Discriminative Neural Embedding Learning for Short-Duration Text-Independent Speaker Verification

Shuai Wang, *Student Member, IEEE*, Zili Huang, *Student Member, IEEE*, Yanmin Qian, *Senior Member, IEEE*, and Kai Yu, *Senior Member, IEEE*

***Abstract*—Short duration text-independent speaker verification remains a hot research topic in recent years, and deep neural network based embeddings have shown impressive results in such conditions. Good speaker embeddings require the property of both small intra-class variation and large inter-class difference, which is critical for the ability of discrimination and generalization. Current embedding learning strategies can be grouped into two frameworks: "Cascade embedding learning" with multiple stages and "direct embedding learning" from spectral feature directly. We propose new approaches to achieve more discriminant speaker embeddings. Within the cascade framework, a neural network based deep discriminant analysis (DDA) is proposed to project *i*-vector to more discriminative embeddings. Within the direct embedding framework, a deep model with more advanced center loss and A-softmax loss is used, the focal loss is also investigated in this framework. Moreover, the traditional *i*-vector and neural embeddings are finally combined with neural network based DDA to achieve further gain. Main experiments are carried out on a short-duration text-independent speaker verification dataset generated from the SRE corpus. The results show that the newly proposed method is promising for short-duration text-independent speaker verification, and it is consistently better than traditional *i*-vector and neural embedding baselines. The best embeddings achieve roughly 30% relative EER reduction compared to the *i*-vector baseline, which could be further enhanced when combined with the *i*-vector system.**

***Index Terms*—Short-duration text-independent speaker verification, center loss, triplet loss, angular softmax, speaker neural embedding.**

I. Introduction

The goal of speaker verification (SV) is to verify a speaker's claimed identity given his speech segment. Considering the lexicon constraint on the spoken text, speaker verification can be divided into two categories, text-dependent (TD) and text-independent (TI). The former one requires the same content for the enrollment and test speech, while the latter one has no restrictions on the spoken content. TI-SV is more flexible for real applications but more challenging. To achieve comparable performance with TD-SV, TI-SV usually needs more data so that the impact of the phonetic variation can be implicitly normalized. However, it's not feasible for many real-world applications, especially when asking the user to speak for a long time during the testing. The performance degradation caused by insufficient data is called the short-duration problem. Research on the more challenging short-duration TI-SV is more demanded recently, which is also our focus in this work.

Compared with the traditional Gaussian Mixture Model- Universal Background Model (GMM-UBM) method [1], using a speaker embedding representation with a fixed-dimension vector has become a leading trend for speaker verification. Research on speaker embedding representation can be traced back to the GMM super-vector [2], [3], which tied up different components of the GMM. Kenny *et al.* proposed a Joint Factor Analysis (JFA) [4] framework as a compensation method in the super-vector space, which is then simplified by the *i*-vector [5] approach. *i*-vector is a low-dimensional embedding representation, modeling the speaker and channel related factors in a total variability space.

Deep neural network (DNN) has shown its remarkable capability on structured representation learning and achieved impressive results for many speech processing tasks such as speech recognition [6]–[9], speech enhancement [10], [11] and speaker recognition [12]–[15]. For speaker recognition, it's natural and intuitive to utilize a DNN to learn speaker embeddings. In many current works, DNNs are usually used to extract bottleneck features [15]–[18] or speaker embeddings directly [14], [17], [19]–[21]. *d*-vector proposed in [14] is a typical speaker embedding learning framework. In the *d*-vector approach, a speaker-discriminative DNN is firstly optimized against the softmax loss function,[1] then it is used to extract frame-level vectors from the last hidden layer. These vectors are averaged over the whole sentence to obtain the utterance-level representation, i.e. *d*-vector. To better match the training phase and evaluation phase, researchers improve *d*-vector from two aspects: (1) a temporal pooling layer is incorporated into the neural network,

Manuscript received June 11, 2018; revised December 27, 2018 and April 24, 2019; accepted June 24, 2019. Date of publication July 11, 2019; date of current version August 1, 2019. This work was supported by the China NSFC project under Grants 61603252 and U1736202. The Associate Editor coordinating the review of this manuscript and approving it for publication was Prof. N. Dehak. *(Corresponding authors: Yanmin Qian; Kai Yu.)*

The authors are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: feixiang121976@sjtu.edu.cn; huangziliandy@sjtu.edu.cn; yanminqian@gmail.com; kai.yu@sjtu.edu.cn).

Digital Object Identifier 10.1109/TASLP.2019.2928128

[1] Noted that following the work in [22], the softmax loss means the combination of the output layer, softmax function and cross entropy loss.

so that the frame aggregation operation can be removed and the utterance-level representation will be obtained directly [23], [24]; (2) better training losses defined on tuples or triplets [25]–[27] are introduced into the end-to-end neural network architecture to get a better performance. However, neural networks with these end-to-end losses are non-trivial for optimization, the appropriate selection on tuples or triplets is critical to get a good system performance, which is usually difficult to be guaranteed.

A speaker embedding with good generalization ability to unseen speakers should have small intra-speaker variation and large inter-speaker difference. *With the "good" enough speaker embeddings, simple scoring methods such as cosine-distance scoring (CDS) will work well.* However, learning a speaker embedding with good quality is difficult, especially under a complex environment such as short-duration and noisy conditions. Thus, an additional post-processing step such as Linear Discriminant Analysis (LDA) or Probabilistic Linear Discriminant Analysis (PLDA) is usually used to get an enhanced speaker embedding. Such post-processing could also be implemented using more complicated methods such as deep neural networks[23], [28]–[31]. For instance, authors in [23] introduced two neural network-based systems, Non-linear Within Class Normalization (NWCN) and Speaker Classifier Network(SCN) to compensate in the *i*-vector space, impressive results were obtained SRE2010 evaluation dataset. Authors in [28] proposed to use Denoising Auto-encoder to perform domain adaptation and achieved good results. To deal with the uncertainty of short-duration *i*-vector systems, authors in [29] proposed to use a Convolutional Neural Network (CNN) to map the short-duration *i*-vector to its corresponding long version. Most of the existing compensation methods focus on reconstructing the embeddings in a generative paradigm, while discriminative DNN based approaches are proposed in this work.

Most of the current speaker embedding learning architectures can be grouped into two classes: (1) **cascade embedding learning** has multiple stages in the whole framework, such as the embeddings generated from the procedure: Gaussian super-vector $\mapsto$ *i*-vector $\mapsto$ LDA; (2) **direct embedding learning** learns from the spectral feature using one model directly, such as the *d*-vector and *x*-vector generated from a deep model.

In this paper, several new techniques are proposed to improve for the two frameworks, which can learn a more discriminative speaker embedding.

1) *Cascade Embedding Learning:* sharing the motivation of Linear Discriminant Analysis (LDA), a non-linear discriminant analysis using deep models is learned to transform *i*-vectors to embeddings with larger inter-speaker difference and smaller intra-speaker variation. To achieve this goal, center loss and angular softmax (A-softmax) loss are introduced to replace the traditional softmax loss.
2) *Direct Embedding Learning:* embeddings are learned from spectral features directly through a deep neural network. In our new strategy, the usual softmax or triplet loss is replaced with a more advanced center loss or angular softmax loss. Moreover, the focal loss is incorporated to make the learning more effective.

