Semi-Supervised Far-Field Speaker Verification with Distance Metric Domain Adaptation

Han Wang*, Mingrui He*, Mingjun Zhang*, Longting Xu*
* School of Information Science and Technology, Donghua University, Shanghai, China
E-mail: wanghan65267@gmail.com, hemingrui2@outlook.com, zmj@mail.dhu.edu.cn, xlt@dhu.edu.cn

***Abstract*—Achieving high performance in far-field speaker verification is challenging due to the scarcity of data and prevalence of low-quality labels. A common approach is to use a semi-supervised learning framework. A pre-trained near-field speaker model extracts embeddings of unlabeled far-field speakers, and clustering methods generate pseudo-labels for fine-tuning. The accuracy of pseudo-label generation through clustering and the domain adaptation during fine-tuning are critical to this semi-supervised learning framework. In this paper, we propose a method that combines Uniform Manifold Approximation and Projection (UMAP) with a community detection algorithm for efficient dimensionality reduction and clustering. UMAP preserves the structural distribution of high-dimensional data, facilitating better clustering. We conduct domain alignment in the distance metric space to achieve domain adaptation during fine-tuning, referring to this approach as distance metric domain adaptation (DMDA). We evaluate the effectiveness of the dimensionality reduction community detection algorithm using the VoxCeleb (pre-trained) and FFSVC (fine-tuned) datasets. Experimental results demonstrate that the accuracy of generating pseudo labels is improved by using the reduced dimensional community detection algorithm. Additionally, the DMDA method effectively reduces domain mismatch issues during the fine-tuning process.**

I. Introduction

Speaker verification (SV) aims to verify whether two utterances are spoken by the same person [1], [2]. In recent years, deep learning has achieved significant success in speaker verification tasks [3], [4]. Speaker verification systems have shown remarkable improvement from the traditional i-vector method [5] to the DNN-based X-vector method [6]. Training X-vector-based speaker verification networks requires a large amount of well-labeled data. Due to difficulties in data recording and labeling, as well as noise interference, the performance of SV systems significantly degrades in far-field environments. Semi-supervised learning (SSL) [7] is commonly used to reduce the reliance on labeled data. A common SSL approach in far-field speaker verification involves first training an SV model using abundant near-field data. The well-trained near-field SV model is then used to extract speaker embeddings, which are clustered to obtain pseudo-labels. Finally, supervised fine-tuning is performed on the SV model using both pseudo-labeled far-field data and near-field data. However, the accuracy of the pseudo-labels largely depends on the robustness of the clustering algorithm and the reliability of the pre-trained model from the source domain. Additionally, domain mismatch between near-field and far-field data during fine-tuning also significantly affects the accuracy of the SV model.

Existing clustering methods typically use unsupervised approaches, such as K-means [8], Spectral Clustering [9], and Agglomerative Hierarchical Clustering (AHC) [10]. These methods rely on specific assumptions and require estimating the value of k in advance. For clustering large-scale datasets, traditional unsupervised methods and recent approaches like Graph Convolutional Networks (GCN) [11] and community detection algorithms [12] are commonly used. Chen et al. [13] propose a semi-supervised learning approach to improve speaker identification accuracy by label propagation on a graph encoding pairwise similarity for all labeled and unlabeled utterances. Tong et al. [14] apply graph convolutional networks to the constructed affinity graph, fully utilizing the local subgraph information to obtain the representation of speaker embeddings. Recently, community detection algorithms [15]–[17] have also been performing well in large-scale unsupervised speaker clustering tasks. Chen et al. [18] utilize Infomap to perform clustering and obtain the initial pseudo labels. The Leiden [17] algorithm also ensures that all communities are well-connected in speaker clustering.

To reduce the domain mismatch problem, recent approaches have been proposed in the field of speaker verification. One common approach is to improve the fine-tuning strategies. X. Qin et al. [19] use both source and target domain data to fine-tune the pre-trained model jointly. Y. Zheng et al. [20] reserve the speaker weights in the pre-training stage, which are subsequently utilized in the fine-tuning stage. Another approach focuses on enhancing the back-end processing techniques in the SV system. Unsupervised PLDA [21] adapts the PLDA model's covariance matrices to align with domain-invariant data. Aligning embedding vectors across multiple domains by minimizing inter-domain cosine distance or maximizing the mean discrepancy (MMD) [22] is also a common domain adaptation method. Recently, methods that align the distribution of distances within and between speakers have been shown to better match the data distribution for the SV task [23], [24]. Additionally, learning domain-invariant speaker representations [25]–[27] is another notable method for reducing domain mismatch.

