INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Deep Speaker Recognition: Modular or Monolithic?

Gautam Bhattacharya[1,2], *Jahangir Alam* [2], *Patrick Kenny*[2]

[1]McGill University
[2]Computer Research Institute of Montreal
gautam.bhattacharya@mail.mcgill.ca, jahangir.alam@crim.ca

Abstract

Speaker recognition has made extraordinary progress with the advent of deep neural networks. In this work, we analyze the performance of end-to-end deep speaker recognizers on two popular text-independent tasks - NIST-SRE 2016 and VoxCeleb. Through a combination of a deep convolutional feature extractor, self-attentive pooling and large-margin loss functions, we achieve state-of-the-art performance on VoxCeleb. Our best individual and ensemble models show a relative improvement of 70% an 82% respectively over the best reported results on this task.

On the challenging NIST-SRE 2016 task, our proposed end-to-end models show good performance but are unable to match a strong i-vector baseline. State-of-the-art systems for this task use a modular framework that combines neural network embeddings with a probabilistic linear discriminant analysis (PLDA) classifier. Drawing inspiration from this approach we propose to replace the PLDA classifier with a neural network. Our modular neural network approach is able to outperform the i-vector baseline using cosine distance to score verification trials.

Index Terms: deep speaker recognition, end-to-end, large margin loss

1. Introduction

In recent years, deep speaker recognition has advanced significantly through the release of publicly-available datasets, continual advancement in neural network architectures, loss function design and deep metric learning [1, 2, 3, 4]. In this work we focus on building neural speaker embedding (NSE) models for two popular speaker verification tasks, namely, NIST-SRE 2016 and VoxCeleb [5, 6]. These tasks have shaped much of the recent developments in learning NSE models. Models for the NIST task have focused on a modular framework that combines speaker embeddings with an external classifier [7, 8], while monolithic or end-to-end (E2E) models represent the state-of-the-art on VoxCeleb. In this work we take the latter approach for learning NSE models and identify the challenges involved in learning such models on both tasks.

The NIST Speaker Recognition Evaluations (SRE) have been a long standing benchmark in the speaker recognition community. The x-vector model has firmly established itself as the state-of-the-art in recent NIST evaluations [9]. The approach follows the same modular recipe for speaker verification as i-vectors, i.e., speaker embeddings are used to train a PLDA classifier. Learning E2E models for the NIST tasks is particularly challenging. In our recent work we showed that by augmenting our NSE models with an adversarial domain adaptation strategy we are able to learn end-to-end models that outperform i-vectors, and are competitive with the x-vector model [10]. That being said, we find that without adaptation, our proposed end-to-end approach lags behind i-vector systems. To tackle this problem, we propose a novel modular approach that draws inspiration from the x-vector/PLDA recipe. Unlike the x-vector model, our approach uses a second neural network instead of PLDA. We show that the speaker embeddings extracted from this model deliver slightly better performance than a strong i-vector baseline. The main advantages of our method is that it is easier to train than the end-to-end model, while also delivering robust performance using cosine distance.

Unlike the NIST-SRE tasks, E2E models have established themselves as the state-of-the-art on the VoxCeleb task. We hypothesize that one of the main reasons for this is because unlike NIST, the VoxCeleb training data contains a balanced distribution of classes (speakers). In this work we propose a novel E2E-NSE model that furthers the state-of-the-art on this task. Our proposed approach shows a significant 70% over the previous best reported result on VoxCeleb [11]. Moreover, our models are trained on VoxCeleb-1, while the method we outperform was trained on the much larger VoxCeleb-2 [12]. We believe that the impressive performance of our models is a combination of several factors. Our models makes use of a deep Residual Network (ResNet) feature extractor [13], self-attention [14], and large margin loss functions. From our analysis we identify*feature normalization* as the single most important factor behind the success of our model.

One of the main insights from our experiments on these two task is that learning E2E-NSE models, is the importance of training data. While both tasks have training datasets of comparable size in terms of recordings, the 140,000 recordings in VoxCeleb are distributed over 1251 speakers, while the 208,000 (approx) NIST recordings are covered by 6000 speakers. We believe that this is the main reason for the challenges we encountered in learning E2E-NSE models on the latter, while on the former our proposed models are able to significantly improve on the state-of-the-art.