The main contributions of this paper are summarized as follows,

- Two different discriminative loss functions, center loss and angular softmax loss, are introduced to speaker embedding learning in two paradigms.
- Deep discriminant analysis is proposed as a non-linear compensation method in *i*-vector space and simple cosine distance scoring on compensated embeddings can achieve a better performance than the *i*-vector/PLDA baseline in our experiments.
- Direct learning speaker embeddings from spectral features under the supervision of center loss and angular softmax loss significantly improve the performance.
- The application of focal loss to softmax loss, center loss and A-softmax loss are investigated and analyzed.
- Besides the main focus on short-duration speaker verification, the effectiveness of proposed methods are also verified on other different duration conditions, where consistent performance improvement can be observed.

The remainder of this paper is organized as follows. Section II revisits the methods on embedding based speaker verification, including both cascade embedding learning and direct embedding learning frameworks. Section III introduces a new cascade embedding learning approach with our proposed deep discriminant analysis (DDA). Two discriminative losses, center and A-softmax loss are utilized in the proposed DDA. The end-to-end deep model based direct embedding learning with new optimization criteria is described in Section IV. Detailed experimental results and analysis are compared and discussed in Section V. Section VI concludes the whole paper.

II. Embedding Learning for Speaker Verification

The procedure of embedding based speaker verification is shown in Figure 1. Each utterance is encoded as a single embedding via different algorithms, such as factor analysis (*i*-vector) and neural networks (neural embeddings). Due to the limited power of the model or training criteria, the learned embeddings may not be discriminative enough. Therefore, post-processing methods, such as LDA, are used to restrict the intra-speaker variation and enlarge the inter-speaker difference, which can be regarded as an additional stage of embedding learning (termed as enhanced embedding learning). The final obtained embeddings can be scored with simple measures such as CDS.

A. Cascade Embedding Learning

As depicted in Figure 1, embedding learning may contain several learning stages, such a learning paradigm is named as cascade embedding learning. *i*-vector/LDA/PLDA framework can be grouped into this category. In the *i*-vector framework [5], the speaker- and session-dependent super-vector $\mathbf{M}$ (derived from UBM) is modeled as

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w} \tag{1}$$

where $\mathbf{m}$ is a speaker and session-independent super-vector, $\mathbf{T}$ is a low rank matrix which captures speaker and session variability, and *i*-vector is the posterior mean of $\mathbf{w}$.

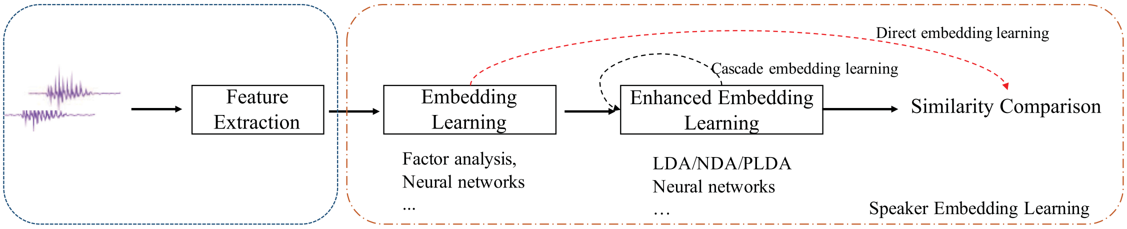


Fig. 1. Embedding based speaker verification. Frame-wise features are encoded to utterance-wise speaker embeddings directly (direct embedding learning) or enhanced to obtain more discriminative embeddings through post-processing learning steps (cascade embedding learning).

The *i*-vector approach has dominated the speaker recognition area for nearly one decade. *i*-vector ties different Gaussian components and learns a low-dimensional speaker space. In the *i*-vector modeling, a set of basis vectors are determined to represent different speakers, which generalizes well to unseen speakers. However, as a linear factor model, *i*-vector clusters embeddings from the same speaker in an unsupervised manner and is vulnerable to short-duration recordings. A second stage of discriminative embedding learning such as LDA or PLDA[2] is usually performed to project *i*-vector to a more discriminative embedding space. In this paper, we propose a neural network based enhanced embedding learning approach, which will be described in Section III.

B. Direct Embedding Learning

A straightforward question is that can we merge multiple stages in cascade embedding learning and learn discriminative embeddings in an end-to-end manner? To achieve this goal, two conditions need to be met. First, the learning machine should hold enough modeling capacity, whereas the deep neural network is an ideal candidate. Second, the model needs to be optimized against proper loss functions to obtain discriminative embeddings, while softmax loss is the most commonly used one. Researchers have made many efforts on learning speaker embedding using neural networks, whereas a typical example is *d*-vector [14], which has shown promising results when combined with *i*-vector. As mentioned in Section I, the frame-wise trained *d*-vector suffers from the mismatch between the training and evaluation phase. Many researchers follow this direction and several kinds of improvements were made. Improvements can be categorized as follows:

- New architectures with stronger feature learning capability are adopted instead of the basic DNN. Models such as CNN, LSTM and TDNN show better performance [23], [32], [33].
- Add temporal pooling layers to perform utterance-level optimization. The temporal pooling operation can be the simple mean and standard variation computations [24], [26], [27], attention mechanism [23], [34] or even more complicated methods such as learnable dictionary encoding (LDE). [35].

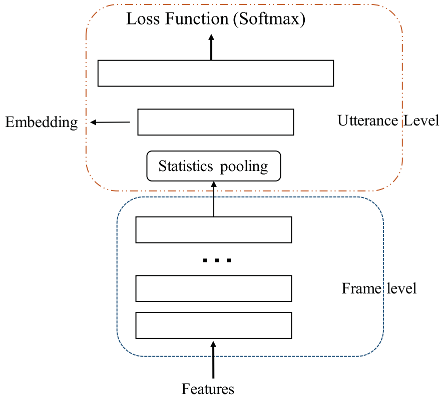


Fig. 2. Softmax loss based utterance-level embeddings.

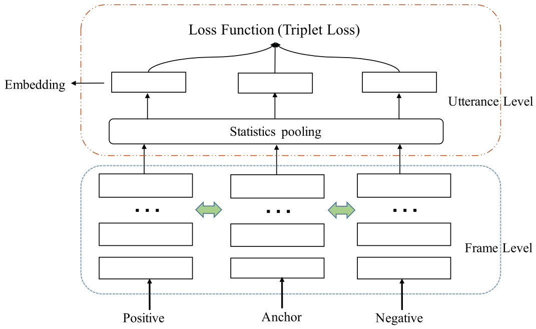


Fig. 3. Triplet-loss based speaker embeddings.

- New optimization metrics to learn more robust speaker embeddings with a better generalization capability [25], [36]–[39].

It has been shown that the utterance-wise training of the neural network can be very useful [24]. In our experience, we will use the utterance-wise training architecture with the temporal pooling operation, as shown in Figure 2. Usually, two types of optimization metrics are applied, i.e., softmax and triplet loss.

1) Softmax-Based Speaker Embeddings: Softmax is the most commonly used classification loss function, which is formulated as

$$P_{y_i} = \frac{e^{\mathbf{w}_{y_i}^\top \mathbf{x}_i + b_{y_i}}}{\sum_j e^{\mathbf{w}_j^\top \mathbf{x}_i + b_j}} \tag{2}$$

$$\mathcal{L}_{softmax} = -\frac{1}{N}\sum_{i}^{N} \log P_{y_i} \tag{3}$$

[2] PLDA can be regarded as a combination of embedding learning and likelihood scoring.

