ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10094954

EXPLORING BINARY CLASSIFICATION LOSS FOR SPEAKER VERIFICATION

Bing Han, Zhengyang Chen, Yanmin Qian[†]

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China

ABSTRACT

The mismatch between close-set training and open-set testing usually leads to significant performance degradation for speaker verification task. For existing loss functions, metric learning-based objectives depend strongly on searching effective pairs which might hinder further improvements. And popular multi-classification methods are usually observed with degradation when evaluated on unseen speakers. In this work, we introduce SphereFace2 framework which uses several binary classifiers to train the speaker model in a pair-wise manner instead of performing multi-classification. Benefiting from this learning paradigm, it can efficiently alleviate the gap between training and evaluation. Experiments conducted on Voxceleb show that the SphereFace2 outperforms other existing loss functions, especially on hard trials. Besides, large margin fine-tuning strategy is proven to be compatible with it for further improvements. Finally, SphereFace2 also shows its strong robustness to class-wise noisy labels which has the potential to be applied in the semi-supervised training scenario with inaccurate estimated pseudo labels.

Index Terms— speaker verification, sphereface2, binary classification, large margin fine-tuning

1. INTRODUCTION

Speaker verification (SV) is the task of determining whether a pair of speech segments belong to the same speaker or not. Recently, with the thriving of deep neural networks (DNN), DNN-based speaker verification systems have obtained excellent performance when compared with traditional Gaussian Mixture Model (GMM)-based i-vector [1]. Generally, a typical SV model consists of three parts: (1) a frame-level speaker feature extractor [2, 3], (2) a pooling layer for statistic aggregation [4, 5, 6] and (3) a loss function for optimization.

For loss functions in SV, it can be mainly divided into two technical routes. Firstly, considering the open-set setting of SV task, it's reasonable to use contrastive learning-based metric objectives (eg. angular prototypical [7]) to optimize the pair-wise similarity. On the other hand, the softmax-based multi-class classifier is adopted to distinguish the different speakers in training set [8, 9]. However, in verification task, both of them have some shortcomings. For metric learning-based methods, the performance strongly depends on the strategy to search effective pairs or triplets which is very time- and computation-consuming with the increasing of training samples number. For multi-classification methods, the embeddings produced by the DNN are not generalizable enough and performance degradation is observed when evaluated on unseen speakers due to the lack of similarity optimization explicitly [8]. Recently, several margin-based softmax variants [10, 11, 12, 13, 14, 15] are proposed to boost the discriminative power of speaker representation. They optimize the speaker embedding in a hyper-sphere space and encourage intra-class compactness by adding a margin to tighten the decision boundary. Although these multi-classification methods obtain significant performance gains, it's still difficult to ignore the mismatch between close-set training and open-set evaluation.

To alleviate the close-set assumption, in this paper, we introduce a novel binary classification-based framework SphereFace2 [16] for speaker verification. Unlike multi-classification training widely used before, it performs binary classification on hyper-sphere space which can effectively bridge the gap between training and evaluation, since both training and evaluation adopt pair-wise comparisons. Specifically, suppose there are K speakers in the training set, it will construct K independent binary classification objectives which regard data from the target speaker as positive samples and the others are negative. Experiments are conducted on Voxceleb [17, 18], and the results illustrate that the SphereFace2 achieves better performance compared with the metric loss and multi-classification loss, which demonstrates that its pair-wise training manner can efficiently alleviate the mismatch between close-set training and open-set testing. Moreover, SphereFace2 also verifies its robustness against class-wise label noise, benefiting from the weak supervision of pair-wise labels.

[†] Corresponding Author.

This work was supported in part by China NSFC projects under Grants 62122050 and 62071288, and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102. Experiments have been carried out on the PI super-computer at Shanghai Jiao Tong University.