2. ResNet Speaker Embeddings

In this work we make use of our recently proposed architecture for learning speaker embeddings based on a deep ResNet model [15]. ResNet models have been widely adopted for learning NSE models, especially for the VoxCeleb task [12, 16, 17, 11]. Unlike these methods our ResNet model uses 1-dimensional convolutional kernels in its residual blocks. We make use of an attentive statistics pooling layer which allows us to train our models on speech inputs of variable size. This also avoids the need for an aggregation strategy for extracting utterance-level embeddings (as we can simply process the entire recordings through the network).For learning NSE models on the NIST data we make use of the same model as our previous work. Specifically, we use a residual block configuration of [3,4,6,3], which including the input convolutional layer, attentive pooling

http://dx.doi.org/10.21437/Interspeech.2019-3146

and fully connected layers makes for a total of 52 layers. We refer the reader to [15] for a detailed description of the model.

We modify the model architecture for the VoxCeleb task due to the smaller size of the dataset. We use a residual block configuration of [2,2,2,2], while the rest of the the model remains the same.

3. Modular Neural Speaker Recognizers

In our experiments we show that learning E2E-NSE models is particularly challenging on the NIST-SRE tasks. We hypothesize that one potential reason for this is the mismatch between conditions seen during training and test time. In the case of NIST-SRE 2016, the enrolment recordings are around 60 seconds long, while test recordings vary between 10 to 60 seconds. Due to computational restrictions, NSE models (including x-vectors) are typically trained on short chunks of speech ranging between 2 to 10 seconds. Thus at test time, the model is typically seeing recordings that are significantly longer than those it saw during training. Our hypothesis is that PLDA modelling plays a crucial role in compensating for this unavoidable mismatch.

In our previous work we showed that by learning small neural network classifiers on top of i-vectors, we can obtain speaker embedding models that perform robustly using cosine distance [18]. In this work we propose to extend this method, and develop an all-neural approach. Our modular framework is divided into two steps:

- **Step 1:** Train a NSE model by minimizing the Softmax loss. We use the deep ResNet architecture presented in the previous section. After the model has converged, extract speaker embeddings from the entire training dataset.
- **Step 2:** Train a small classifier using the embeddings extracted in step 1 as input. . This model can be trained using any of the loss functions detailed in section 4. We extract embeddings from this second model to perform speaker verification.

The modular approach proposed here draws inspiration from PLDA in that the second stage neural network makes classifications based on full-duration embeddings, thus avoid any duration related mismatch between training and test conditions. For step 1, we use the ResNet architecture presented in the previous section and learn a speaker model by minimizing the Softmax loss. Embeddings from this model are 512-dimensional. For step-2, we train a feedforward classifier on top of these embeddings using the model described in [18].

We note that while approach presented here is not strictly E2E, we are able to evaluate the resulting speaker embeddings in a similar fashion, i.e., using cosine distance.

4. Optimizing for Embedding Similarity

The most popular approach for learning NSE models is optimizing the Softmax loss [9]. The softmax loss is tailored for closed-set identification tasks, but typically does not learn deep representations or embeddings that are discriminative enough for open-set verification. In speaker recognition, such models are often combined with a PLDA classifier [18] or an advanced embedding aggregation strategy [11]. The Softmax loss does not optimize for embedding similarity, i.e., the quantity we are interested in at test time. An end-to-end model recognition model typically need to be optimized in this way as we wish to compute scores for verification trials using simple distance metrics like mean squared error or cosine similarity.

In recent years, several enhanced versions of the Softmax loss have been proposed to make it robust for verification tasks [19, 20, 21]. In combination with deep CNNs, these loss functions have achieved state-of-the-art performance. In this work we focus on the Additive Margin Softmax [22, 23] and the recently introduced Additive Angular Margin losses [24]. The AM-Softmax loss has been successful translated to speaker recognition for both the VoxCeleb and NIST-SRE 2016 tasks [17, 11, 15, 10]. The basic Softmax loss is given in eq. 1. By setting the bias terms to zero, The loss can be seen as optimizing the inner product of the weight vector, $W_.$ and embedding $x_.$. Setting the biases to zero is important as the model can make classification decisions based on the bias term alone [20].

$$\begin{aligned}\mathcal{L}_S &= -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{W_{y_i}^T \boldsymbol{f}_i}}{\sum_{j=1}^{c} e^{W_j^T \boldsymbol{f}_i}} \\ &= -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{\|W_{y_i}\|\|\boldsymbol{f}_i\| cos(\theta_{y_i})}}{\sum_{j=1}^{c} e^{\|W_j\|\|\boldsymbol{f}_i\| cos(\theta_j)}},\end{aligned} \tag{1}$$