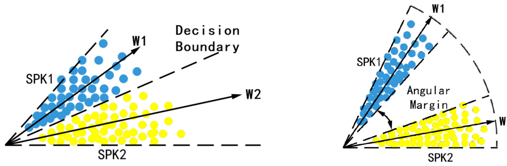


Fig. 4. Decision boundaries learned by Softmax and A-softmax loss. Left: Decision boundaries learned by Softmax loss. Right: Decision boundary with a predefined margin learned by A-softmax loss.

where N is the number of samples, $\mathbf{x}_i$ is the deep feature of the i-th sample and y_i is the corresponding label index. $\mathbf{W}$ is the parameter of the last fully connected layer and $\mathbf{b}$ is the bias term. $\mathbf{w}_j$ denotes the weight vector associated with the j-th speaker, and can be treated as the embedding center of that speaker. The softmax based SV system is shown in Figure 2. The deep neural network takes the spectral features as the input. After several frame-level layers, a temporal pooling layer aggregates the frame-level features from the same utterance to a single utterance representation. Compared with the classical *d*-vector [12], the DNN is now trained at utterance level, which is more natural. Two temporal pooling operations can be utilized: 1) mean alone 2) concatenation of mean and standard deviation.

In the evaluation phase, the softmax layer is removed and speaker embeddings are extracted from the embedding layer.

2) Triplet-Loss Based Speaker Embeddings: Triplet loss is designed for verification tasks, it takes in three inputs, including an anchor (an utterance from a specific speaker), a positive sample (an utterance from the same speaker) and a negative sample (an utterance from a different speaker). The goal of the learning process is to reduce the distance between the positive and anchor, while increasing the distance between the negative and anchor sample. The loss $\mathcal{L}$ for an utterance triplet (u^a, u^p, u^n) is defined as

$$\begin{aligned}\mathcal{L}(u^a, u^p, u^n) = [&\|\mathcal{F}(u^a) - \mathcal{F}(u^p)\| \\ &- \|\mathcal{F}(u^a) - \mathcal{F}(u^n)\| + \alpha]_+ \end{aligned} \tag{4}$$

where $\mathcal{F}(u)$ denotes the embedding of the utterance u, α is an empirically defined margin enforced between positive and negative pairs and the operator $[x]_+ = max(x, 0)$. $\|\mathcal{F}(u_1) - \mathcal{F}(u_2)\|$ denotes the Euclidean distance between two embeddings $\mathcal{F}(u_1)$ and $\mathcal{F}(u_2)$. The total loss is the sum of loss computed on all triplets. Similar to the pre-described softmax embeddings, the triplet-based embeddings are extracted from the embedding layer of the well trained neural network. Triplet loss based speaker verification system is often named as the end-to-end system since its optimization goal is similar to the final evaluation metric and the trained neural network can directly output scores given a test triplet. This approach needs careful triplet preparation, which is critical to system performance.

III. Cascade Embedding Learning With Deep Discriminant Analysis

As described in the previous section, cascade learning incorporates an enhanced embedding learning stage such as LDA to project the *i*-vector to a more discriminative embedding space. Sharing the spirit of LDA, an NN based approach named as deep discriminant analysis (DDA) is proposed in this section.

A. Linear Discriminative Analysis (LDA)

LDA is widely used in pattern recognition tasks such as image recognition [40] and speaker recognition [41]. LDA calculates a matrix $\mathbf{W}$ that projects high dimensional feature vectors with $\mathbf{x}$ (*i*-vectors in this paper) into a lower-dimensional and more discriminative subspace ($\mathbf{W} : \mathbb{R}^h \mapsto \mathbb{R}^l$). The projection can be represented as:

$$\mathbf{y} = \mathbf{W}^\top \mathbf{x} \tag{5}$$

where $\mathbf{y}$ denotes the compensated embedding and $\mathbf{W}$ is a rectangular matrix of shape $h \times l$. $\mathbf{W}$ is determined by

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \frac{\text{tr}(\mathbf{W}^\top \mathbf{S}_b \mathbf{W})}{\text{tr}(\mathbf{W}^\top \mathbf{S}_w \mathbf{W})} \tag{6}$$

$$= \arg\max_{\mathbf{W}} [\text{tr}(\mathbf{W}^\top \mathbf{S}_b \mathbf{W})] \tag{7}$$

$$\text{s.t.} \quad \mathbf{W}^\top \mathbf{S}_w \mathbf{W} = \mathbf{I} \tag{8}$$

$\mathbf{S}_b$ and $\mathbf{S}_w$ are the between-class and within-class covariance matrices respectively, and they can be computed as

$$\mathbf{S}_b = \frac{1}{N} \sum_{s=1}^{S} N_s (\boldsymbol{\mu}_s - \boldsymbol{\mu})(\boldsymbol{\mu}_s - \boldsymbol{\mu})^\top \tag{9}$$

$$\mathbf{S}_w = \frac{1}{N} \sum_{s=1}^{S} \sum_{i=1}^{N_s} (\mathbf{x}_i^s - \boldsymbol{\mu}_s)(\mathbf{x}_i^s - \boldsymbol{\mu}_s)^\top \tag{10}$$

where S represents the total number of speakers, and N represents the total number of *i*-vectors from all speakers. $\boldsymbol{\mu}$ represents the global mean of all N *i*-vectors, whereas $\boldsymbol{\mu}_s$ represents the mean of *i*-vectors from the specific s-th speaker. $\mathbf{x}_i^s$ represents the *i*-vector of the i-th utterance from the s-th speaker, and N_s is the number of utterances from the s-th speaker.

LDA has an analytic solution and the optimized $\hat{\mathbf{W}}$ is a matrix whose columns are the l eigenvectors corresponding to the largest eigenvalues of $\mathbf{S}_w^{-1} \mathbf{S}_b$. However, despite its simpleness and effectiveness, LDA assumes the data follows the Gaussian distribution and learns a simple linear transform, which limits its performance in a complicated situation.

B. Probabilistic Linear Discriminant Analysis (PLDA)

Researchers in [42], [43] introduce probabilistic linear discriminant analysis (PLDA) as a back-end for *i*-vector. Several variants of PLDA have been investigated into the speaker verification task, including the standard PLDA [44], two-cov PLDA [42], heavy-tailed PLDA [43] and the Simplified PLDA [43], [45]. In this paper, the two-cov PLDA implemented in Kaldi [46] is adopted, where *i*-vector $\mathbf{x}$ is assumed to be generated as,

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{A}\mathbf{y} \tag{11}$$

$$\mathbf{y} \sim \mathcal{N}(\mathbf{v}, \mathbf{I}) \tag{12}$$

$$\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}) \tag{13}$$

where $\mathbf{v}$ represents the class (speaker), and $\mathbf{y}$ represents a sample of that class in the projected space. $\boldsymbol{\mu}$ is the global mean and

$\mathbf{A}$ denotes the transform matrix learned. Kaldi-PLDA (will be simply refered to as PLDA) is trained using EM algorithm, and more training and inference details can be found in [47] or the Kaldi recipe [46].

PLDA can also be regarded as an embedding learning process, where a newly projected embedding $\mathbf{y} = \mathbf{A}^{-1}(\mathbf{x} - \boldsymbol{\mu})$ is learned.

C. Deep Discriminant Analysis (DDA)

Both LDA and PLDA learn a linear transform function to project raw *i*-vectors onto a more discriminative space. This can be writen to a more general form as below,

$$\mathbf{y} = \mathcal{G}(\mathbf{x}) \tag{14}$$

The function $\mathcal{G}()$ can denote any enhanced embedding learning process, including the usual LDA/PLDA transformation. Instead of the linear projection, $\mathcal{G}$ can also be nonlinear and deep neural networks can be adopted. Such an NN based embedding learning approach is named as Deep Discriminant Analysis (DDA) in this work. Projection learned by DDA has two advantages: it poses no restrictions on the data distribution and learns a more complex non-linear transformation. Different from the common softmax and triplet loss described in the previous section, two more discriminative losses are introduced to force the learned embeddings to have smaller intra-speaker variation and larger inter-speaker difference.