2. RELATED WORK

2.1. Metric Learning-based Loss

Prototypical loss [19] is a widely used metric learning-based loss function in speaker verification. During the training, each mini-batch consists of a support set S and a query set Q. In our implementation, we sample $N \times M$ utterances in each mini-batch, where N is the speaker number and M is the utterance number for each speaker. Besides, we consider the M-th utterance for each speaker as the query and the others as the support set. Then the prototype c_j for each speaker can be calculated by $c_j = \frac{1}{M-1}\sum_{m=1}^{M-1} x_{j,m}$. Then, i-th query can be classified against N speakers based on softmax to optimize the distance between samples and prototypes:

$$L_P = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{S_{i,i}}}{\sum_{k=1}^{N} e^{S_{i,k}}} \tag{1}$$

where $S_{i,k} = ||x_{i,M} - c_k||_2$, the squared Euclidean distance between the i-th query and k-th prototype. Besides, we can replace the L2-distance function with a cosine-based similarity metric to get angular prototypical loss [7]:

$$S_{i,k} = w \cdot \cos(x_{i,M}, c_k) + b \tag{2}$$

where w and b are learnable scale and bias parameters.

2.2. Margin-based Softmax

For margin-based softmax loss function, the general formula can be summarized as:

$$L_M = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s\cdot\psi(\theta_{y_i})}}{e^{s\cdot\psi(\theta_{y_i})} + \sum_{j\neq y_i} e^{s\cdot\cos(\theta_j)}} \tag{3}$$

where $\psi(\theta_{y_i}) = \cos(m_1\theta_{y_i} + m_2) - m_3$ and s is the scale factor to accelerate and stabilize the training. The m_1, m_2 and m_3 correspond to angular softmax (A-softmax) [10], additive angular softmax (AAM-softmax) [11] and additive margin softmax (AM-softmax) [12] respectively. With these margins, the decision boundary is tightened which can explicitly enhance the similarity of intra-class samples and enlarge the distance between inter-class samples.

3. SPHEREFACE2: BINARY CLASSIFICATION

Given K speakers in the training set, SphereFace2 is designed to alleviate the mismatch between close-set training and open-set evaluation by explicitly constructing K independent binary classification heads to perform the pair-wise comparison. Specifically, for the i-th sample in a batch, $\boldsymbol{x}_i \in \mathbb{R}^d$ represents the corresponding input to the classification layer and y_i is the ground truth. For the projection head, we denote the weights of the j-th binary classifier by $\boldsymbol{W}_j$. Then the loss can be formulated as:

$$\begin{aligned} L_i =& \log(1 + \exp(-\boldsymbol{W}_{y_i}^\top \boldsymbol{x}_i - b_{y_i})) \\ &+ \sum_{j\neq y_i}^{K} \log(1 + \exp(\boldsymbol{W}_j^\top \boldsymbol{x}_i + b_j)) \end{aligned}$$

where L_i is a summation of K standard binary logistic regression losses. Following [10, 12, 11], binary classification can also be optimized in hyper-sphere space by removing the bias b_i and fixing all the binary classifier $||\boldsymbol{W}_j||_2 = 1$ and speaker embedding $||\boldsymbol{x}||_2 = 1$. Due to the lack of norm information, the variation range of cosine similarity is very small. So another parameter s is introduced to scale the cosine similarity for accelerating and stabilizing the optimization [12]:

$$\begin{aligned} L_i =& \log(1 + \exp(-s \cdot \cos(\theta_{y_i}))) \\ &+ \sum_{j\neq y_i}^{K} \log(1 + \exp(s \cdot \cos(\theta_j))) \end{aligned}$$

For K independent binary classifier and a sample x_i, they can only construct one positive sample and $K-1$ negative sample which is highly imbalanced. A simple but effective method is to introduce a weight parameter $\lambda \in [0, 1]$ to balance the gradients for positive and negative samples. Then, the loss function becomes:

$$\begin{aligned} L_i =& \lambda \log(1 + \exp(-s \cdot \cos(\theta_{y_i}))) \\ &+ (1-\lambda) \sum_{j\neq y_i}^{K} \log(1 + \exp(s \cdot \cos(\theta_j))) \end{aligned}$$