In this paper, we employ the TAU community detection algorithm, combined with Uniform Manifold Approximation and Projection (UMAP) for dimensionality reduction to achieve the most efficient and accurate clustering results. To mitigate the issue of domain mismatch, we propose distance metric domain adaptation (DMDA), conducting domain alignment in

the distance metric space and to use Discriminative Joint Probability Maximum Mean Discrepancy (DJP-MMD) to measure the intra-speaker and inter-speaker differences. Experimental results on the FFSVC dataset demonstrate the improvement in clustering performance and the effectiveness of the DMDA method in reducing domain mismatch during the fine-tuning process.

II. Methods

A. Uniform Manifold Approximation

UMAP is a dimensionality reduction algorithm based on graph theory and manifold learning that maps high-dimensional data to low-dimensional space for visualization and analysis. The algorithm relies on a theoretical framework involving Riemannian geometry and algebraic topology. UMAP learns a mapping from the high-dimensional data distribution to a low-dimensional space, preserving the significant topological structures of the original high-dimensional space. We employ UMAP to learn the underlying manifold structure and construct a weighted k-nearest neighbor graph.Let $X = \{x_1, x_2, \ldots, x_n\}$ denote the extracted unlabeled speaker embeddings, where the connectivity of the neighborhood graph is determined by the edge weights. To construct the weighted k-nearest neighbor graph, we can calculate the weights between the speaker embeddings and their k nearest neighbors as follows:

$$w\left(\left(x_i, x_{i_j}\right)\right) = \exp\left(\frac{-\max\left(0, d\left(x_i, x_{i_j}\right) - \rho_i\right)}{\sigma_i}\right) \quad (1)$$

where $d(x_i, x_{i_j})$ denotes the distance between embeddings x_i and x_{i_j}, ρ_i is a distance threshold for embedding x_i, and σ_i is a scaling parameter for embedding x_i.

After approximating the manifold in the high-dimensional space, the next step in UMAP is to project it into a low-dimensional space. Once the manifold structure has been learned, it is necessary to define a minimum distance between the embedded points, to avoid multiple overlapping points. UMAP can find a good low-dimensional manifold representation by minimizing the cross-entropy cost function. Denoting the adjacency matrix of the graph as E, our ultimate goal is to find the optimal edge weights for the low-dimensional representation. These optimal weights are obtained by minimizing the cross-entropy function:

$$CE = \sum_{e \in E} w_h(e) \log\left(\frac{w_h(e)}{w_l(e)}\right) + (1 - w_h(e)) \log\left(\frac{1 - w_h(e)}{1 - w_l(e)}\right) \quad (2)$$

where $w_h(e)$ and $w_l(e)$ denote the high-dimensional and low-dimensional edge weights, respectively.

B. TAU Community Detection Agorithm

We use the TAU community detection algorithm for clustering. This algorithm is an efficient method for modularity optimization, combining the greedy optimization approaches of Louvain and Leiden with genetic algorithms to better explore the solution space. We set each speaker embedding as a node in the community detection network, using the cosine similarity scores between embeddings as edge weights. The TAU algorithm utilizes the Leiden algorithm to detect communities in the input graph through modularity optimization. The modularity optimization function used by Leiden is expressed as follows:

$$\mathcal{Q} = \frac{1}{2m} \sum_i \sum_j \left(A_{ij} - \frac{k_i k_j}{2m}\right) \delta\left(C^{(i)}, C^{(j)}\right) \quad (3)$$

where $\mathcal{Q}$ is the modularity score, m is the total number of edges, A_{ij} represents the elements of the adjacency matrix, k_i and k_j are the degrees of nodes i and j, respectively, and $\delta(C^{(i)}, C^{(j)})$ is 1 if nodes i and j are in the same community, and 0 otherwise.