D. Center Loss

Center loss [48] is formulated as

$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{c}_{y_i}\|^2 \tag{15}$$

where $\mathbf{c}_{y_i}$ represents the center of y_i-th class (which the i-th sample belongs to) and is updated along with the training procedure, $\| * \|$ denotes the L2 norm. The neural network will be trained under the joint supervision of softmax loss and center loss. The joint loss is referred to as $\mathcal{L}_{center}$

$$\mathcal{L}_{center} = \mathcal{L}_{softmax} + \lambda\mathcal{L}_C \tag{16}$$

Where λ is adopted for balancing the two loss functions. Intuitively, the softmax loss forces the learned embeddings of different classes far from each other, while the center loss pulls the embeddings from the same class close to their centers. With the joint supervision of softmax loss and center loss, the neural network enlarges the inter-class differences and reduces the intra-class variations of the learned embeddings. In the real implementation the centers can be computed in an online mode [48] or an offline mode [49]. The former treats the center of the embeddings as the learnable parameters and the latter computes the statistics using the results from the previous epoch. In this paper, we follow the online updating rule as described in [48], and learn the center embedding through the standard back-propagation algorithm.

E. A-Softmax Loss

In the previous work [48], it's observed that the embeddings learned by the softmax loss are angularly distributed, which inspires the creation of A-softmax loss to extract angular discriminative features.

In the softmax loss function, if we constrain the parameters of the last fully connected layer to have $\|\mathbf{w}_j\| = 1$ and $b_j = 0$, it becomes the modified softmax loss,

$$P_{y_i} = \frac{e^{\|\mathbf{x}_i\|\cos(\theta_{y_i,i})}}{\sum_j e^{\|\mathbf{x}_i\|\cos(\theta_{j,i})}} \tag{17}$$

$$\mathcal{L}_{modified} = -\frac{1}{N}\sum_{i}^{N} \log P_{y_i} \tag{18}$$

where $\theta_{j,i}$ denotes the angle between $\mathbf{w}_j$ and $\mathbf{x}_i$. As Equation (17) indicates, the probability of an utterance i belongs to a speaker j is only determined by the angle between them. The optimization goal is to learn the separation boundary between different classes in the training stage.

The modified softmax shows that the learned features are angularly separable, however, they are not discriminative enough for open-set problems like speaker verification. A-softmax loss was proposed to use more stringent requirements on the the separation boundary. The modified softmax loss classifies an utterance i into the corresponding speaker y_i if $\forall k \neq y_i, \cos(\theta_{y_i,i}) > \cos(\theta_{k,i})$. A-softmax loss requires $\forall k \neq y_i, \cos(m\theta_{y_i,i}) > \cos(\theta_{k,i})$ where m is an integer and $m \geq 2$. We formulate this idea into the modified softmax loss directly, and we can obtain

$$P_{y_i} = \frac{e^{\|\mathbf{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\mathbf{x}_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|\mathbf{x}_i\|\cos(\theta_{j,i})}} \tag{19}$$

$$\mathcal{L}_{asoftmax} = -\frac{1}{N}\sum_{i}^{N} \log P_{y_i} \tag{20}$$

However, it is required that $\theta_{y_i,i} \in [0, \frac{\pi}{m}]$ because the cosine function is not monotonic. This constraint can be removed if we replace it with a monotonic function $\psi(\theta_{y_i,i}) = (-1)^k \cos(m\theta_{y_i,i}) - 2k$, $\theta_{y_i,i} \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}]$ and $k \in [0, m-1]$. When $m = 1$, A-softmax loss is equivalent to the modified softmax loss. The definition of Equation (19) then becomes

$$P_{y_i} = \frac{e^{\|\mathbf{x}_i\|\psi(\theta_{y_i,i})}}{e^{\|\mathbf{x}_i\|\psi(\theta_{y_i,i})} + \sum_{j\neq y_i} e^{(\|\mathbf{x}_i\|\cos(\theta_{j,i}))}} \tag{21}$$

IV. Direct Embedding Learning Using Deep Models

In the previous section, a cascade embedding learning framework with deep discriminative analysis (DDA) was proposed, where two enhanced criteria are introduced to optimize the deep model. On the other hand, it's also intuitive to use deep models to directly learn neural embeddings from spectral features in an end-to-end manner. Accordingly, a direct discriminative neural embedding learning framework with the two mentioned criteria is proposed in this section.

A. Generalization Ability of Softmax Based Embedding Learning

Different from speech recognition where phonemes (senones) to recognize have appeared in the training phase, speaker recognition is naturally an open-set problem, there is usually no overlap in speakers in the training and evaluation dataset. Therefore, softmax supervised speaker embedding learning seems not able to generalize well on unseen speakers. The neural embedding learning system for speaker verification can be divided into two parts, the before-embedding part $\mathcal{F}^-$ and the after-embedding part $\mathcal{F}^+$. $\mathcal{F}^-$ encodes the input feature $\mathbf{o}$ into the neural embedding $\mathbf{x}$ via $\mathbf{x} = \mathcal{F}^-(\mathbf{o})$, while in our setting $\mathcal{F}^+$ projects the embedding to the output vector $\mathbf{y}$. We can write the last affine layer as,

$$\mathbf{y} = \mathcal{F}^+(\mathbf{x}) \tag{22}$$

$$= \mathbf{b} + \mathbf{W}\mathbf{x} \tag{23}$$

Assume $\mathbf{x} \in \mathbb{R}^D$ and the output vector $\mathbf{y} \in \mathbb{R}^C$, $\mathbf{y}$ will then be passed to the softmax function. $\mathbf{W}$ is the weight matrix of shape $D * C$. If $D \geq C$, the network is likely to learn a trivial representation, such as a one-hot embedding. In most speaker embedding learning systems, we have $D << C$, $\mathbf{rank}(\mathbf{W}) \leq D$. Similar to the concept of basis in *i*-vector framework, $\mathbf{W}$ also encodes a latent factor space and is the factor loading matrix. $\mathbf{x}$ represents the factors. One difference with *i*-vector is that *i*-vector is trained to learn the basis in a generative manner, while the neural embedding learning is supervised by discriminative signals. Thus, we expect that the learned neural embeddings encode some intrinsic properties of speakers and can generalize to unseen speakers to some extent. However, the normal softmax loss neither ensures the margin between different classes nor restrict the within-class variation, which limits its generative power and makes the learning easily converge to a local minimum.

B. Focal Loss

To make the neural network converge to a better optimum, different refinement can be used in the training process. For example, in the triplet loss based system, the hard-trial selection operation will be used. Strategies such as curriculum learning have also been proven very useful for both deep neural network training [50] and statistical models such as PLDA [51].

In this work, the focal loss is introduced to ease model optimization. Focal loss tries to optimize the model against more easily misclassified samples, firstly used in [52] for the image object detection task. The focal version of softmax loss is defined as

$$P_{y_i} = \frac{e^{\mathbf{w}_{y_i}^\top \mathbf{x}_i + b_{y_i}}}{\sum_j e^{\mathbf{w}_j^\top \mathbf{x}_i + b_j}} \tag{24}$$

$$L_{fsoftmax} = -\frac{1}{N}\sum_i^N (1 - P_{y_i})^\gamma \log P_{y_i} \tag{25}$$

Compared to Equation (3), the modulating factor $(1 - P_{y_i})^\gamma$ is added. It should be noted that the hyper-parameter α in the original paper [52], which balances different classes, is omitted since we treat each class equally. There are two properties about the focal loss: 1) when a sample is misclassified and P_{y_i} is small, the modulating factor will approach 1.0 and the related loss is unaffected. In contrast, for those well-classified samples with P_{y_i} approaching 1.0, the corresponding loss will be down-weighted. 2) The adjustable hyper-parameter γ controls the weighting scale of samples, and it will become the normal softmax loss when γ is set to 0. The same idea can be applied to many loss functions, while the key point is to define the "hardness" of samples and a weighting function which is negatively correlated with the "hardness". For instance, we extended the focal loss to multi-class classification problem to deal with the co-channel speaker identification task in [53].