For multi-classification soft-based loss [10, 11, 12], the decision boundary among different classes is not unified since there exists a competition among different classifiers and the boundary will be largely affected by the neighbor classifiers. As for SphereFace2, it can avoid such competition by utilizing K independent binary classifiers, and then achieve a universal confidence threshold 0 ($s \cdot \cos(\theta_{y_i}) = 0$). However, it's difficult to achieve the universal threshold 0 in practice. Thus, the bias that was removed before comes back again to improve the training stability:

$$\begin{aligned} L_i =& \lambda \log(1 + \exp(-s \cdot \cos(\theta_{y_i}) - b)) \\ &+ (1-\lambda) \sum_{j\neq y_i}^{K} \log(1 + \exp(s \cdot \cos(\theta_j) + b)) \end{aligned}$$

where b means the bias term. Then, the bias b becomes the universal confidence threshold for all the binary classifiers and the decision boundary is turned into $s \cdot \cos(\theta_{y_i}) + b = 0$ which can increase the stability of training.

The introduction of large margin penalty [10, 11, 12] on decision boundary, which can enforce the intra-class tightness and the inter-class discrepancy, has boosted the verification performance significantly. Similarly, an additive angular margin is added to SphereFace2 framework on two sides including positive and negative samples. Then the loss function can be formulated as:

$$
\begin{aligned}
L_i = & \lambda \log(1 + \exp(-s \cdot (\cos(\theta_{y_i}) - m) - b)) \\
& + (1 - \lambda) \sum_{j \neq y_i}^{K} \log(1 + \exp(s \cdot (\cos(\theta_j) + m) + b))
\end{aligned}
$$

where m is the adjustable margin parameter which can be used to further tighten the boundary. The final decision boundary of positive and negative samples are $s \cdot (cos(\theta_{y_i}) - m) + b = 0$ and $s \cdot (cos(\theta_{y_i}) - m) + b = 0$ respectively.

A large inconsistency between the positive and negative pairs' score distribution is observed in [16], and this discrepancy will make it difficult to find a threshold to distinguish the positive pairs due to the large overlap. To tackle this problem, a similarity adjustment method is proposed to map from angle to similarity score during training for discriminative distribution. Then, the final loss function can be summarized as:

$$
\begin{aligned}
L_i = & \lambda \log(1 + \exp(-s \cdot (g(\cos(\theta_{y_i})) - m) - b)) \\
& + (1 - \lambda) \sum_{j \neq y_i}^{K} \log(1 + \exp(s \cdot (g(\cos(\theta_j)) + m) + b))
\end{aligned} \tag{4}
$$

where $g(z) = 2(\frac{z+1}{2})^t - 1$ is a mapping function to adjust the score similarity distribution.

It is noteworthy that all types of angular margins are compatible with SphereFace2. In addition to the Additive-type margin [12] in Equation 4, we also explore the ArcFace-type margin [11] and a combination of two types margins which are denoted as SphereFace-A and SphereFace-M respectively.

4. EXPERIMENTS SETUP

4.1. Dataset

In our experiment, we trained all the systems on the development set of Voxceleb2 [18], which contains 1,092,009 utterances among 5,994 speakers. The evaluation trials in our experiment include three cleaned version trials Vox1-O, Vox1-E and Vox1-H constructed from 1251 speakers in Voxceleb1 [17]. In addition, the validation trials from VoxSRC 2020 and VoxSRC 2021 are introduced to evaluate the performance on hard trials.