From the second line of eq. 1 we see that if the weight vectors and embeddings are constrained to have unit norm, we can optimize the cosine similarity directly. The key insight of the recent margin-based softmax loss functions is that an angular and/or additive margin term can be incorporated into this formulation which enforces intra-class compactness and inter-class diversity [24]. The additive margin loss utilizes a margin m in cosine space:

$$\begin{aligned}\mathcal{L}_{AMS} &= -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{s\cdot\left(cos\theta_{y_i}-m\right)}}{e^{s\cdot\left(cos\theta_{y_i}-m\right)}+\sum_{j=1,j\neq y_i}^{c} e^{s\cdot cos\theta_j}} \\ &= -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{s\cdot\left(W_{y_i}^T \boldsymbol{f}_i-m\right)}}{e^{s\cdot\left(W_{y_i}^T \boldsymbol{f}_i-m\right)}+\sum_{j=1,j\neq y_i}^{c} e^{sW_j^T \boldsymbol{f}_i}}.\end{aligned} \tag{2}$$

The additive margin loss also uses a scale parameter s. The scale parameter is used to rescale the embeddings to be of norm s. and is a crucial part of the loss as simply normalizing the weights and embeddings to have unit norms leads to issues with model convergence [20]. Intuitively, the margin encourages the similarity of correct classes to be greater than that of incorrect classes by a margin m.

Instead of penalizing the cosine of the angle, we can also penalize the angle directly. The angular margin softmax combine a margin with the angle in a multiplicative way. More recently, the additive angular margin loss proposed to do this in a an additive way:

$$L_{AAM} = -\frac{1}{N}\sum_{i=1}^{N} \log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}. \tag{3}$$

The AAM loss also uses weight and feature normalization followed by rescaling. For all models in this work we set $m = 0.6$ and $s = 40$.

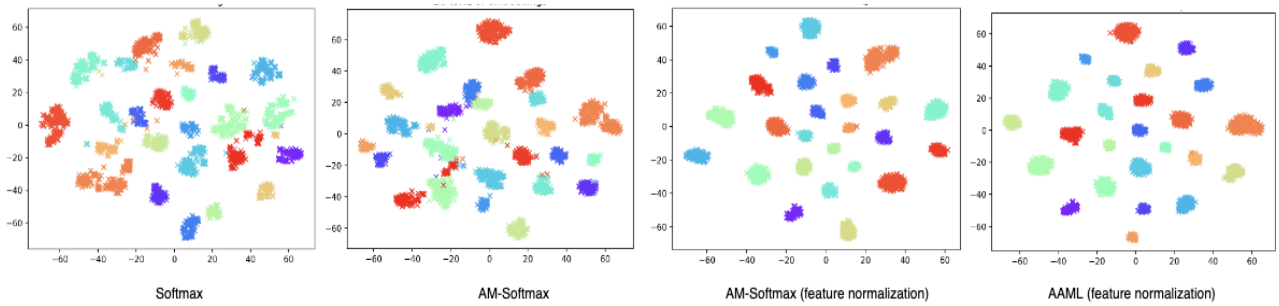

Figure 1: *t-SNE Visualization of the Embedding Space of different NSE models.*

4.1. Feature Normalization

Rescaling of the feature norms is essential for training the additive and angular margin losses. The authors of [23] argue that feature normalization is crucial, noting that adaptively learning the L2-norm of the embeddings can lead to a relatively weak cosine constraint. While embedding normalization has generally found to be beneficial, models like the angular margin softmax do not use it [21]. The authors of [22] note that embedding normalization is most helpful when the quality of images is low, thus suggesting that its requirement is data dependent.

Our experience with feature normalization agrees with the data-dependent hypothesis. On the VoxCeleb task we find that it leads to large improvement over un-normalized features. On the other hand, we did not find that feature normalization improved performance on the NIST-SRE 2016 task. Nonetheless, given the improvement in performance yielded by feature normalization on VoxCeleb, we believe that exploring different feature normalization strategies presents an interesting direction for future research.

Figure 1. visualizes speaker embeddings extracted from different NSE models using t-sne [25]. We see that embeddings from the models that use feature normalization produce tighter speaker clusters with a minimal amount of class overlap. In contrast, we see that the Softmax and un-normalized AM-Softmax models exhibit significantly more overlap between classes.

5. Experiments

5.1. Datasets

VoxCeleb: For this task we use the VoxCeleb-1 training data for learning NSE models. The dataset consists of more than 140K utterances from 1,251 speakers. There are 37,720 trial pairs from 40 speakers are used as evaluation data for the verification process. The average duration of training and evaluation data is 8.24s and 8.28s, respectively. We do not use any data augmentation for this task.

