negative examples, while the softmax-based loss encodes the information of negative examples in the class weight matrix. Because of this, the triplet loss or metric learning-based approaches in general often require hard sample mining and larger batch sizes than their softmax counterpart.

During the fine-tuning stage, we used the normalized temperature-scaled cross-entropy loss (NT-Xent) [29], which has attracted a lot of attention recently. Instead of pair-wise or triplet-wise selections, given an embedding vector pair $\{\mathbf{z}_i^a,\ \mathbf{z}_i^p\}$ from one class and a set of embedding vectors, $\mathbf{Z}_i^n = \{\mathbf{z}_{i,k}^n\}_{k=1}^K$ from the other K classes, NT-Xent computes a softmax by contrasting the same-class score $\text{sim}(\mathbf{z}_i^a, \mathbf{z}_i^p)$ against all the different-class scores $\{\text{sim}(\mathbf{z}_i^a, \mathbf{z}_{i,k}^n)\}_{k=1}^K$:

$$\mathcal{L}_{\text{nt}}(\mathbf{z}_i^a, \mathbf{z}_i^p, \mathbf{Z}_i^n) = -\log \frac{\exp\left(\text{sim}\left(\mathbf{z}_i^a, \mathbf{z}_i^p\right)/\tau\right)}{\sum_{k=1}^K \exp\left(\text{sim}\left(\boldsymbol{z}_i^a, \boldsymbol{z}_{i,k}^n\right)/\tau\right)}. \tag{3}$$

This is also referred to as negative sampling in the literature. Compared with the regular cross-entropy loss, Eq. 3 does not involve any extra parameters such as the class-weight matrix. Therefore, it does not require learning extra parameters during the fine-tuning stage. We found this property greatly simplifies the fine-tuning process. Compared with the triple loss in Eq. 2, Eq. 3 compares one positive pair with multiple negative pairs.

D. PLDA and Backend Adaptation

Probabilistic linear discriminant analysis (PLDA) [19], [30] has been a popular backend for x-vector systems. Given a set of D-dimensional length-normalized [31] DNN embedding vectors $\{\mathbf{z}_{ij}; i = 1, \ldots, N; j = 1, \ldots, H_j\}$ from N speakers, each with H_i sessions, PLDA assumes that the embedding vectors can be expressed as the following factor analysis model:

$$\mathbf{z}_{ij} = \mathbf{m} + \mathbf{V}\mathbf{f}_i + \boldsymbol{\epsilon}_{ij}, \tag{4}$$

where $\mathbf{m}$ is the global mean of the embedding vectors, $\mathbf{V}$ defines the speaker subspace, $\mathbf{f}_i$ is the speaker factor, and $\boldsymbol{\epsilon}_{ij}$ is the residue whose covariance matrix $\boldsymbol{\Sigma}_w$ represents non-speaker variability. For detailed derivations and discussions on the training and scoring of PLDA models, refer to [4].

Given an enrollment speaker embedding $\mathbf{z}_e$ and a test speaker embedding $\mathbf{z}_t$, a PLDA score is computed by evaluating the log-likelihood ratio between the same-speaker hypothesis and the different–speaker hypothesis:

$$s = g(\mathbf{z}_e, \mathbf{z}_t, \boldsymbol{\Sigma}_b, \boldsymbol{\Sigma}_w), \tag{5}$$

where $\boldsymbol{\Sigma}_b = \mathbf{V}\mathbf{V}^{\mathsf{T}}$ is the between-speaker covariance matrix.

For PLDA adaptation, we use the source-domain data to train a PLDA model and obtain the source-domain between-speaker covariance matrix $\boldsymbol{\Sigma}_b^s$ and within-speaker covariance matrix $\boldsymbol{\Sigma}_w^s$. Then, we train a PLDA model using target-domain data to obtain the target-domain between-speaker covariance matrix $\boldsymbol{\Sigma}_b^t$ and within-speaker covariance matrix $\boldsymbol{\Sigma}_w^t$. Then, we carry out adaptation by interpolating the source-domain covariance matrices with target-domain covariance matrices:

$$\begin{aligned}\boldsymbol{\Sigma}_b^{adapt} &= \alpha\boldsymbol{\Sigma}_b^s + (1-\alpha)\boldsymbol{\Sigma}_b^t \\ \boldsymbol{\Sigma}_w^{adapt} &= \alpha\boldsymbol{\Sigma}_w^s + (1-\alpha)\boldsymbol{\Sigma}_w^t,\end{aligned} \tag{6}$$

where $\alpha \in [0, 1]$ is an interpolating coefficient. Then the PLDA score is computed using the interpolated covariance matrices [32]:

$$s = g(\mathbf{z}_e, \mathbf{z}_t, \boldsymbol{\Sigma}_b^{adapt}, \boldsymbol{\Sigma}_w^{adapt}). \tag{7}$$

.

Fig. 1. T-SNE plots showing the distributions of the speaker embeddings of the Mandarin speakers (top row) and English speakers (bottom row) from a base model, a fine-tuned model, and a WSE model. Different colors represent different speakers. The base model was trained by using English speech from the VoxCeleb corpus. For the fined-tuned model in the 2nd column, it was fine-tuned by using Mandarin speech from the CN-Celeb corpus.