After the initialization step of generating initial individual partitions, the algorithm iteratively performs the following steps:

Step1. Applying Leiden to optimize each individual partition.

Step2. Evaluating fitness and selecting individuals.

Step3. Performing crossover and mutation on the population.

Step4. Migrating new individuals into the population.

C. Distance Matric Domain Adaptation

As shown in Fig. 1, We aim to reduce the domain mismatch issues in the SV system during the fine-tuning process.

1) Pre-training: In the pre-training stage, we train a deep speaker embedding model using out-of-domain data. To enhance the model's discrimination ability and improve generalization, we employ the angular margin softmax (AM-softmax) [28] loss function.

2) DJP-MMD for domain adaptation: The issue of label mismatch between the source domain and the target domain cannot be resolved merely by aligning distributions at the feature level, which may prevent the model from learning domain-invariant features. After obtaining the pre-trained SV model, we use the speaker labels from the source domain and the pseudo-labels from the target domain to construct sample pairs both within and between speakers. We then use cosine distance to estimate the distance distribution for these sample pairs. We measure the distance between the source domain and the target domain distributions using DJP-MMD.

$$d(\mathcal{D}_s, \mathcal{D}_t) = \mathcal{M}_T - \mu \mathcal{M}_D, \quad (4)$$

where $\mu > 0$ is a trade-off parameter. M_T measures the transferability between the same class in different domains, and M_D measures the discriminability between different classes in different domains.

We compute M_T using MMD, aiming to measure the overall distribution difference between the source domain $p(s)$ and the target domain $p(t)$.

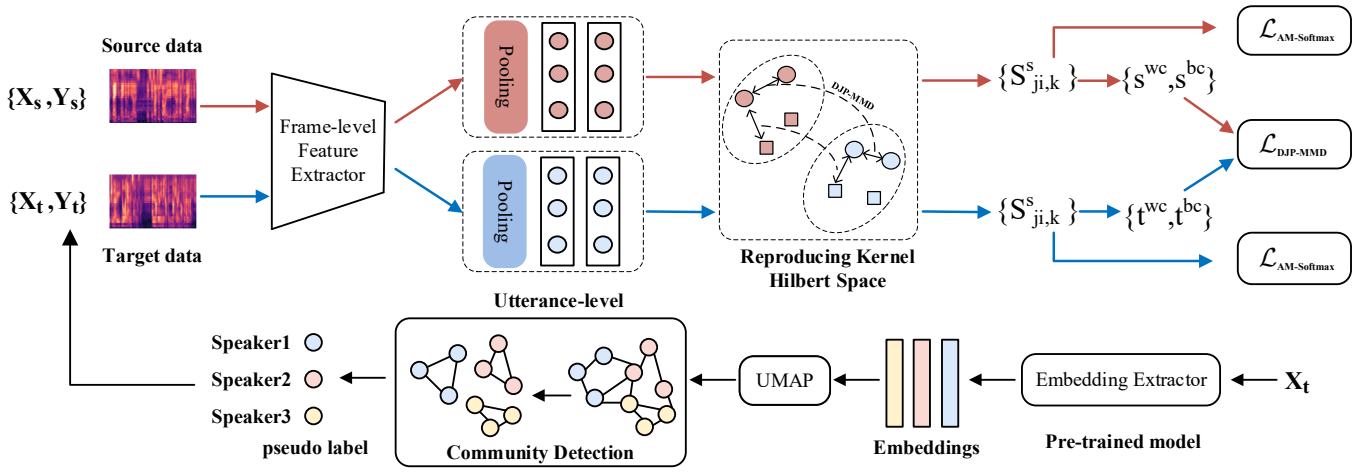

Fig. 1. Overview of the proposed DMDA method with pairwise-distance distribution alignment.

$$
\begin{aligned}
M_T(p(s),p(t)) =& \frac{1}{T_s^2}\sum_{n=1}^{T_s}\sum_{m=1}^{T_s} k(s_n,s_m) \\
&+ \frac{1}{T_t^2}\sum_{n=1}^{T_t}\sum_{m=1}^{T_t} k(t_n,t_m) \\
&- \frac{2}{T_sT_t}\sum_{n=1}^{T_s}\sum_{m=1}^{T_t} k(s_n,t_m)
\end{aligned} \tag{5}
$$

where T_s and T_t are the numbers of samples from the source and target domains in a mini-batch. $k(s,t)$ denotes the radial basis function (RBF) kernel.