NIST-SRE 2016: We use data from past NIST evaluations (2004-2010) and a part of the Switchboard data for training speaker models. We augment the dataset with noise and reverberation as in [9]. We note that the training data is primarily in English, while the NIST-SRE 2016 evaluation data contains Cantonese and Tagalog speakers.

5.2. Setup

One of the objectives of this work is to analyse the differences between the two tasks (VoxCeleb and NIST-SRE 2016) in terms of learning NSE models. Consequently we choose to learn such models using the same speech features for both tasks. We use 23-dimensional mel frequency cepstral coefficients (MFCC) features as input to our models. MFCC features are extracted using a 25ms with with a hop size of 10ms. Additionally, we down-sample the VoxCeleb audio to 8 Khz to match the NIST data. We also use a voice activity detector (VAD) to remove silence frames. One advantage of using a compact speech representation like MFCCs is that it allows us to train of relatively large chucks of audio (upto 8 seconds). This allows us to better simulate conditions seen at test time during model training.
We report verification performance on all tasks in terms of equal error rates (EER),

5.3. Network Training

In our experiments we found that pre-training our NSE models with the Softmax loss was essential for training margin based losses on the NIST data, and also lead to faster model convergence on VoxCeleb. For margin losses we add an additional 64-dimensional fully connected layer on top of pre-trained model. We also use to extract speaker embeddings for training the second stage network of our proposed modular framework. All VoxCeleb models are trained using the RMSprop optimizer with a learning rate of 0.0003 for 50 epochs, after which the learning rate is annealed until convergence. For the NIST models we use a small validation set to determine a threshold for early stopping.

All NSE models are trained on 3-8 second long segments of speech. We do not use any data pre-processing and sample random chucks of audio from each recording during model training. For VoxCeleb we found it sufficient to sample each recording once per training epoch, while for our NIST models we sample each recording 10 times in each training epoch. We train the i-vector and x-vector baselines for both tasks using the open-source Kaldi implementation.

5.4. Verification on VoxCeleb

Table 1 compares the performance of our proposed NSE models to the current state-of-the-art on the VoxCeleb1 test set. Our method, specifically, our models that employ feature normalization, outperform all other methods by a large margin. We see that our AM-Softmax model without feature normalization shows comparable performance to the model in [17] (4.63% vs

Table 1: *Results for verification on VoxCeleb. All x-vector systems make use of data augmentation.*

	Front-end model	Loss	Dims	Training set	EER (%)
VoxCeleb1 test set					
Shon *et al*. [26]	i-vectors + PLDA	–	600	VoxCeleb1	5.41
Shon *et al*. [26]	x-vector*	Softmax	512	VoxCeleb1	6.10
Cai *et al*. [16]	ResNet-34	A-Softmax + PLDA	128	VoxCeleb1	4.46
Okabe *et al*. [14]	x-vector	Softmax	1500	VoxCeleb1	3.85
Hajibabaei *et al*. [17]	ResNet-20	AM-Softmax	128	VoxCeleb1	4.30
Chung *et al*. [12]	ResNet-50	Softmax + Contrastive	512	VoxCeleb2	4.19
Xie *et al*. [11]	ResNet-34	Softmax	512	VoxCeleb2	3.22
Ours	ResNet-28	Softmax	512	VoxCeleb1	8.19
Ours	ResNet-28	AM-Softmax	64	VoxCeleb1	4.63
Ours (normalized features)	ResNet-28	AM-Softmax	64	VoxCeleb1	1.03
Ours (normalized features)	ResNet-28	AAM	64	VoxCeleb1	**0.95**
Ours (Ensemble)	ResNet-28	AAM/AM-Softmax	64	VoxCeleb1	**0.55**

4.30%). We see that our AM-Softmax model with normalized features outperforms both by a large margin, achieving an EER of 1.03%. We believe that this result highlights the importance of feature normalization for learning robust E2E-NSE models.

The current state-of-the-art NSE model for this task uses a ResNet model on top of spectrograms, followed by a powerful aggregation strategy. Our AAM model with feature normalization significantly improves over this approach, achieving an EER of 0.95% versus 3.22%. This corresponds to a 70% relative improvement. Notably, the standard x-vector/PLDA system does not perform well on this task, achieving an EER of 6.1%.