Correspondingly, by substituting the softmax part in the loss function to its focal version, we get the focal center loss

$$\mathcal{L}_{fcenter} = \mathcal{L}_{fsoftmax} + \lambda\mathcal{L}_C \tag{26}$$

Noted that we only change the softmax part to the focal version in this paper for simplicity, while the center loss part can also be changed to the focal version.

Similarly a focal version of A-softmax loss is obtained by changing the Equation (20) to Equation (27):

$$\mathcal{L}_{fasoftmax} = -\frac{1}{N}\sum_i^N (1 - P_{y_i})^\gamma \log P_{y_i} \tag{27}$$

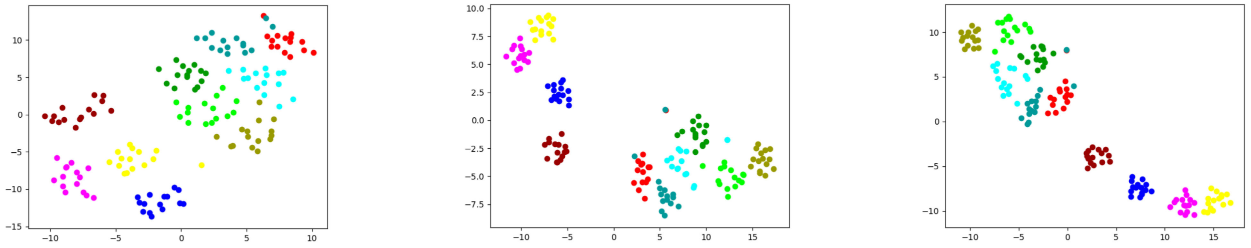
The learned neural embeddings are extracted from the embedding layer as the same architecture in Figure 2, but with more advanced criteria, then they will be scored with simple CDS directly.

V. Experiments and Results Analysis

In this section, experiments on the short-duration text-independent speaker verification are performed and compared. The main results will be divided into two parts, experiments on cascade embedding learning (DDA) and direct neural embedding learning.

A. Dataset

The short-duration text-independent dataset is generated from the NIST SRE corpus. The training set consists of selected data from SRE04-08, Switchboard II phase 2, 3 and Switchboard Cellular Part1, Part2. After removing silence frames using an energy-based VAD, the utterances are chopped into short segments (ranging from 3–5s). The final training set contains 4000 speakers and each speaker has 40 short utterances. The enrollment set and test set are derived from NIST SRE 2010 following a similar procedure. The enrollment set contains 300 speakers (150 males and 150 females) and each speaker is enrolled by 5 utterances. The test set contains 4500 utterances from the same 300 speakers in the enrollment set. The created trial list contains 392660 trials. There are 15 positive samples and 1294 negative samples for each model on average. No cross-gender trial exists. The detailed segmentation files and the trial list can be accessed by the website https://github.com/wsstriving/DEL_Segments.git.



(a) *i*-vector

(b) embedding: DDA (center)

(c) embedding: DDA (A-softmax)

Fig. 5. Visualization of compensated speaker embeddings. (a) Raw *i*-vector. (b) Embedding compensated with DDA (center). (c) Embedding compensated with DDA (A-softmax). (**Best viewed in color**).

TABLE I
EER (%) OF DIFFERENT COMPENSATION METHODS, AND THE DIMENSION OF PROJECTED EMBEDDINGS IS 300

Methods	EER (%)	minDCF08	minDCF10	minCprimary
ivec-CDS	6.8	0.5062	0.9888	0.9346
ivec-PLDA	4.96	0.3336	0.9706	0948
ivec-LDA-CDS	5.67	0.3696	0.9613	0.8001
ivec-DDA-CDS (Softmax)	4.67	0.2886	0.8594	0.6647
ivec-DDA-CDS (Center)	**4.13**	0.2487	0.8426	0.611
ivec-DDA-CDS (A-softmax)	4.18	**0.2285**	**0.7279**	**0.5094**

B. Baseline Settings

The baseline *i*-vector system is implemented using the Kaldi toolkit [46]. 20-dimensional Mel-frequency cepstral coefficients (MFCC) with their first and second order derivatives are extracted from the speech segments, which is obtained with an energy-based VAD. A 25 ms Hamming window with a 10 ms frame shift is adopted in the feature extraction process. The universal background model (UBM) contains 2048 Gaussian mixtures and the *i*-vector dimension is set to 400. Different scoring methods are applied to the length-normalized *i*-vectors. Equal error rate (EER), minDCF08, minDCF10 and minCprimary (NIST SRE 16) [54] are used as the evaluation metrics.

C. Cascade Embedding Learning from i-Vectors

As discussed in Section III, for embeddings which are not discriminative enough, an additional stage of embedding learning will be performed.

The proposed neural network based DDA comprises a single hidden layer followed by Rectified Linear Units (ReLU) activation. As shown in Table I, compared to LDA, DDA with two newly proposed losses obtains a large improvement with simple Cosine Distance Scoring (CDS). Furthermore, both methods outperform the *i*-vector/PLDA baseline. To show the effectiveness of the proposed losses, we also include the DDA trained with normal softmax loss as a baseline, which also obtains better performance than the basic *i*-vector systems. The best EER 4.13% is achieved by DDA (Center), while DDA (A-softmax) obtains best results in the other DCF related metrics. To better understand the proposed method, we use t-SNE [55] to visualize the *i*-vectors and their corresponding DDA-compensated embeddings in Fig. 5.

TABLE II
DETAILED NEURAL NETWORK CONFIGURATION

Input	36 × 17 Feature Map (17-frame context)		
Conv Layers	filter size	padding	channels
1	3×3	1	24
2	3×3	1	32
3	3×3	1	32
4	3×3	1	16
FC Layer	400 nodes		
Temporal Pooling	Mean (400 dim)	Mean+STD (800 dim)	
Embedding Layer	400 nodes		
Output	4000 nodes		

Figure 5(a) depicts the distribution of *i*-vectors from 10 speakers randomly chosen from the test set, while the distribution of corresponding compensated embeddings using two types of DDA are shown in Figure 5(b) and 5(c), respectively. As shown in the figures, using the proposed DDA with two new criteria, the distribution of embeddings from the same speaker seems more compact, which means the intra-speaker variation is significantly reduced. The discriminative neural embedding learning strategy acts as a compensation method in the *i*-vector space.

D. Direct Embedding Learning From Spectral Features

As described in Section IV, instead of learning an enhanced neural embedding from *i*-vectors in a cascade manner, center loss and A-softmax loss can provide strong supervision signals for direct embedding learning from spectral features.

The softmax, triplet loss and A-softmax based systems adopt the same neural network architecture illustrated in Figure 2. It's a VGG-style CNN with 4 convolution layers, 2 max-pooling layers (after the 2nd and 4th convolutional layer) and 1 fully-connected layer to produce the frame-level embeddings. Detailed neural network settings can be found in Table II. ReLU is used as the activation function after each parameterized layer (except the output layer). Frame-level deep features are averaged to utterance embeddings via a temporal pooling layer. The embedding dimension is set to 400 in all experiments.