III. Deep Weight Space Ensemble

Assume that we have a speaker embedding network with parameters denoted by $\boldsymbol{\theta}$. For notational simplicity, the acoustic features of an utterance are represented by $\mathbf{x}_i$. The forward pass of the network that maps the speech features $\mathbf{x}_i$ to an embedding vector $\mathbf{z}_i$ can be written as:

$$\mathbf{z}_i = f(\mathbf{x}_i, \boldsymbol{\theta}). \tag{8}$$

Typically, we have a large amount of source-domain data $\mathcal{S} = \{\mathbf{x}_i, y_i\}$ for training a base model, where y_i is a speaker label. To train a base model, we use additive large margin softmax in Eq. 1. The optimized $\mathbf{W}_s$ and $\boldsymbol{\theta}_s$ for source domain data is:

$$\mathbf{W}_s, \boldsymbol{\theta}_s = \arg\min_{\mathbf{W},\boldsymbol{\theta}} \left\{ \sum_{(\mathbf{x}_i, y_i)\in\mathcal{S}} \mathcal{L}_{AM}(\mathbf{W}, f(\mathbf{x}_i, \boldsymbol{\theta}), y_i) \right\}. \tag{9}$$

After the base network has been trained, we discard the classification head $\mathbf{W}_s$ and fine-tune the embedding network parameter $\boldsymbol{\theta}_s$ on target domain data $\mathcal{T} = \{\mathbf{x}_i^a, \mathbf{x}_i^p, \mathbf{X}_i^n\}$ using NT-Xent, where $\mathbf{x}_i^a$ and $\mathbf{x}_i^p$ are the speech segments of the same speakers and $\mathbf{X}_i^n = \{\mathbf{x}_{i,k}^n\}_{k=1}^K$ is a set comprising the speech segments from the speakers other than that of $\mathbf{x}_i^a$ and $\mathbf{x}_i^p$. Using Eq. 3 and Eq. 8, the optimized embedding network parameters $\boldsymbol{\theta}_t$ for target-domain data can be obtained by:

$$\boldsymbol{\theta}_t = \arg\min_{\boldsymbol{\theta}} \left\{ \sum_{(\mathbf{x}_i^a, \mathbf{x}_i^p, \mathbf{X}_i^n)\in\mathcal{T}} \mathcal{L}_{\text{nt}}(f(\mathbf{x}_i^a, \boldsymbol{\theta}), (f(\mathbf{x}_i^p, \boldsymbol{\theta}), (f(\mathbf{X}_i^n, \boldsymbol{\theta})) \right\}, \tag{10}$$

where the optimization starts from $\boldsymbol{\theta}_s$.

Because $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_t$ are optimized for source-domain data $\mathcal{S}$ and target-domain data $\mathcal{T}$, respectively, the source-domain embedding vectors extracted using $\boldsymbol{\theta}_t$ would perform much worse than the ones extracted using $\boldsymbol{\theta}_s$. In the case of domain adaptation, where we only care about the performance in the target domain, this is acceptable. However, in some scenarios, the model must perform well in both domains. One simple approach is aggregating data from both the source- and target-domain to train a unified model. However, the distribution gap between the two domains could lead to sub-optimal

performance for both. What's more, the amount of target-domain data is often smaller than the source-domain data. Aggregating data from both domains could lead to a model that strongly favors the source-domain data. Another approach is to

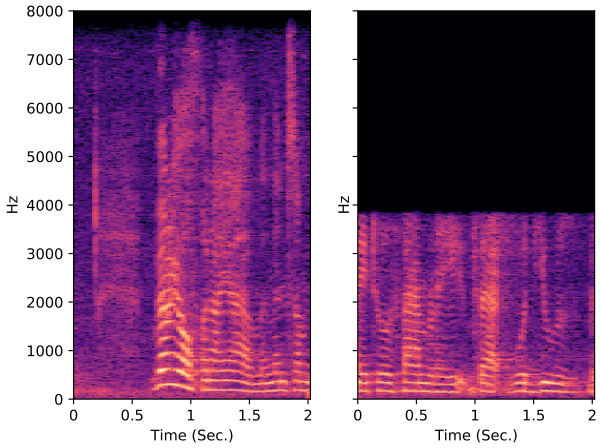

Fig. 2. The spectrograms of a 16kHz microphone speech (left) and an 8kHz telephone speech (right) that was upsampled to 16kHz using Sox.

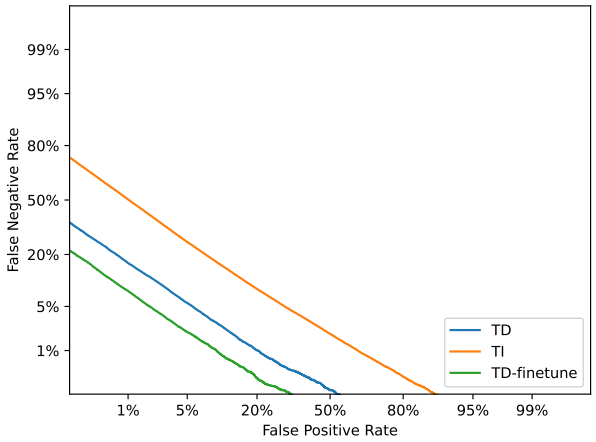

Fig. 3. Detection error tradeoff (DET) curves of a text-dependent (TD) system, a text-independent (TI) system, and a fine-tuned text-dependent system.

interpolate the source and target models. While it is feasible to interpolate linear models (as in Eq. 6), DNNs are highly non-linear. Therefore, interpolating two DNNs generally would not produce sensible results.

Recently, it was found that by carefully choosing a small learning rate, an error minimum of a pre-trained model and an error minimum of a fine-tuned model are connected by a linear path of nonincreasing error in the weight space [13]. By interpolating the parameters of the two models, we can obtain a model that performs well on both domains, i.e., we perform:

$$\boldsymbol{\theta}_{\text{wse}} = (1-\alpha)\boldsymbol{\theta}_s + \alpha\boldsymbol{\theta}_t, \tag{11}$$

where $\alpha \in [0, 1]$ is the interpolating coefficient and is set using a validation set. $\boldsymbol{\theta}_{\text{wse}}$ stands for weight space ensemble. We will show in the following sections that the model with parameters $\boldsymbol{\theta}_{\text{wse}}$ performs well on both the source domain and the target domain.

IV. Domain Shifts and Effect of WSE

Domain shifts frequently happen in speaker verification. The most common domain shifts are language- and channel-related. In the former, a model trained in one language is deployed to another language; in the latter, a model trained by the speech recorded on one device is deployed to the speech recorded by another device. A less obvious domain shift is the shift between text-independent speech and text-dependent speech, which we will explain in detail in the following sub-section.