4.2. Training Detail

To explore the extreme performance of SphereFace2, three online data augmentation methods including adding noise [20], reverberation[1] and speed perturbation [21] are applied here for robust training. The length of training samples is 2 seconds and we extract 80-dimensional Fbank with 25ms length Hamming windows and 10ms window shift as the input feature, while no voice activity detection (VAD) is involved here. The encoder we adopt is 32 channels ResNet34 [22] with statistic pooling, and stochastic gradient descent (SGD) with momentum of 0.9 and weight decay of $1e$-4 is employed as the optimizer to train the model. The whole training process will last 150 epochs and the learning rate decrease from 0.1 to $1e$-5 exponentially. As for large margin fine-tuning [23], the initial learning rate is set to $1e$-4 and we train the models with only 5 epochs. It should be noted the margin is set to 0.35 and the segment duration increases to 6s in this stage.

4.3. Evaluation Metrics

For evaluation, we use cosine distance as the scoring criterion. After that, adaptive score normalization (A-snorm) [24] and quality-aware score calibration [23] applied for further improvements. Performance is measured in terms of the equal error rate (EER) and the minimum detection cost function (minDCF).

5. RESULTS AND ANALYSIS

5.1. Comparison between Different Loss Functions

In this section, we first give a results comparison between different loss functions in Table. 1. In [7], metric learning-based objectives are boosted and achieved competitive performance with the classification-based losses when there is no data augmentation. However, the metric learning-based objectives require large batch-size to mine enough negative pairs and are sensitive to the hard negative mining strategy. Besides, equipped with more extensive data augmentation and advanced training strategies, the results in Table .1 show that the classification-based losses have a more obvious advantage over the angular prototypical loss.

Among all the multi-classification losses, it's obvious to find a performance leap when the margin penalty is introduced to boost the discriminative power of speaker representation compared with traditional Softmax. In addition, SphereFace2 replaces the multi-classification with K binary classifiers, and trains the model in pair-wise learning paradigm. In Table. 1, SphereFace2 based loss functions surpass all other loss functions including metric- and softmax-based objectives, especially in hard trials (VoxSRC20-val and VoxSRC21-val). Moreover, the results of SphereFace2-A and SphereFace-M are also provided in Table. 1, and we observe that different types of angular margins perform well and obtain similar performance.

Large margin fine-tuning (LM-FT) [23] is a training strategy to further optimize the inter- and intro-class distance by enlarging the margin and duration, which is widely used in

[1] https://www.openslr.org/28

Table 1. **Voxceleb and VoxSRC results comparison between different loss functions.** LM-FT denotes the large margin fine-tuning strategy. For AM and AAM, the margin and scale are set to 0.2 and 32 respectively. And for circle loss, the margin and scale are 0.25 and 64 following the setup in [13]. For A-softmax, the margin is 4.

Loss Function	Vox-O		Vox-E		Vox-H		VoxSRC20-val		VoxSRC21-val	
	$\text{DCF}_{0.01}$	EER	$\text{DCF}_{0.01}$	EER	$\text{DCF}_{0.01}$	EER	$\text{DCF}_{0.05}$	EER	$\text{DCF}_{0.05}$	EER
Angular Prototypical [7]	0.2286	1.356	0.1927	1.468	0.2885	2.699	0.2366	4.100	0.3361	6.614
Softmax	0.1425	1.324	0.1532	1.292	0.2274	2.295	0.1955	3.549	0.2185	4.166
Circle Loss [14, 13]	0.1014	0.946	0.1166	1.031	0.1702	1.823	0.1547	2.878	0.1743	3.147
A-Softmax [10]	0.1200	0.984	0.1300	1.087	0.1992	1.930	0.1628	3.003	0.2115	3.771
AM-Softmax [12]	0.0914	**0.840**	0.1147	0.987	0.1743	1.796	0.1553	2.919	0.1875	3.453
AAM-Softmax [11]	0.0840	0.861	0.1122	0.996	0.1749	1.767	0.1531	2.830	0.1925	3.450
SphereFace2-A	**0.0690**	0.862	0.1069	0.993	**0.1649**	1.731	0.1494	2.761	**0.1686**	**2.836**
SphereFace2-M	0.0851	0.914	0.1182	1.059	0.1772	1.831	0.1576	2.901	0.1838	3.060
SphereFace2	0.0757	0.877	**0.1065**	**0.969**	0.1699	**1.726**	**0.1476**	**2.731**	0.1741	3.067
+ LM-FT	0.0571	0.670	0.0852	0.809	0.1384	1.424	0.1242	2.345	0.1376	2.362

building challenge systems. And we find that LM-FT is also compatible with SphereFace2 and leads to a great performance gain.