We train the triplet loss based system with the same configuration and strategy as in our previous work [27]. For the softmax based system, we set the initial learning rate as 0.01 and adjust it

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT SPEAKER EMBEDDINGS USING MEAN AND MEAN+STD POOLING (CONCATENATION OF MEAN AND STANDARD DEVIATION VECTORS) VIA THE DIRECT NEURAL EMBEDDING LEARNING FRAMEWORK, CDS IS USED AS THE SCORING BACK-END

Embeddings	EER (%)	minDCF08	minDCF10	minCprimary	EER (%)	minDCF08	minDCF10	minCprimary
Temporal Pooling	Mean				Mean + STD			
i-vector/PLDA	4.96	0.3336	0.9706	0.7948	-	-	-	-
Softmax	4.65	0.3043	0.9989	0.8026	4	0.2281	0.9041	0.5788
Focal Softmax	4.37	0.2774	0.9969	0.6957	4.07	0.2177	0.8502	0.5433
Triplet	4.33	0.2855	0.9325	0.6921	4.25	0,2635	0.9158	0.6759
Center	4.22	0.291	0.9387	0.7431	**3.44**	0.2063	0.8448	0.5401
Focal Center	4.11	0.2841	0.9723	0.7255	**3.44**	0.2133	0.9106	0.5877
A-softmax	3.75	0.2329	0.8874	0.5963	3.51	**0.1801**	**0.6334**	**0.4169**
Focal-A-softmax	3.74	0.2566	0.8975	0.64	3.49	0.2037	0.8109	0.5

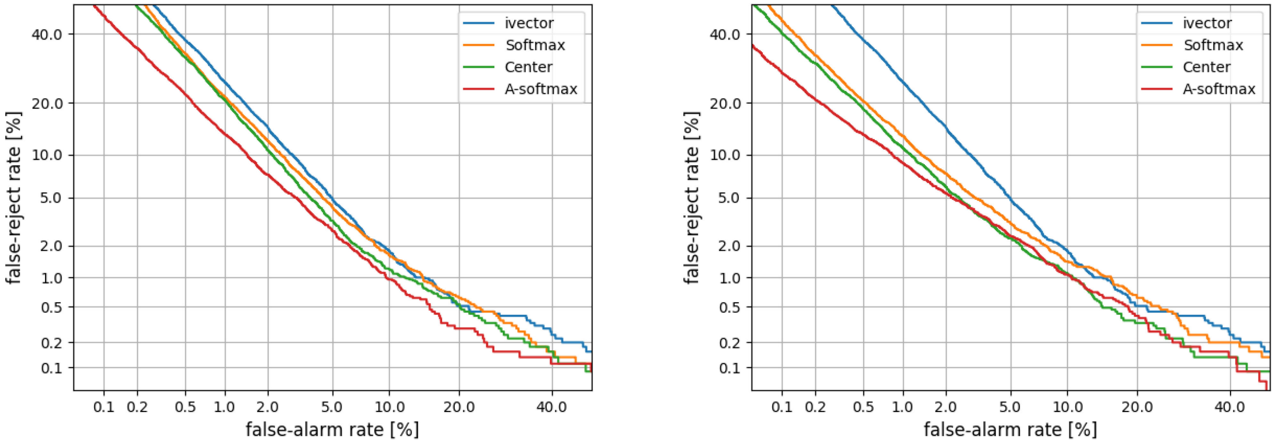


Fig. 6. DET plot of different speaker embeddings. Left: Neural embeddings are obtained via mean pooing. Right: Neural embeddings are obtained via mean+std pooing. (**Best viewed in color**).

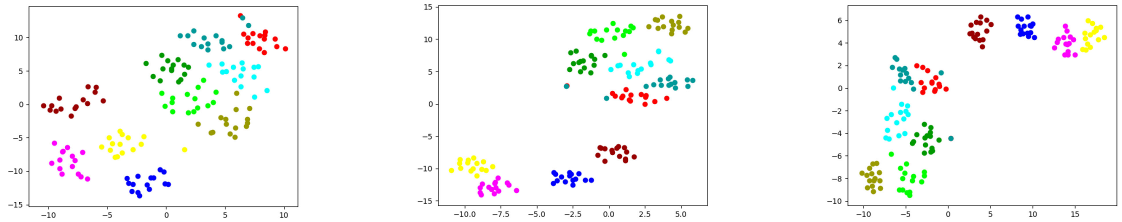
according to the accuracy on the validation set. A-softmax system comprises of several stages of training with an increasing m in Equation 27, details will be given in Section IV. 36-dimension filter bank (Fbank) features are extracted as front-end features for all the three systems and we extend 8 frames on each side to form the 17×36 time-frequency feature maps for each frame.

Four kinds of utterance-level learned neural embeddings are compared in Table III, including softmax embeddings, triplet embeddings, center embeddings and A-softmax embeddings, corresponding to the related optimization metric. Furthermore, mean pooling and mean+std pooling are compared for different embeddings. Focal loss is introduced to softmax, center and A-softmax loss, the preliminary results can also be found in Table III. Compared to the *i*-vector / PLDA baseline, neural speaker embeddings achieve significant performance gains,[3] which demonstrates the ability of NNs on capturing speaker identity knowledge. Detailed analysis of the results will be given from different aspects in the following.

1) Impact of Temporal Pooling Methods: As mentioned in previous sections, speaker verification is an utterance-level decision-making task. A temporal pooling layer is adopted to aggregate frame-level features into utterance-level representations. Most of the researchers use mean pooling [26], [27], [24] also takes the variance statistics into consideration, which can capture more context information. We compared the two pooling methods and results can be found in Table III. It can be observed that with the variance statistics appended, large performance gains in terms of EER are obtained, e.g. 4.65% to 4.0% for softmax embeddings, 4.22% to 3.44% for center embeddings. The consistency shows the effective information encoding ability of the variance statistics, which is also exhibited in our previous work [56]. Moreover, Mean+STD pooled A-softmax based embeddings achieve around 50% reduction in terms of minDCF08, minDCF10 and minCprimary compared to the *i*-vector baseline, and obtains the best result. The individual DET plots are shown in Figure 6, for Mean and Mean+STD pooling respectively.

2) Impact of Focal Loss: As described in Section IV-B, instead of regarding all samples equally, by assigning different weights to different samples, focal loss pays more attention to optimizing hard samples which are easily misclassified. The results are shown in Table III. As the table depicts, the focal loss can further improve the softmax system's performance. However, the focal loss doesn't work for embeddings which take STD statistics into modeling, which reflects that STD cannot only introduce more information but also helps the training process. However, we didn't observe consistent performance gains for the center and A-softmax embeddings. There are two possible reasons: 1) Supervision signals provided by the center loss and A-softmax loss is strong enough to help the network converge to a good optimum. 2) The focal version of these two losses needs more a careful design.

[3] It should be noted that compared to the results in our previous work in [27], the triplet embedding baseline has improved a lot.



(a) *i*-vector (b) embedding: center loss (c) embedding: A-softmax loss

Fig. 7. Visualization of different speaker embeddings: (a) *i*-vector. (b) neural embeddings optimized by center loss. (c) neural embeddings optimized by A-softmax loss. (**Best viewed in color**).