A. Language-Related Domain Shift

The language-related domain shift is one of the most studied domain shifts in speaker verification [8], [33], [34]. This kind of shift typically happens when a model is trained in one language but deployed in another language. In such a case, the discriminative ability of the speaker embeddings is significantly compromised. To demonstrate this, we trained a model (speaker embedding network) using the English portion of the VoxCeleb [35] corpus (which is referred to as the base model) and used it to produce a set of speaker embeddings from English and Mandarin speech. The first column of Figure 1 shows the t-distributed stochastic neighbor embeddings (T-SNE) of the speaker embeddings obtained from the base model. We can see that the speaker embeddings of English speakers are well separated (bottom row) while the speaker embeddings of Mandarin speakers are mixed (top row).

To enhance the discriminative ability of the model on Mandarin speakers, we fine-tuned the base model using a hold-out portion of Mandarin speech from the CN-Celeb dataset [36] and then used the fine-tuned model to produce the speaker embeddings from the same speakers in the second column of Figure 1. Compared with the first column, we can observe that the embeddings of Mandarin speakers become better separated. However, the between-speaker separation of the English speakers becomes smaller. This result suggests that fine-tuning sacrifices the source domain's speaker discriminative ability for that of the target domain.

Next, we performed weight space ensemble (WSE) on the base and the fine-tuned model and used the WSE model to produce the speaker embeddings from the same set of speakers. The results are shown in the third column of Figure 1. We can see that both Mandarin and English speaker embeddings have distinct speaker clusters. This result shows that the WSE model does not sacrifice the performance of one domain for improving the performance of the other.

B. Cross-Channel Domain Shift

The most prominent cross-channel domain shift is the shift between 8kHz telephone speech and 16kHz microphone speech [37], [38]. Besides the mismatch between the telephone channel and the microphone channel, the difference in sampling frequency also brings a significant mismatch in the spectral domain. Figure 2 shows the spectrograms from a 16kHz microphone channel recording and an 8kHz telephone recording that was upsampled to 16kHz using Sox. We can see that the frequency components above 4kHz are completely lost in the 8kHz recording. This will bring a distribution shift in the acoustic features.

C. Domain Shift Between TI and TD

Although text-independent SV has been the most popular SV research topic for many years, recently, text-dependent SV has attracted a lot of attention [39]–[43]. The domain shift from text-independent data to text-dependent data is less obvious than the domain shifts we mentioned in Section IV-A and Section IV-B. Suppose we have speech data sampled from the distribution $p(\mathbf{x}, y)$, where $\mathbf{x}$ is the acoustic features, and y is the speaker label. Because speech contents in TD-SV are constrained, the speech data are sampled according to the conditional distribution $p(\mathbf{x}, y|\text{text})$. On the other hand, in TI-SV, the speech data are sampled from the marginal distribution $p(\mathbf{x}, y)$. Therefore, the model optimized for the marginal distribution may not be optimal for a conditional distribution. The domain shift from the marginal distribution to the conditional distribution is also detrimental to SV performance. A major advantage of TD-SV is that it can achieve a lower error rate, thanks to the content-match protocol. Figure 3 shows the detection-error-rate trade-off (DET) curves of three systems using the same set of data selected from RSR2015-Part2. The enrollment data and test data for the three systems are the same. But for the TI system, the trials were constructed by randomly choosing the enrollment and test pairs, irrespective of speech contents. For the TD system, the trials were constructed by picking the enrollment and test pairs of the same speech content. We can see from the figure that the TD systems produce significantly lower error rates than the TI systems. What's more, fine-tuning on TD data can further reduce the error rates.

V. Experimental Setup

Throughout the remainder of the paper, we will refer to the models trained on the VoxCeleb dataset as the base model. For different downstream SV tasks, we have different models that are fine-tuned on different datasets. The WSE models were obtained by interpolating the base models and fine-tuned models, as described in Section III.

A. Base Model Training

The base model was trained using VoxCeleb datasets [35], [44]. VoxCeleb1&2 comprises utterances spoken by speakers from a wide range of ethnicities, professions, and ages in various environments, including red carpets, outdoor stadiums, and indoor studios. Specifically, the training data include Voxceleb1-dev and Voxceleb2-dev. The test set of Voxceleb1 was used for performance evaluation. We followed the data augmentation strategy in the Kaldi SRE16 recipe.[2] Specifically, the training data were augmented by adding noise, music, reverb, and babble to the original speech files. After filtering out the utterances shorter than 4 seconds and the speakers with less than 8 utterances, we were left with 7,302 speakers. We used the filter-bank features implemented in Kaldi, with 40 mel-scale filters and cutoff frequencies at 20Hz and 7,600Hz. We used a frame length of 25ms, with a frameshift of 10ms. Mean normalization was applied to the filter-bank features using a 3-second sliding window. Non-speech frames were removed by Kaldi's energy-based voice activity detector.

We experimented with two DNN architectures, namely, the standard x-vector network as in [17] and DenseNet121 as mentioned in Section II-B. The networks were trained using additive margin softmax with a margin of 0.35 and a scaling factor of 30. The networks were optimized using stochastic gradient descent (SGD). For each mini-batch, we randomly selected 64 utterances from the training set, and for each utterance, we randomly cropped a 4-second speech segment from the utterance. As a result, each mini-batch has 64 speech segments, and each segment is of 4 seconds. We defined one epoch as iterating through 120,000 such segments. We trained the networks for 320 epochs. The embedding dimension of x-vectors and wide x-vectors is 512.[3] The embedding dimension of DenseNet121 is 256. The learning rate was set to 0.005 and

TABLE III. The performance of fine-tuned systems using three loss functions mentioned in Section II-C. Each scenario represents a domain shift. For the details of the domain shifts, refer to Section V-D.

Scenario	Loss	Target-domain EER(%)	Target-domain minDCF
TI to TD	NT-Xent	1.96	0.236
	AM-Softmax	1.86	0.243
	Triplet loss	2.67	0.331
Lang. Shift (English → Mandarin)	NT-Xent	9.93	0.545
	AM-Softmax	10.22	0.562
	Triplet loss	12.10	0.735
Channel Shift (16kHz → 8kHz)	NT-Xent	17.15	0.912
	AM-Softmax	17.62	0.932
	Triplet loss	18.80	0.967

TABLE IV. The performance of fine-tuned systems using cosine and PLDA backends. The fine-tuning loss is NT-Xent.