Finally, we provide an ablation study to analysis the effect of hyperparameters λ, t, s and m in SphereFace2 loss, and the results are listed in Table 2. According to the results, we observe that the SphereFace2 achieve the best performance under $\lambda = 0.7$, $t = 3$, $s = 32$ and $m = 0.2$.

Table 2. **Ablation study of hyperparameters λ, t, s and m.** Results are given with EER(%).

λ	t	s	m	Vox-O	Vox-E	Vox-H	VoxSRC20	VoxSRC21
0.7	3	32	0.2	0.877	**0.969**	**1.726**	**2.731**	**3.067**
0.8	3	32	0.2	**0.835**	0.976	1.728	2.780	3.083
0.7	2	32	0.2	0.808	0.989	**1.724**	2.821	3.183
0.7	3	32	0.2	0.877	**0.969**	1.726	**2.731**	**3.067**
0.7	4	32	0.2	**0.829**	0.980	1.750	2.834	3.160
0.7	3	24	0.2	**0.761**	1.000	1.800	2.849	3.407
0.7	3	32	0.2	0.877	**0.969**	**1.726**	**2.731**	**3.067**
0.7	3	$40^\dagger$	0.2	-	-	-	-	-
0.7	3	32	0.1	**0.824**	1.000	1.806	2.878	3.333
0.7	3	32	0.2	0.877	**0.969**	**1.726**	**2.731**	**3.067**
0.7	3	32	0.3	0.909	1.056	1.838	2.947	3.263

†: for parameter s, 40 is too large to train.

5.2. Robustness on Noisy Label

All the loss functions we discussed in section 5.1 are supervised training loss, which require precisely labeled data. Although the Voxceleb dataset is collected through an automated pipeline [17, 18], the authors found very few label errors after manual inspection [18]. Thus, it is curious how well these algorithms perform on data with noisy labels. In this section, we select the best performed multi-classification loss AAM-softmax and compare it with the SphereFace2 loss. In this experiment, we randomly select 30% of the data and change their labels before training. The corresponding results are shown in Table 3. From the results, we find that the performance of the models trained on data with noisy labels has a clear drop in performance. Surprisingly, the performance degradation of the model trained with SphereFace2 loss is much smaller than the model trained with AAM-softmax. This is because the multi-class softmax function has very strong supervision, which pushes all the similarities between embedding and non-target centers smaller than the similarity between embedding and target centers. However, this strong supervision loss can be counterproductive when there are some data with noisy labels. In comparison, SphereFace2 loss has weaker supervision and is more robust to the noisy labels.

Table 3. **EER(%) results of different loss functions trained on data with noisy labels.** It should be noted that A-snorm and score calibration are not used here because these strategies require accurate speaker labels.

Loss Function	AAM-softmax		SphereFace2	
Noisy Proportion(%)	0	30	0	30
Vox-O	1.058	2.005	1.032	1.638
Vox-E	1.147	2.106	1.060	1.665
Vox-H	2.087	3.744	1.907	2.931
VoxSRC20-val	3.398	5.669	3.120	4.646
VoxSRC21-val	4.074	5.743	3.373	4.798

6. CONCLUSION

In this paper, we introduce SphereFace2, a binary classification-based loss function for speaker verification to alleviate the mismatch between open-set training and close-set testing. Experiments conducted on Voxceleb show its leading performance compared with popular metric learning or multi-classification based loss functions. Moreover, the large margin fine-tuning strategy is applicable to further boost the performance. Finally, SphereFace2 also shows its strong robustness to class-wise noisy labels which has the potential to be applied in the semi-supervised training scenario with inaccurate estimated pseudo labels.