Model Ensembling: We are able to further improve the performance of our systems by creating an ensemble of our E2E-NSE models. We combine models by averaging the scores of individual systems. We note that an ensemble of 2 AMML models improves the EER from 0.95% to 0.63%, and adding 2 AM-softmax models to the ensemble leads to an EER of 0.55%. This corresponds to a 42% relative improvement over our best individual system.

5.5. Verification on NIST-SRE 2016

Table 2 compares the performance of our proposed E2E and modular networks to the baseline i-vector and x-vector systems. We see that the AM-Softmax model performs fairly well for an E2E system, but lags behind both i-vector and x-vector systems. The AM-Softmax model outperforms the basic Softmax model significantly, achieving an EER or 15.88% versus 18.52% on the pooled task. We note that we do not use feature normalization in conjunction with the AM-Softmax loss as were unable to learn such models for this dataset, encountering the convergence issues described in [20].

We extract speaker embeddings from the Softmax model for training our proposed modular framework. The modular approach performs significantly better than the E2E AM-Softmax model. Somewhat surprisingly, we found that using the basic Softmax loss worked slightly better than the AM-Softmax loss. Once again, we were unable to get models with feature normalization to converge. The modular approach is able to beat the i-vector baseline (13.24% vs 13.65%), but still lags behind the x-vector system.

Table 2: *Results for verification on NIST-SRE 2016. Performance is reported in terms of EER.*

Front-end model	Cantonese	Tagalog	Pooled
PLDA Models			
i-vectors	9.51	17.61	13.65
x-vectors	7.52	15.96	11.73
E2E Models			
Softmax	14.25	22.91	18.52
AM-Softmax	11.05	20.93	15.88
Modular-NN	8.52	18.36	13.24
AdversarialNN	**7.26**	**15.77**	**11.59**

We also include results of a NSE model developed in our previous work (AdversarialNN) that uses adversarial domain adaptation [10]. While this approach is able to close the gap to x-vectors, we were not able to combine this approach successfully with our modular framework. We also note that the performance of x-vectors can be significantly improved by adapting the PLDA classifier [27, 9].

6. Conclusion

In this work we furthered the state-of-the-art in E2E deep speaker recognition on the VoxCeleb task. While we identified feature normalization as a key component of our models, we hypothesize that another reason for the success of our models is that the proposed architecture and speech features allow us to accurately simulate test-time conditions during model training.

Our work on NIST-SRE 2016 has shown that E2E-NSE models can be more challenging to learn, and work remains in order to match the performance of the embedding + PLDA approach. We found that our proposed modular neural network approach outperforms an E2E model, however we were unable to combine the former with adversarial training which leads to the best E2E performance overall.

As to the question posed in the title of the paper, we offer a somewhat unsurprising answer - it depends on the data.

7. References

[1] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern Recognition*, vol. 90, pp. 119–133, 2019.

[2] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, "Densenet: Implementing efficient convnet descriptor pyramids," *arXiv preprint arXiv:1404.1869*, 2014.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[5] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. Greenberg, E. S. Reynolds, L. Mason, and J. Hernandez-Cordero, "The 2016 nist speaker recognition evaluation," in *Proc. Interspeech*, 2017, pp. 1353–1357.

[6] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[7] G. Bhattacharya, M. J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification." in *Interspeech*, 2017, pp. 1517–1521.

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

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

[10] G. Bhattacharya, J. Monteiro, J. Alam, and P. Kenny, "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification," 2019.

[11] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," 2019.

[12] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.

[15] G. Bhattacharya, J. Alam, and P. Kenny, "Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training," in *Acoustics Speech and Signal Processing (ICASSP), 2019 IEEE International Conference on*. IEEE, 2019.

[16] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," *arXiv preprint arXiv:1804.05160*, 2018.

[17] M. Hajibabaei and D. Dai, "Unified hypersphere embedding for speaker recognition," *arXiv preprint arXiv:1807.08312*, 2018.

[18] G. Bhattacharya, J. Alam, P. Kenn, and V. Gupta, "Modelling speaker and channel variability using deep neural networks for robust speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 192–198.

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

[20] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: l2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*. ACM, 2017, pp. 1041–1049.

[21] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.

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

[23] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[24] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *arXiv preprint arXiv:1801.07698*, 2018.

[25] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[26] S. Shon, H. Tang, and J. Glass, "Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 1007–1013.

[27] D. Garcia-Romero, A. McCree, S. Shum, N. Brummer, and C. Vaquero, "Unsupervised domain adaptation for i-vector speaker recognition," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2014.