Scenario	Backend	Target-domain EER(%)	Target-domain minDCF
TI to TD	PLDA	1.96	0.236
	Cosine	2.18	0.255
Lang. Shift (English → Mandarin)	PLDA	9.93	0.545
	Cosine	10.43	0.577
Channel Shift (16k) (16kHz → 8kHz)	PLDA	17.15	0.912
	Cosine	17.33	0.924

[2]https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2

[3]The numbers of channels in a wide x-vector network are double that of an x-vector network.

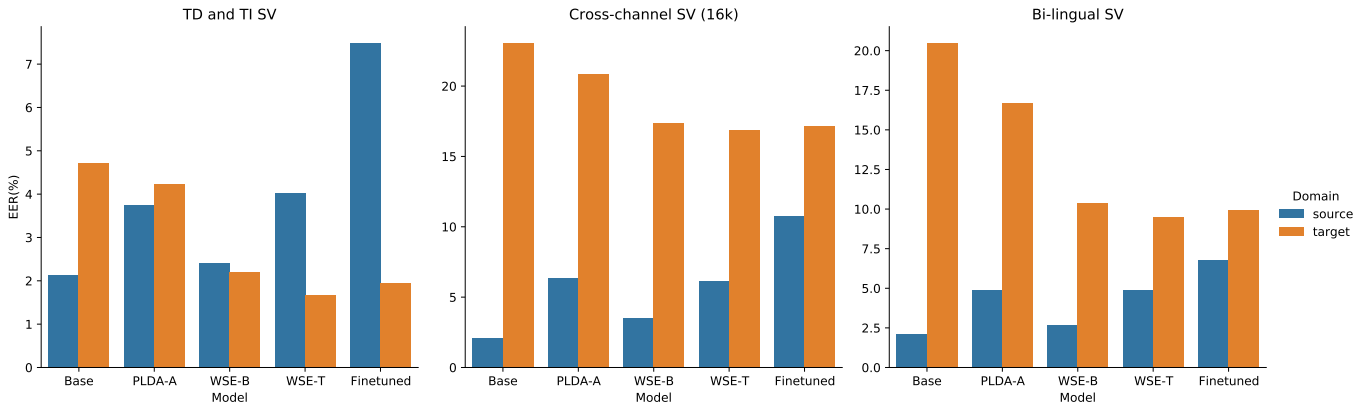

Fig. 4. The EERs of the models on the source- and target-domain data using different adaptation methods. *PLDA-A*: PLDA adaptation; *WSE-B*: WSE balance; *WSE-T*: WSE target; *WSE*: weight space ensemble.

was divided by 10 at Epochs 80, 120, and 160. All networks were implemented in PyTorch [45].

B. Fine-tuning Strategies

For the x-vector networks, we fine-tuned all layers in Table I. The learning rate for Conv1–Conv5 was set to 0.0005, and the learning rate for FC6 and FC7 was set to 0.001. For the DenseNet121 in Table II, we froze the first convolution layer, Dense Block 1, and Transition Layer 1. The learning rate for the remaining frame-level layers (Dense Block 2 – Dense Block 4) was set to 0.005, and the FC layer's learning rate was set to 0.001. When fine-tuning with AM-Softmax, all pre-trained weights were frozen first. Then, we trained the AM-Softmax layer for 30 epochs with a starting learning rate of 0.01 and annealed the learning rate to 0.001 using cosine scheduling. Finally, we fine-tuned the AM-softmax layer together with the pre-trained layers using a learning rate of 0.001 for another 30 epochs.

C. Backend

We used a standard backend comprised of linear discriminant analysis (LDA), length-normalization, and probabilistic linear discriminant analysis (PLDA). In VoxCeleb1 and VoxCeleb2, for each speaker, we concatenated his/her speech from the same video session. These concatenated speech segments were used to train the LDA and the PLDA models. We also presented an ablation study of the PLDA and cosine backends in SectionVI-A.

D. Fine-tuning and Evaluations

Besides domain shift, we are also interested in domain mixing, i.e., the systems are expected to handle the data from both the source and target domains. The three scenarios that we have investigated include (1) text-independent and text-dependent speaker verification (TI- and TD-SV), where the systems are expected to handle both pre-defined speech phrases and speech of arbitrary contents, (2) cross-channel speaker verification, where the systems are expected to handle 16kHz microphone speech and 8kHz telephone speech, and (3) bi-lingual speaker verification, where the systems are expected to handle speech from two languages. For each scenario, we evaluated the models for each domain (source or target) separately. Then, we mixed the source- and target-domain data to create mixed domain data and evaluated the models on the mixed domain data. To ensure that the mixed dataset neither favors the target-domain data nor the source-domain data, we ensured that the numbers of trials from both domains are the same. The details of the three scenarios are as follows.

- **TD and TI SV**. The source-domain in this setting is unconstrained speech (text-independent), while the target-domain is constrained speech (text-dependent). For text-independent evaluation, we used the VoxCeleb1-test set. VoxCeleb1-test consists of 40 speakers speaking unconstrained content. VoxCeleb2-test was used as the validation data for determining the interpolation coefficients in Eq. 6 and Eq. 11. For text-dependent fine-tuning and evaluation, we used Part3 of RSR-2015 [46], which consists of people speaking random digit strings. There are 300 speakers in RSR-2015 Part3. We split them into a training set, a validation set, and a test set, with 200, 20, and 80 speakers in each split. We trained the base models using the procedures described in Section V-A. Then, we fine-tuned the base models using the training split. After fine-tuning, we performed weight space ensemble by choosing the best interpolation coefficient (α in Eq. 11) on the validation set. Then, we evaluated the adapted models on the test split as a performance measure on the target domain. The mixed domain dataset was created by combining the VoxCeleb1 test set and the test split of RSR-2015 Part3. The validation data for the mixed domain data comprise the aggregation of VoxCeleb2-test data and RSR-2015 Part3 validation split.
- **Bi-lingual SV**. The source domain in this setting is English speech, while the target domain is Mandarin