TABLE IV
PERFORMANCE COMPARISON OF A-SOFTMAX EMBEDDING IN DIFFERENT TRAINING STAGES

Value of m	EER (%)	minDCF08	minDCF10	minCprimary	EER (%)	minDCF08	minDCF10	minCprimary
Temporal Pooling	Mean				Mean + STD			
1	4.51	0.2649	0.9352	0.6498	4.16	0.2358	0.8821	0.5732
2	4.1	0.2513	0.9043	0.6223	3.62	0.1903	0.7272	0.4581
3	3.75	0.2329	0.8874	0.5963	**3.51**	0.1801	0.6334	0.4169
4	3.82	0.2327	0.9005	0.5927	3.58	**0.1788**	**0.6289**	**0.4094**

3) Visualization of Speaker Embeddings: To illustrate the effectiveness of the discriminative neural embedding learning, we also use t-SNE [55] to project different embeddings extracted from the same utterances (10 speakers) to 2D plots. As shown in Figure 7, embeddings optimized by the center loss and A-softmax loss are more likely to separate different speakers. Compared to the plot of *i-vector*, the proposed neural embeddings hold larger inter-speaker difference and smaller intra-speaker variance.

4) Hyper-Parameter Tuning. λ **in the center loss***:* as mentioned in the above sections, a weight λ is used to balance the softmax loss and center loss. A small λ implies strong supervision provided by the softmax loss, whereas a large λ implies strong supervision from the center loss. When the weight is too large, the network is actually not trainable. Though the center loss degrades quickly, the softmax loss hardly changes. In this case, the embeddings are trained to be similar to each other and became indistinguishable. As the value of λ is reduced, the softmax loss degrades faster due to its relatively stronger supervision. We recommend a λ between 0.05 and 0.1 (Other settings will cause performance degradation or even model collapse).

m **in the A-softmax loss***:* for the A-softmax based neural embedding learning, the neural network training can be split into several stages with an increasing m. The performance comparison of the learned embeddings in different stages is shown in Table IV, it should be noted that each model with a larger m is initialized with the trained model in the previous stage. Intuitively, the hyper-parameter m controls the scale of the angular margin. A larger m gives a more stringent constraint on the distribution of the deep embeddings and enforces a larger angular margin between classes. However, in practice, a larger m also leads to a slower convergence. In our experience, if m is directly set to 3 and the model is trained from scratch, the neural network converges slower and to a worse optimum. The performance comparison of different stages can be found in Table IV, and there is no performance improvement with a m larger than 4 in our experiments.

TABLE V
EER (%) OF *i*-VECTOR AND NEURAL EMBEDDING (MEAN+STD) COMBINATIONS. DIFFERENT EMBEDDINGS ARE CONCATENATED AND PROJECTED BY DDA, CDS IS USED AS THE SCORING BACK-END

Input Embeddings	DDA (center)	DDA (A-softmax)
i-vector-PLDA	4.96	
i-vector + softmax embedding	3.58	3.52
i-vector + center embedding	**3.11**	**3.25**
i-vector + A-softmax embedding	3.20	3.34

E. Joint Cascade Embedding Learning

Our previous work has shown the complementary properties within the *i*-vector and neural embeddings [27], so in this section, we tried to combine the *i*-vector with neural embeddings to get a further improved system. To be more specific, *i*-vector is concatenated with its corresponding neural embedding and forms the new input of DDA.

As shown in Table V, a further improvement can be observed in all setups, and the best EER of 3.11% is obtained when combining *i*-vector and center embedding with center-loss based DDA.

F. Validation on Different Duration Conditions

Although this work focuses on dealing with short-duration speaker verification, we believe the proposed methods should also work well with longer duration speaker verification. To validate the effects of the proposed frameworks for evaluations with different duration, two additional experiments are carried out.

In the first experiment, we keep the trials the same with the one used in Table III, while varying the number of utterances for the enrollment. As shown in Table VI, with more enrollment data, the performance will be enhanced for all the systems, such

TABLE VI
EER (%) COMPARISON USING DIFFERENT NUMBERS OF ENROLLMENT UTTERANCES

	# enroll utts	1	3	5	10
System	i-vector	8.53	5.47	4.96	4.44
Mean	Softmax	9.29	5.36	4.65	4.02
	Center	8.40	4.87	4.22	3.69
	A-softmax	7.69	4.33	3.75	3.33
Mean+std	Softmax	8.51	4.60	4.00	3.49
	Center	**7.64**	4.11	**3.44**	3.16
	A-softmax	7.82	**4.07**	**3.44**	**3.09**

TABLE VII
EER (%) COMPARISON OF TEST UTTERANCES WITH VARIOUS DURATION

	Duration (s)	1	3	5	10	15
System	i-vector	12.73	7.85	4.82	4.21	3.98
Mean	Softmax	10.73	6.40	4.37	3.67	3.524
	Center	9.04	5.71	3.89	3.24	3.00
	A-softmax	7.67	4.16	3.19	2.81	2.38
Mean+std	Softmax	9.04	4.64	3.63	2.81	2.52
	Center	**7.89**	**4.00**	**3.00**	**2.19**	2.14
	A-softmax	8.53	4.13	**3.00**	2.33	**2.00**

an enhancement is more obvious for the neural network based embeddings. Results with five enrollment utterances correspond to the results in Table III. With more utterances enrolled, the performance gain of proposed systems can be further improved. Another observation is that the mean+std pooling always works better than the mean-only pooling, which is consistent with the previous experiments.

In the second experiment, we keep the enrollment condition the same with the one used in Table III, while new trials are generated with different duration. Utterances with lengths ranging from 1s to 15s. Results can be found in Table VII. As shown in the table, with longer test utterances, the system performance can be further boosted, which is consistent with our expectation.

From the above two validation experiments, it could be inferred that the effectiveness of the proposed methods are not limited to the short-duration condition, and they can be applied to more scenarios.

VI. CONCLUSION

Speaker embeddings exhibit impressive performance on the short-duration speaker verification, which is a common task in real-world scenarios. In this paper, we categorized the embedding learning into two frameworks, i.e., cascade embedding learning and direct embedding learning. The former consists of multiple stages on the learning process, while the latter directly learns discriminative embeddings from the spectral features for scoring. Center loss and A-softmax loss are introduced into the neural speaker embedding learning in these two paradigms. Moreover, Focal loss is proposed to integrate with other losses. Main experiments are carried out on a short-duration text-independent speaker verification task, and four metrics including EER, minDCF08, minDCF10 and minCprimary are used to evaluate the performance. The proposed direct neural embedding learning methods show better results compared to the *i*-vector/PLDA system, achieving ∼30% reduction on EER. The proposed cascade embedding learning with DDA also outperforms traditional compensation methods such as LDA and PLDA. Finally combining neural embedding with traditional i-vector can further improve the performance. The best system can obtain 3.11% in terms of EER, which is significantly better than the *i*-vector/PLDA baseline with 4.96%. Furthermore, we discussed the application of the proposed methods to various duration conditions, a consistent performance improvement can be observed from the additional experiments.

REFERENCES

[1] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digit. Signal Process.*, vol. 10, no. 1–3, pp. 19–41, 2000.

[2] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308–311, May 2006.

[3] 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. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 1, 2006, pp. I-97–I-100.

[4] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," CRIM, Montreal, QC, Canada, Rep. CRIM-06/08-13, 2005.

[5] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[6] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[7] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *IEEE Workshop Autom. Speech Recognit. Understanding*, 2011, pp. 24–29.

[8] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. 12th Annu. Conf. Int. Speech Commun. Assoc.*, 2011, pp. 1–2 .

[9] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[10] Y. Xu, J. Du, L. R. Dai, and C. H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 65–68, Jan. 2014.

[11] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Proc. Interspeech*, 2013, pp. 436–440.

[12] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2014, pp. 1695–1699.

[13] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition," in *Proc. Odyssey*, 2014, pp. 293–298.

[14] 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. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2014, pp. 4052–4056.

[15] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Commun.*, vol. 73, pp. 1–13, 2015.