$$
\begin{aligned}
M_T(p(s),p(t)) =& \frac{1}{C}\sum_{c=1}^{C}\Bigg(\frac{1}{T_{s,c}^2}\sum_{n=1}^{T_{s,c}}\sum_{m=1}^{T_{s,c}} k(s_{c,n},s_{c,m}) \\
&+ \frac{1}{T_{t,c}^2}\sum_{n=1}^{T_{t,c}}\sum_{m=1}^{T_{t,c}} k(t_{c,n},t_{c,m}) \\
&- \frac{2}{T_{s,c}T_{t,c}}\sum_{n=1}^{T_{s,c}}\sum_{m=1}^{T_{t,c}} k(s_{c,n},t_{c,m})\Bigg)
\end{aligned} \tag{6}
$$

where C is the total number of classes. $T_{s,c}$ and $T_{t,c}$ are the numbers of samples from the source and target domains in class. $s_{c,n}$ and $s_{c,m}$ are samples from the source domain in class. $t_{c,n}$ and $t_{c,m}$ are samples from the target domain in class.

The vectors s and t can represent either within-class or between-class distance representations. After computing the two DJP-MMD distances, we obtain the final DJP-MMD loss as follows:

$$
\begin{aligned}
\mathcal{L}_{\text{DJP-MMD}} =& \mathcal{L}_{\text{DJP-MMD}}(p(\mathbf{s}^{wc}),p(\mathbf{t}^{wc})) \\
&+ \mathcal{L}_{\text{DJP-MMD}}(p(\mathbf{s}^{bc}),p(\mathbf{t}^{bc}))
\end{aligned} \tag{7}
$$

where $\mathbf{s}^{wc}$ and $\mathbf{t}^{wc}$ are the within-class distance vectors from the source and target domains, respectively. $\mathbf{s}^{bc}$ and $\mathbf{t}^{bc}$ are the between-class distance vectors from the source and target domains, respectively.

The overall objective of DMDA is the weighted sum of the AM-softmax loss and the DJP-MMD losses:

$$
\mathcal{L}_{\text{DMDA}} = \mathcal{L}_{\text{DJP-MMD}} + \lambda\mathcal{L}_{\text{AM-softmax}} \tag{8}
$$

III. EXPERIMENTS

A. Datasets and Evaluation Metrics

We conduct experiments on the VoxCeleb 1&2 [29] and FFSVC [30] datasets. The VoxCeleb 1&2 dataset serves as the large-scale dataset for pre-training near-field models. The FFSVC2022 supplement is used as the unlabeled far-field target domain dataset. We evaluate the model on the development trials and evaluation trials of FFSVC2022. We strictly adhere to the competition requirements of FFSVC2022 Task 2, where adjustments to hyperparameters and testing model performance are only allowed on the development set. All data used is consistent with the official baseline for FFSVC2022. All trial pairs are single-channel speech segments. The evaluation set consists of a large trials file and anonymized audios. The experimental results are evaluated using the Equal Error Rate (EER) and the minimum of the normalized detection cost function (minDCF) with $P_{target} = 0.01$.

B. Implementation Details

We extract 80-dimensional log-mel filter bank features. The frame length for all features is set to 25ms with a frame shift of 10ms. Furthermore, we perform mean normalization within a sliding window of three seconds.

In the pre-training stage, the chunk-size is set to 200 for sampling. We utilize the AdamW optimizer [31] with a momentum of 0.9 and a weight decay of 5e-1. Additionally, the margin gradually increases from 0 to 0.2. We use 1 GPU with a mini-batch size of 512 and an initial learning rate of 0.01 to train all of our models.

For community detection, we first use UMAP to reduce the dimensionality by setting the number of neighbors in UMAP to 20, allowing it to find low-dimensional manifolds in the high-dimensional space. The UMAP algorithm reduces the dimensionality to 60 in the new feature space. We use cosine distance as the distance metric to compute sample similarities, applying it consistently across all clustering methods. We extract 40, 80, and 120 speakers from the supplement dataset to compare the clustering performance under different numbers of speakers.