TABLE V. THE PERFORMANCE OF VARIOUS MODELS IN THE MIXED DOMAIN, THE SOURCE-DOMAIN, AND THE TARGET-DOMAIN. IN "TD AND TI SV", THE SOURCE DOMAIN IS UNCONSTRAINED SPEECH (TEXT-INDEPENDENT), AND THE TARGET-DOMAIN IS CONSTRAINED SPEECH (TEXT-DEPENDENT). IN "BI-LINGUAL", THE SOURCE DOMAIN IS ENGLISH SPEECH, AND THE TARGET DOMAIN IS MANDARIN SPEECH. IN "CROSS-CHANNEL SV", THE SOURCE DOMAIN IS 16KHZ MICROPHONE SPEECH, AND THE TARGET DOMAIN IS 8KHZ TELEPHONE SPEECH. THE MIXED DOMAIN DATASETS WERE CREATED BY CONGREGATING THE SOURCE- AND TARGET-DOMAIN DATA. FOR THE DETAILS OF THE CREATION OF MIXED DOMAIN DATASETS, REFER TO SECTION V-D.

		X-vector Network						DenseNet121					
		Mixed domain		Source domain		Target domain		Mixing domain		Source domain		Target domain	
	Model	EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF
TD and TI SV	Base model	2.55	0.289	**2.13**	**0.173**	4.72	0.477	2.21	0.263	**1.31**	**0.112**	4.26	0.532
	Finetuned model	3.16	0.339	7.49	0.481	1.96	0.236	2.98	0.301	8.43	0.606	1.72	0.232
	Weighted model	4.11	0.737	3.38	0.332	13.79	0.949	3.77	0.688	2.48	0.235	11.71	0.842
	WSE target	1.98	0.218	4.03	0.289	**1.67**	**0.222**	1.66	0.182	3.74	0.339	**1.56**	**0.204**
	WSE balance	**1.54**	**0.171**	2.42	0.201	2.20	0.278	**1.31**	**0.153**	1.50	0.141	1.73	0.243
	PLDA adaptation	3.32	0.342	3.75	0.275	4.23	0.455	3.54	0.322	2.45	0.221	2.98	0.274
Bi-lingual SV	Base model	6.09	0.485	**2.13**	**0.173**	20.49	0.725	5.54	0.452	**1.31**	**0.112**	17.44	0.674
	Finetuned model	6.91	0.514	6.80	0.570	9.94	0.541	6.35	0.482	6.89	0.597	9.01	0.561
	Weighted model	6.02	0.983	3.23	0.309	15.58	0.706	5.82	0.774	2.87	0.294	13.21	0.682
	WSE target	5.32	0.342	4.89	0.439	**9.48**	**0.524**	4.92	0.348	3.28	0.295	**8.01**	0.509
	WSE balance	**4.97**	**0.331**	2.68	0.226	10.39	0.567	**4.53**	**0.315**	1.63	0.140	8.76	**0.504**
	PLDA adaptation	6.22	0.487	4.92	0.462	16.70	0.652	5.94	0.461	4.52	0.443	14.31	0.632
Cross-channel SV (up-sample 16kHz)	Base model	10.21	0.683	**2.13**	**0.173**	23.07	0.996	9.54	0.610	**1.31**	**0.112**	23.58	0.996
	Finetuned model	9.33	0.548	10.77	0.798	17.15	0.912	8.71	0.533	4.48	0.490	16.12	0.881
	Weighted model	12.23	0.774	4.32	0.462	19.20	0.926	11.82	0.792	3.11	0.362	17.20	0.837
	WSE target	7.82	0.492	6.17	0.599	**16.86**	**0.855**	6.92	0.477	2.64	0.279	**14.78**	0.836
	WSE balance	**7.36**	**0.457**	3.53	0.404	17.35	0.864	**6.43**	**0.452**	1.57	0.159	16.82	**0.765**
	PLDA adaptation	9.02	0.498	6.35	0.685	20.85	0.933	8.84	0.513	3.45	0.382	18.29	0.901
Cross-channel SV (down-sample 8kHz)	Base model	15.84	0.792	**4.67**	**0.411**	21.61	0.947	15.24	0.763	**4.32**	**0.447**	21.23	0.918
	Finetuned model	14.32	0.774	12.18	0.835	16.08	0.893	13.82	0.733	11.84	0.813	15.47	0.891
	Weighted model	15.62	0.801	7.21	0.509	20.7	0.932	15.32	0.788	7.06	0.485	18.73	0.905
	WSE target	11.76	0.725	8.82	0.544	**15.61**	**0.866**	10.92	0.701	8.14	0.526	**15.11**	**0.823**
	WSE balance	**10.8**	**0.667**	5.09	0.436	16.77	0.902	**10.12**	**0.632**	4.71	0.398	16.01	0.863
	PLDA adaptation	13.54	0.793	8.67	0.553	21.62	0.918	13.04	0.745	8.25	0.549	20.30	0.884

speech. For English speech evaluation, we used the VoxCeleb1 test set. For Mandarin speech evaluation, we used the CN-Celeb1 evaluation set [36]. CN-Celeb is a multi-genre speaker recognition set collected from several Chinese websites. We randomly chose 290 speakers from the CN-Celeb1 development set with 270 and 20 speakers for fine-tuning and validation, respectively. The mixed domain dataset was created by congregating the VoxCeleb1 test set and the CN-Celeb1 test split. The validation data for the mixed domain data comprise the aggregation of VoxCeleb2-test data and CN-Celeb validation split.

- **Cross-channel SV**. The source domain in this setting is microphone speech (originally 16kHz), while the target domain is telephone speech (originally 8kHz). To make speech features from different sampling rates compatible, we conducted experiments on both up-sampled 16kHz speech and down-sampled 8kHz speech using the Sox toolkit.[4] For the evaluation on microphone speech, we used the VoxCeleb1 test set mentioned earlier. For the evaluation on telephone speech, we randomly selected 300 speakers from the SRE04–SRE10 corpora and assigned 200, 20, and 80 of them to the training, validation, and test sets, respectively. The mixed domain dataset was created by congregating the VoxCeleb1 test set and the 80 SRE speakers in the test split. The validation data for the mixed domain data comprise the aggregation of VoxCeleb2-test data and the SRE validation split.