7. REFERENCES

[1] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Trans. ASLP*, vol. 19, no. 4, pp. 788–798, 2010.

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

[3] Bei Liu, Zhengyang Chen, Shuai Wang, Haoyu Wang, Bing Han, and Yanmin Qian, “DF-ResNet: Boosting Speaker Verification Performance with Depth-First Design,” in *Proc. ISCA Interspeech*, 2022, pp. 296–300.

[4] Shuai Wang, Yexin Yang, Yanmin Qian, and Kai Yu, “Revisiting the statistics pooling layer in deep speaker embedding learning,” in *Proc. ISCSLP*. IEEE, 2021, pp. 1–5.

[5] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, “Attentive statistics pooling for deep speaker embedding,” *arXiv preprint arXiv:1803.10963*, 2018.

[6] Miao Zhao, Yufeng Ma, Yiwei Ding, Yu Zheng, Min Liu, and Minqiang Xu, “Multi-query multi-head attention pooling and inter-topk penalty for speaker verification,” in *Proc. ICASSP*. IEEE, 2022, pp. 6737–6741.

[7] Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han, “In defence of metric learning for speaker recognition,” *arXiv preprint arXiv:2003.11982*, 2020.

[8] Xu Xiang, Shuai Wang, Houjun Huang, Yanmin Qian, and Kai Yu, “Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition,” in *Proc. APSIPA ASC*. IEEE, 2019, pp. 1652–1656.

[9] Yi Liu, Liang He, and Jia Liu, “Large margin softmax loss for speaker verification,” *arXiv preprint arXiv:1904.03479*, 2019.

[10] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song, “Sphereface: Deep hypersphere embedding for face recognition,” in *Proc. CVPR*, 2017, pp. 212–220.

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in *Proc. CVPR*, 2019, pp. 4690–4699.

[12] 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.

[13] Runqiu Xiao, “Adaptive margin circle loss for speaker verification,” *arXiv preprint arXiv:2106.08004*, 2021.

[14] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei, “Circle loss: A unified perspective of pair similarity optimization,” in *Proc. CVPR*, 2020, pp. 6398–6407.

[15] Li Ruida, Fang Shuo, Ma Chenguang, and Li Liang, “Adaptive rectangle loss for speaker verification,” *Proc. ISCA Interspeech*, pp. 301–305, 2022.

[16] Yandong Wen, Weiyang Liu, Adrian Weller, Bhiksha Raj, and Rita Singh, “Sphereface2: Binary classification is all you need for deep face recognition,” *arXiv preprint arXiv:2108.01513*, 2021.

[17] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, “Voxceleb: a large-scale speaker identification dataset,” *arXiv preprint arXiv:1706.08612*, 2017.

[18] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, “Voxceleb2: Deep speaker recognition,” *arXiv preprint arXiv:1806.05622*, 2018.

[19] Jake Snell, Kevin Swersky, and Richard Zemel, “Prototypical networks for few-shot learning,” *Proc. NIPS*, vol. 30, 2017.

[20] David Snyder, Guoguo Chen, and Daniel Povey, “MUSAN: A Music, Speech, and Noise Corpus,” 2015, arXiv:1510.08484v1.

[21] Jenthe Thienpondt, Brecht Desplanques, and Kris Demuynck, “The idlab voxceleb speaker recognition challenge 2020 system description,” *arXiv preprint arXiv:2010.12468*, 2020.

[22] 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.

[23] Jenthe Thienpondt, Brecht Desplanques, and Kris Demuynck, “The idlab voxsrc-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification,” in *Proc. ICASSP*. IEEE, 2021, pp. 5814–5818.

[24] Sandro Cumani, Pier Domenico Batzu, Daniele Colibro, Claudio Vair, Pietro Laface, and Vasileios Vasilakakis, “Comparison of speaker recognition approaches for real applications,” in *Interspeech*, 2011, pp. 2365–2368.