For domain adaptation, we use a speaker-balanced sampling strategy to construct intra-speaker and inter-speaker pairs, ensuring that each speaker in a mini-batch contains the same number of utterances. The batch size is set to 128, with each speaker having 4 utterances. ResNet34 is chosen as the base backbone. Cosine similarity is used for backend scoring, and all experiments are conducted on ASV-subtools [32] and Wespeaker [33].

IV. RESULTS AND DISCUSSIONS

A. The Effectiveness of community detection

As shown in Table 1, we employ four different clustering methods with varying numbers of speakers and evaluate their performance based on average precision, recall, and F-score. When the number of speakers is 40, the traditional K-means method performs well, achieving accuracy surpassing that of the basic community detection algorithm. When the number of speakers increases to 80, the average precision of traditional K-means clustering decreases significantly. At 120 speakers, the advantage of the community detection algorithm becomes more apparent. The best clustering results are achieved when TAU and UMAP are used together for dimensionality reduction, and this method maintains good performance as the number of speakers increases.

TABLE I
COMPARISON OF SPEAKER CLUSTERING PERFORMANCE FOR VARIOUS NUMBER OF SPEAKERS

Nums	Methods	Precision	Recall	F-score
40	K-means	0.88	0.79	0.80
	Infomap	0.84	0.77	0.80
	Leiden	0.85	0.78	0.82
	TAU	0.86	0.79	0.80
	UMAP+TAU	0.90	0.83	0.87
80	K-means	0.85	0.77	0.81
	Infomap	0.83	0.74	0.80
	Leiden	0.84	0.76	0.82
	TAU	0.85	0.77	0.82
	UMAP+TAU	0.89	0.81	0.85
120	K-means	0.82	0.72	0.74
	Infomap	0.83	0.76	0.77
	Leiden	0.85	0.76	0.79
	TAU	0.87	0.78	0.82
	UMAP+TAU	0.89	0.80	0.84

B. The Effectiveness of DMDA

The experimental results are shown in Table 2. The table compares the performance of different methods on the FFSVC2022 development and evaluation datasets in terms of EER and minDCF. We compare the DMDA method with the FFSVC official baseline, the methods proposed by other participating teams, and the Unsupervised PLDA approach. Vanilla fine-tuning (using only the near-field dataset for fine-tuning) shows improvement over the pre-trained model. This indicates that fine-tuning the near-field speaker verification (SV) model with a far-field dataset is an effective approach. However, this process may lead to overfitting and catastrophic forgetting. The use of domain adaptation techniques further enhances the SV model's performance. Unsupervised PLDA effectively reduces domain mismatch problems. When we use our proposed DMDA method, the performance improves further. Notably, we achieve the best results with the DJP-MMD distance.

TABLE II
PERFORMANCE COMPARISON OF DMDA AND OTHER METHODS ON FFSVC2022 DATASETS

Method	FFSVC2022 dev		FFSVC2022 eval	
	EER%	minDCF	EER%	minDCF
Official baseline	7.79	0.746	7.64	0.739
rank2 HiMia	5.59	0.563	5.34	0.566
rank1 SPEAKIN	5.98	0.500	6.21	0.523
Pre-trained	8.96	0.751	-	-
Vanilla fine-tuning	6.74	0.749	7.14	0.695
Unsupervised PLDA	6.15	0.614	6.59	0.644
DMDA(MMD)	5.75	0.524	5.72	0.558
DMDA(DJP-MMD)	**5.24**	**0.507**	**5.21**	**0.535**

V. CONCLUSIONS

In this paper, we present a novel approach to enhance far-field speaker verification through a semi-supervised learning framework. By integrating Uniform Manifold Approximation and Projection (UMAP) with a community detection algorithm, we achieve efficient dimensionality reduction and clustering while preserving the structural distribution of high-dimensional data. Furthermore, we propose a distance metric domain adaptation (DMDA) method to address domain mismatch issues during fine-tuning by aligning domains in the distance metric space. We compare the accuracy of various clustering methods with different numbers of speakers and contrast the DMDA method with several domain adaptation techniques. Experimental results demonstrate that our proposed method effectively improves the performance of far-field speaker verification.