[4]http://sox.sourceforge.net

E. WSE and PLDA Adaptation

We also conducted experiments on PLDA adaptation. We followed the same protocol as WSE (see Section V-D) by using the validation set to select the adaptation parameters.

VI. RESULTS

We report results in terms of equal error rate (EER) and minimum cost function (DCF) with P_{target} = 0.01.

A. Effect of Loss Functions and Backends on Fine-tuning

In this section, we investigate the effectiveness of fine-tuning in mitigating domain shifts. There are three domain shifts that we are interested in, namely, domain shift from unconstrained speech to constrained speech (text-independent to text-dependent), domain shift from one language to the other (Lang. shift), domain shift from 16kHz microphone speech to 8kHz telephone speech (Channel shift).

For each scenario, we trained the networks using three kinds of loss functions, namely, NT-Xent (Eq. 3), AM-Softmax (Eq. 1), and Triplet loss (Eq. 2). Then, a PLDA backend was used to perform scoring. Table III summarizes the result of fine-tuning using these loss functions. The NT-Xent loss has a distinct advantage over the triplet loss for fine-tuning because NT-Xent contrasts every positive sample with multiple negative samples in a batch instead of a single negative

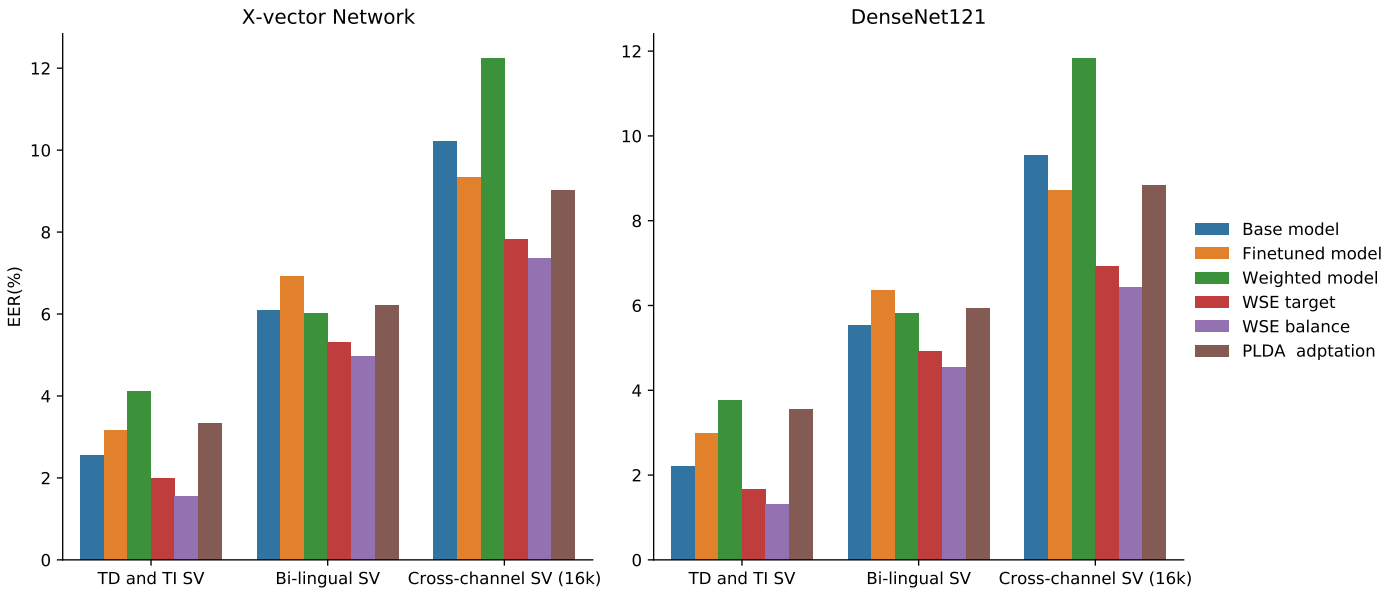

Fig. 5. EER of six models on the mixing domain evaluations. For the details of the creation of the mixed domain, refer to Section V-D.

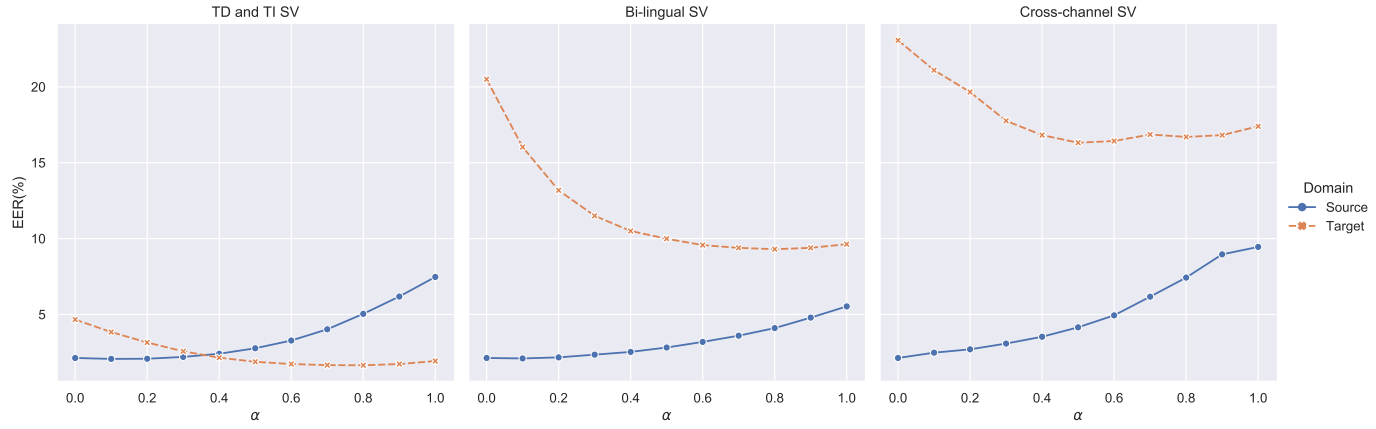

Fig. 6. The effect of the interpolation parameter α in Eq. 11 on WSE models' performance on the source- and target-domains across three domain-shift scenarios.

sample in a triplet. The performance of NT-Xent, however, is close to that of AM-Softmax, and there is no obvious winner. In particular, AM-Softmax wins in the "TI to TD" task, whereas NT-Xent wins in the "Lang. Shift" and "Channel Shift" tasks. Nevertheless, NT-Xent's advantage lies in its non-parametric nature, which avoids setting additional learning rates and schedulers like AM-Softmax. Because of the good performance and simplicity of NT-Xent, we will use NT-Xent loss for all the tasks involving fine-tuning in the rest of the paper.