[16] T. Fu, Y. Qian, Y. Liu, and K. Yu, "Tandem deep features for text-dependent speaker verification," in *Proc. Interspeech*, 2014, pp. 1327–1331.

[17] Y. Tian, M. Cai, L. He, and J. Liu, "Investigation of bottleneck features and multilingual deep neural networks for speaker verification," in *Proc. Interspeech*, 2015, pp. 1151–1155.

[18] F. Richardson, D. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," 2015, *arXiv:1504.00923*.

[19] L. Li, Y. Lin, Z. Zhang, and D. Wang, "Improved deep speaker feature learning for text-dependent speaker recognition," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2015, pp. 426–429.

[20] N. Chen, Y. Qian, and K. Yu, "Multi-task learning for text-dependent speaker verification," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 185–189.

[21] S. Wang, Y. Qian, and K. Yu, "What does the speaker embedding encode?" in *Proc. Interspeech*, vol. 2017, 2017, pp. 1497–1501.

[22] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “Sphereface: Deep hypersphere embedding for face recognition,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, vol. 1, pp. 6738–6746, 2017.

[23] G. Bhattacharya, J. Alam, T. Stafylakis, and P. Kenny, “Deep neural network based text-dependent speaker recognition: Preliminary results,” *Proc. Odyssey*, 2016, pp. 2–15.

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

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

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

[27] Z. Huang, S. Wang, and Y. Qian, “Joint i-vector with end-to-end system for short duration text-independent speaker verification,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4869–4873.

[28] T. Pekhovsky, S. Novoselov, A. Sholohov, and O. Kudashev, “On autoencoders in the i-vector space for speaker recognition,” in *Proc. Odyssey*, 2016, pp. 217–224.

[29] J. Guo, U. A. Nookala, and A. Alwan, “CNN-based joint mapping of short and long utterance i-vectors for speaker verification using short utterances,” in *Proc. Interspeech*, 2017, pp. 3712–3716.

[30] S. Novoselov, T. Pekhovsky, O. Kudashev, V. S. Mendelev, and A. Prudnikov, “Non-linear PLDA for i-vector speaker verification,” in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 214–218.

[31] O. Ghahabi and J. Hernando, “Deep belief networks for i-vector based speaker recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2014, pp. 1700–1704.

[32] Y.-h. Chen, I. Lopez-Moreno, T. N. Sainath, M. Visontai, R. Alvarez, and C. Parada, “Locally-connected and convolutional neural networks for small footprint speaker recognition,” in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 1136–1140.

[33] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang, “Deep speaker feature learning for text-independent speaker verification,” 2017, *arXiv:1705.03670*.

[34] F. Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, “Attention-based models for text-dependent speaker verification,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 5359–5363.

[35] W. Cai, Z. Cai, X. Zhang, X. Wang, and M. Li, “A novel learnable dictionary encoding layer for end-to-end language identification,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 5189–5193.

[36] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, “End-to-end attention based text-dependent speaker verification,” in *Proc. Spoken Lang. Technol. Workshop*, 2016, pp. 171–178.

[37] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” 2017, *arXiv:1710.10467*.

[38] W. Cai, J. Chen, and M. Li, “Exploring the encoding layer and loss function in end-to-end speaker and language recognition system,” in *Proc. Odyssey Speaker Lang. Recognit. Workshop*, 2018, pp. 74–81. [Online]. Available: http://dx.doi.org/10.21437/Odyssey.2018-11

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

[40] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[41] S. O. Sadjadi, S. Ganapathy, and J. W. Pelecanos, “The IBM 2016 speaker recognition system,” 2016, *arXiv:1602.07291*.

[42] N. Brümmer and E. De Villiers, “The speaker partitioning problem,” in *Proc. Odyssey*, 2010, p. 34.

[43] P. Kenny, “Bayesian speaker verification with heavy-tailed priors,” in *Proc. Odyssey*, 2010, p. 14.

[44] S. J. Prince and J. H. Elder, “Probabilistic linear discriminant analysis for inferences about identity,” in *Proc. IEEE 11th Int. Conf. Comput. Vision*, 2007, pp. 1–8.

[45] D. Garcia-Romero and C. Y. Espy-Wilson, “Analysis of i-vector length normalization in speaker recognition systems,” in *Proc. 12th Annu. Conf. Int. Speech Commun. Assoc.*, 2011, pp. 249–252.

[46] D. Povey *et al.*, “The Kaldi speech recognition toolkit,” in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2011, p. 33.

[47] S. Ioffe, “Probabilistic linear discriminant analysis,” in *Proc. Eur. Conf. Comput. Vision*, 2006, pp. 531–542.

[48] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 499–515.

[49] L. Li, Z. Tang, and D. Wang, “Full-info training for deep speaker feature learning,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, 5369–5373.

[50] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.

[51] S. Ranjan, J. H. Hansen, S. Ranjan, and J. H. Hansen, “Curriculum learning based approaches for noise robust speaker recognition,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 1, pp. 197–210, Jan. 2018.

[52] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” 2017, *arXiv:1708.02002*.

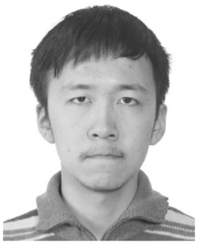
[53] S. Wang, Y. Qian, and K. Yu, “Focal kl-divergence based dilated convolutional neural networks for co-channel speaker identification,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 5339–5343.

[54] S. O. Sadjadi *et al.*, “The 2016 NIST speaker recognition evaluation,” in *Proc. Interspeech*, 2017, pp. 1353–1357.

[55] L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, no. Nov., pp. 2579–2605, 2008.

[56] S. Wang, H. Dinkel, Y. Qian, and K. Yu, “Covariance based deep feature for text-dependent speaker verification,” in *Proc. Int. Conf. Intell. Sci. Big Data Eng.*, 2018, pp. 231–242.

Shuai Wang (S’18) received the B.S degree from Northwestern Polytechnical University, Xi’ an, China, in 2014. He is currently working toward the Ph.D. degree at the SpeechLab of Shanghai Jiao Tong University, Shanghai, China, under the supervision of Kai Yu and Yanmin Qian. His current research mainly focuses on speaker recognition and speaker diarization.



Zili Huang (S’18) received the B.S degree from Shanghai Jiao Tong University, China in 2018. He is currently working toward the Ph.D. degree at Johns Hopkins University, Baltimore, MD, USA, under the supervision of Daniel Povey. His current research mainly focuses on speaker recognition and speaker diarization.

Yanmin Qian (S’09–M’13–SM’19) received the B.S degree from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2012. He joined the Department of Computer Science and Engineering at Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2013, and is currently an Associate Professor there. From 2015 to 2016, he also worked as an Associate Research with the Speech Group at Cambridge University Engineering Department, Cambridge, U.K. His current research interests include the acoustic and language modeling in speech recognition, speaker and language recognition, key word spotting, and multimedia signal processing.

Kai Yu (M’06–SM’11) received the B.Eng. degree in automation and the M.Sc. degree from Tsinghua University, China, in 1999 and 2002, respectively. He then joined the Machine Intelligence Lab with the Engineering Department at Cambridge University, Cambridge, U.K., where he received the Ph.D. degree in 2006. He is currently a Research Professor with Shanghai Jiao Tong University, Shanghai, China. His main research interests include the area of speech-based human-machine interaction, including speech recognition, synthesis, language understanding, and dialogue management. Dr. Yu was selected into the 1000 Overseas Talent Plan (Young Talent) by the Chinese government and the Excellent Young Scientists Project of NSFC China. He is a member of the Technical Committee of the Speech, Language, Music and Auditory Perception Branch of the Acoustic Society of China.