ACKNOWLEDGMENT

This work is supported by the Donghua University Language and Text Application Research Project (No. Y2024-8).

REFERENCES

[1] F. Bimbot, J.-F. Bonastre, C. Fredouille, *et al.*, "A tutorial on text-independent speaker verification," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, pp. 1–22, 2004.

[2] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5115–5119.

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

[4] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," *Neural Networks*, vol. 140, pp. 65–99, 2021.

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

[6] 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)*, 2018, pp. 5329–5333.

[7] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine learning*, vol. 109, no. 2, pp. 373–440, 2020.

[8] G. Hamerly and C. Elkan, "Learning the k in k-means," *Advances in neural information processing systems*, vol. 16, 2003.

[9] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[10] S. Zhou, Z. Xu, and F. Liu, "Method for determining the optimal number of clusters based on agglomerative hierarchical clustering," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 12, pp. 3007–3017, 2016.

[11] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *International conference on machine learning*, PMLR, 2020, pp. 1725–1735.

[12] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 80, no. 5, p. 056 117, 2009.

[13] L. Chen, V. Ravichandran, and A. Stolcke, "Graph-Based Label Propagation for Semi-Supervised Speaker Identification," in *Proc. Interspeech 2021*, 2021, pp. 4588–4592.

[14] F. Tong, S. Zheng, M. Zhang, *et al.*, "Graph convolutional network based semi-supervised learning on multi-speaker meeting data," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 6622–6626.

[15] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the national academy of sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[16] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, P10008, 2008.

[17] V. A. Traag, L. Waltman, and N. J. Van Eck, "From louvain to leiden: Guaranteeing well-connected communities," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[18] Z. Chen, J. Wang, W. Hu, L. Li, and Q. Hong, "Unsupervised speaker verification using pre-trained model and label correction," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, pp. 1–5.

[19] X. Qin, D. Cai, and M. Li, "Far-Field End-to-End Text-Dependent Speaker Verification Based on Mixed Training Data with Transfer Learning and Enrollment Data Augmentation," in *Proc. Interspeech 2019*, 2019, pp. 4045–4049.

[20] Y. Zheng, J. Peng, Y. Chen, *et al.*, "The speakin speaker verification system for far-field speaker verification challenge 2022," *arXiv:2209.11625*, 2022.

[21] N. Brummer, A. Mccree, S. Shum, D. Garcia-Romero, and C. Vaquero, "Unsupervised Domain Adaptation for I-Vector Speaker Recognition," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2014)*, 2014, pp. 260–264.

[22] J. Huang and T. Bocklet, "Intel Far-Field Speaker Recognition System for VOiCES Challenge 2019," in *Proc. Interspeech 2019*, 2019, pp. 2473–2477. DOI: 10.21437/Interspeech.2019-2894.

[23] L. Yi and M. W. Mak, "Cross-domain adaptation in distance space for speaker verification," in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2023, pp. 2238–2243.

[24] J. Li, J. Han, and H. Song, "Cdma: Cross-domain distance metric adaptation for speaker verification," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 7197–7201.

[25] Y. Wei, J. Du, H. Liu, and Z. Zhang, "Centriforce: Multiple-domain adaptation for domain-invariant speaker representation learning," *IEEE Signal Processing Letters*, vol. 29, pp. 807–811, 2022.

[26] Q. Wang, W. Rao, S. Sun, L. Xie, E. S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4889–4893.

[27] W. Huang, B. Han, S. Wang, Z. Chen, and Y. Qian, "Robust cross-domain speaker verification with multi-level domain adapters," in *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 11 781–11 785.

[28] 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.

[29] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.

[30] X. Qin, M. Li, H. Bu, S. Narayanan, and H. Li, "The 2022 far-field speaker verification challenge: Exploring domain mismatch and semi-supervised learning under the far-field scenario," *arXiv preprint arXiv:2209.05273*, 2022.

[31] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.

[32] F. Tong, M. Zhao, J. Zhou, *et al.*, "Asv-subtools: Open source toolkit for automatic speaker verification," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6184–6188.

[33] H. Wang, C. Liang, S. Wang, *et al.*, "Wespeaker: A research and production oriented speaker embedding learning toolkit," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, pp. 1–5.