The backend is an important part of any SV system. Table IV presents the performance of the fine-tuned models with a cosine or PLDA backend. We can see that the PLDA backend, in general, outperforms the cosine backend. Therefore, we will use the PLDA backend for the rest of this paper.

B. The Effect of Domain Shift on SV Systems

In this section, we evaluate the base models and the adapted models on both source-domain and target-domain data under three scenarios, namely, TI and TD SV, cross-channel SV, and bi-lingual SV. A straightforward way to improve a model's performance on mixed domain data is to train a model on pooled data obtained from both the target and source domains. However, source-domain data is typically much more abundant than target-domain data. If we naively pool the data together, the target-domain data would not have much influence. Therefore, we changed the sampling probability of each speaker in the source and target domains so that the total probabilities of the data sampled from both domains are the same. In other words, the target-domain data get sampled more often. We refer to the models trained in this sampling scheme as

"weighted models."

Table V summarizes the performance of the base models, fine-tuned models, weighted models, and weight space ensemble models on both domains for the three scenarios. For the details of the three scenarios, readers can refer to Section V-D. For WSE, we investigated ten different values for the interpolation parameter α, ranging from 0 to 1 with an interval of 0.1. We evaluated the WSE performance on the validation set, and for each scenario, we selected two WSE models, namely, "WSE target" and "WSE balance". "WSE target" means that we selected the model that performs the best on the target-domain validation set. "WSE balance" means that we select the model that performs the best on both the source domain and target domain validation set (smallest total EER). The selected WSE models were then evaluated on the test set of each scenario.

To help readers better understand the results in Table V, the EERs of the x-vector networks in the source-domain and target-domain were visualized in Figure 4, where PLDA-A stands for PLDA adaptation, WSE-B stands for weight-space-ensemble balance, and WSE-T stands for weight-space-ensemble target. The x-axis of Figure 4 is ordered as the base model, PLDA model, WSE-B model, WSE-T model, and fine-tuned model, with the base model being the least optimized toward the target-domain data and the fine-tuned model being the most optimized towards the target-domain data. It is obvious from Figure 4 that as we optimize the models toward the target-domain, the target-domain performance improves, but the source-domain performance degrades. However, the level of degradation is different for the four adapted models (PLDA-A, WSE-B, WSE-T, and fine-tuned), with the fine-tuned model performing the worst in the source-domain and WSE-B performing on par with the base model in the source-domain, which indicates that it is possible to achieve good performance using a single model. Another surprising result is that WSE-T performs better than the fine-tuned model on the target domain even though the fine-tuned model was optimized towards the target-domain data. This could be that the interpolations between the base models and the fine-tuned models have some regularization effect.

C. Mixed Domain Evaluation

In this section, we investigate the adapted models' performance on mixed domain data. Specifically, we are interested in three kinds of mixed domains: mixed TD and TI speaker verification (TI and TD SV), where the contents of enrollment speech or test speech can either be constrained or unconstrained, cross channel speaker verification (Cross-channel SV), where the enrollment and test speech can either be 16kHz microphone speech or 8kHz telephone speech and bi-lingual speaker verification (Bi-lingual SV), where enrollment and test speech can either be English or Mandarin. The EER and minDCF of the mixed domain are shown in Figure 5. We can see that WSE-B is the clear winner in mixed domain data in terms of both EER and minDCF. Also, the performance gap between the x-vector network and DenseNet121 is not as pronounced as in the single domain evaluation, which shows that a larger model cannot solve the problem of domain mixing.

D. Effect of Interpolation Coefficient

If we decide to perform weight space ensemble on all layers, then the only hyper-parameter is the interpolation coefficient α. Figure 6 shows the effect of α on three SV scenarios. Unsurprisingly, as we biased towards the fine-tuned model by increasing α, the performance improvement in the target-domain drops rapidly and saturates quickly; on the other hand, the performance in the source domain degrades slowly at first and then deteriorates rapidly. This behavior implies that there is a sweet spot in the interpolation where the trade-off in the performance is negligible.

VII. Conclusions

In this paper, we proposed using weight space ensemble to solve three challenging problems in speaker verification. Specifically, we showed that the WSE models could perform well on text-dependent and text-independent SV, bi-lingual SV, and cross-channel SV. What's more, we also found that WSE can enhance the performance of the fine-tuned models if desired. In future work, we will explore different loss landscapes and their properties to solve speaker verification problems.

References

[1] Seyed Omid Sadjadi, Timothée Kheyrkhah, Audrey Tong, Craig Greenberg, Douglas Reynolds, Elliot Singer, Lisa Mason, and Jaime Hernandez-Cordero, "The 2016 NIST speaker recognition evaluation," in *Proc. Interspeech*, 2017, pp. 1353–1357.

[2] Weiwei Lin, Man-Wai Mak, and Jen-Tzung 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.

[3] Weiwei Lin, Man-Wai Mak, Na Li, Dan Su, and Dong 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.

[4] Man-Wai Mak and Jen-Tzung Chien, *Machine Learning for Speaker Recognition*, Cambridge University Press, 2020.

[5] Gabriela Csurka, "Domain adaptation for visual applications: A comprehensive survey," *arXiv preprint arXiv:1702.05374*, 2017.

[6] Daniel Garcia-Romero, Alan McCree, Stephen Shum, Niko Brummer, and Carlos Vaquero, "Unsupervised domain adaptation for i-vector speaker recognition," in *Proc. Odyssey*, 2014, pp. 260–264.

[7] Baochen Sun, Jiashi Feng, and Kate Saenko, "Correlation alignment for unsupervised domain adaptation," in *Domain Adaptation in Computer Vision Applications*, pp. 153–171. Springer, 2017.

[8] Kong Aik Lee, Qiongqiong Wang, and Takafumi Koshinaka, "The CORAL+ algorithm for unsupervised domain adaptation of PLDA," in *Proc. ICASSP*, 2019, pp. 5821–5825.

[9] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio, *Deep Learning*, MIT press Cambridge, 2016.

[10] Xiaoyi Qin, Hui Bu, and Ming Li, "Hi-mia: A far-field text-dependent speaker verification database and the baselines," in *Proc. ICASSP*, 2020, pp. 3456–3460.

[11] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio, "Sharp minima can generalize for deep nets," in *Proc. International Conference on Machine Learning*, 2017, pp. 1019–1028.

[12] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger, "Snapshot ensembles: Train 1, get m for free," *arXiv preprint arXiv:1704.00109*, 2017.

[13] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin, "Linear mode connectivity and the lottery ticket hypothesis," in *Proc. International Conference on Machine Learning*, 2020, pp. 3259–3269.

[14] Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt, "Robust fine-tuning of zero-shot models," *arXiv preprint arXiv:2109.01903*, 2021.

[15] Daniel Povey, Xiaohui Zhang, and Sanjeev Khudanpur, "Parallel training of dnns with natural gradient and parameter averaging," *arXiv preprint arXiv:1410.7455*, 2014.

[16] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson, "Averaging weights leads to wider optima and better generalization," in *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence,*. 2018, pp. 876–885, AUAI Press.

[17] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*, 2018, pp. 5329–5333.

[18] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015, pp. 3214–3218.

[19] Simon Prince, Peng Li, Yun Fu, Umar Mohammed, and James Elder, "Probabilistic models for inference about identity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 144–157, 2011.

[20] Hossein Zeinali, Shuai Wang, Anna Silnova, Pavel Matějka, and Oldřich Plchot, "BUT system description to Voxceleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[21] Weiwei Lin, Man-Wai Mak, and Lu Yi, "Learning mixture representation for deep speaker embedding using attention," in *Proc. Odyssey*, 2020, pp. 210–214.

[22] Ling-jun Zhao and Man-Wai Mak, "Channel interdependence enhanced speaker embeddings for far-field speaker verification," in *Proc. ISCSLP 2021)*. IEEE, 2021, pp. 1–5.

[23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.

[24] Hagai Aronowitz et al., "Compensating inter-dataset variability in PLDA hyper-parameters for robust speaker recognition," in *Proc. Odyssey*, 2014, pp. 282–286.

[25] Andrew O Hatch, Sachin S Kajarekar, and Andreas Stolcke, "Within-class covariance normalization for svm-based speaker recognition.," in *Proc. Interspeech 2006*. Citeseer, 2006, pp. 1471–1474.

[26] Mitchell McLaren, Miranti Indar Mandasari, and David A. van Leeuwen, "Source normalization for language-independent speaker recognition using i-vectors," in *Proc. Odyssey 2012*, Haizhou Li, Bin Ma, and Kong-Aik Lee, Eds. 2012, pp. 55–61, ISCA.

[27] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[28] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[29] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. International conference on machine learning*, 2020, pp. 1597–1607.

[30] Patrick Kenny, Themos Stafylakis, Pierre Ouellet, Md Jahangir Alam, and Pierre Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *Proc. ICASSP*, 2013, pp. 7649–7653.

[31] Daniel Garcia-Romero and Carol Y Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. Interspeech*, 2011, pp. 249–252.

[32] Daniel Garcia-Romero and Alan McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *Proc. ICASSP*, 2014, pp. 4047–4051.

[33] Wei-Wei Lin, Man-Wai Mak, Longxin Li, and Jen-Tzung Chien, "Reducing domain mismatch by maximum mean discrepancy based autoencoders.," in *Proc. Odyssey*, 2018, pp. 162–167.

[34] Johan Rohdin, Themos Stafylakis, Anna Silnova, Hossein Zeinali, Lukáš Burget, and Oldřich Plchot, "Speaker verification using end-to-end adversarial language adaptation," in *Proc. ICASSP*, 2019, pp. 6006–6010.

[35] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[36] Lantian Li, Ruiqi Liu, Jiawen Kang, Yue Fan, Hao Cui, Yunqi Cai, Ravichander Vipperla, Thomas Fang Zheng, and Dong Wang, "CN-Celeb: multi-genre speaker recognition," *arXiv preprint arXiv:2012.12468*, 2020.

[37] Konstantin Simonchik, Timur Pekhovsky, Andrey Shulipa, and Anton Afanasyev, "Supervised mixture of PLDA models for cross-channel speaker verification," in *Proc. Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[38] Patrick Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," in *Proc. Odyssey 2010*, 2010, p. 14.

[39] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *Proc. ICASSP*, 2016, pp. 5115–5119.

[40] Yuan Liu, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang, and Kai Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[41] Victoria Mingote, Antonio Miguel, Dayana Ribas, Alfonso Ortega, and Eduardo Lleida, "Knowledge distillation and random erasing data augmentation for text-dependent speaker verification," in *Proc. ICASSP*, 2020, pp. 6824–6828.

[42] Yexin Yang, Shuai Wang, Xun Gong, Yanmin Qian, and Kai Yu, "Text adaptation for speaker verification with speaker-text factorized embeddings," in *Proc. ICASSP*, 2020, pp. 6454–6458.

[43] Themos Stafylakis, Md Jahangir Alam, and Patrick Kenny, "Text-dependent speaker recognition with random digit strings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 7, pp. 1194–1203, 2016.

[44] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.

[45] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in PyTorch," 2017.

[46] Anthony Larcher, Kong Aik Lee, Bin Ma, and Haizhou Li, "The RSR2015: Database for text-dependent speaker verification using multiple pass-phrases," in *Proc. Interspeech*, 2012, pp. 1580–1583.