Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

Generalizable Audio Deepfake Detection via Hierarchical Structure Learning and Feature Whitening in Poincaré sphere

Mingru Yang[*1], *Yanmei Gu*[*2], *Qianhua He* [†1], *Yanxiong Li* [†1], *Peirong Zhang*[1], *Yongqiang Chen*[1], *Zhiming Wang*[2], *Huijia Zhu* [†2], *Jian Liu*[2], *Weiqiang Wang*[2]

[1]School of Electronic and Information Engineering, South China University of Technology, China
[2]Ant Group, China

eemryang@mail.scut.edu.cn, eeqhhe@scut.edu.cn, eeyxli@scut.edu.cn

Abstract

Audio deepfake detection (ADD) faces critical generalization challenges due to diverse real-world spoofing attacks and domain variations. However, existing methods primarily rely on Euclidean distances, failing to adequately capture the intrinsic hierarchical structures associated with attack categories and domain factors. To address these issues, we design a novel framework **Poin-HierNet** to construct domain-invariant hierarchical representations in the Poincaré sphere. Poin-HierNet includes three key components: 1) Poincaré Prototype Learning (PPL) with several data prototypes aligning sample features and capturing multilevel hierarchies beyond human labels; 2) Hierarchical Structure Learning (HSL) leverages top prototypes to establish a tree-like hierarchical structure from data prototypes; and 3) Poincaré Feature Whitening (PFW) enhances domain invariance by applying feature whitening to suppress domain-sensitive features. We evaluate our approach on four datasets: ASVspoof 2019 LA, ASVspoof 2021 LA, ASVspoof 2021 DF, and In-The-Wild. Experimental results demonstrate that Poin-HierNet exceeds state-of-the-art methods in Equal Error Rate.
Index Terms: audio deepfake detection, anti-spoofing, generalization, hierarchical structure learning, feature whitening

1. Introduction

Audio deepfake detection (ADD) is critical for protecting speaker verification systems from various attacks, including speech synthesis, voice conversion and voice cloning. Significant progress has been made in developing both front-end features [1, 2, 3] and back-end models [4, 5, 6], achieving promising results in intra-database deepfake detection. However, generalizing to diverse unseen domains remains challenging, particularly in real-world scenarios with diverse, unpredictable attacks and spoofed utterances spanning various domains like channels, codecs, and compression formats [7, 8].

To develop generalizable deepfake detectors, existing methods focus on two directions: 1) refining learning strategies to capture complex forgery patterns, and 2) using data augmentation to simulate unseen attacks. The first includes domain adversarial learning [9] and knowledge distillation [10, 11] for domain-invariant feature extraction, one-class learning [12, 13] for bonafide audio alignment, contrastive learning [14, 15] for spoof-genuine discrimination, and prototype-based methods [16] for feature space modeling. Data augmentation methods introduce perturbations, including speed perturbation, SpecAugment [17], and RawBoost [18]. However, as shown in Fig. 1(a), all these methods operate in Euclidean space and rely on human-labeled classes, neglecting the intrinsic hierarchy of ADD data. This hierarchical structure is evident because attack algorithms form distinct categories, whereas domain factors (e.g., channels, codecs, compression methods) define finer subcategories. Leveraging this structure is key to understanding the data and improving model generalization against unseen attacks, yet it remains unexplored in audio deepfake detection.

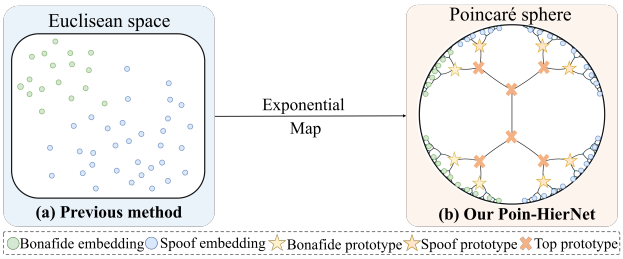

Figure 1: *Motivation of Poin-HierNet. Previous works represent data distribution in Euclidean space only based on human-labeled classes, whereas Poin-HierNet constructs hierarchical feature space in the Poincaré ball model.*

In this paper, we introduce the Poincaré ball model, which is inherently well-suited for representing hierarchical data structures [19, 20]. Then we propose **Poin-HierNet** (Fig. 1(b)), a framework that constructs a domain-invariant hierarchical feature space in the Poincaré sphere. Poin-HierNet comprises three key learning processes: Poincaré Prototype Learning (PPL), Hierarchical Structure Learning (HSL), and Poincaré Feature Whitening (PFW). PPL introduces a set of learnable data prototypes to represent the overall feature distribution, aligning these prototypes with sample features using the constraint $\mathcal{L}_{PPL}$. Multiple prototypes are assigned to each bonafide and spoof class, thereby capturing hierarchical structures beyond human-labeled categories. HSL integrates the top prototypes and leverages the Poincaré ball model to explore the correlations between data prototypes and top prototypes, thus establishing a tree-like hierarchical structure. PFW applies feature whitening to suppress domain-sensitive features, ensuring the learned hierarchical feature space remains domain-invariant.

In summary, our main contributions are as follows. We propose Poin-HierNet, a novel framework that constructs a domain-invariant hierarchical feature space. To the best of our knowledge, this is the first method that addresses generalizable audio deepfake detection from the perspective of hierarchical space. The Poin-HierNet consists of PPL, HSL, and PFW. PPL and HSL model the data distribution and establish a hierarchical structure through prototype learning, while PFW applies feature whitening to ensure that this hierarchical structure remains domain-invariant. Experimental results on multiple datasets

* Equal Contribution.
† Corresponding Authors.

10.21437/Interspeech.2025-1594

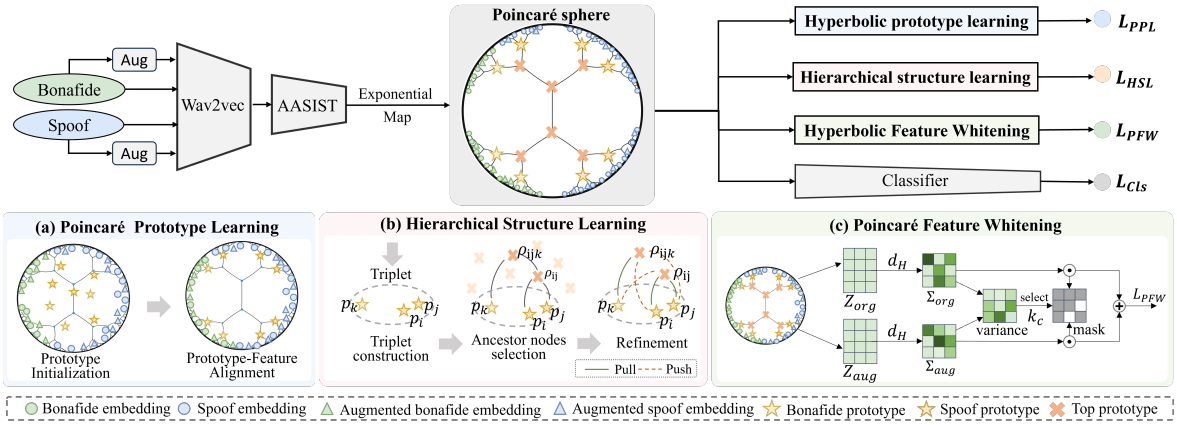

Figure 2: *Overall framework of the proposed Poin-HierNet.* ***TOP****: Illustration of the training procedures for Poin-HierNet.* ***Bottom****: Schematic of Poincaré Prototype Learning (**a**), Hierarchical Structure Learning (**b**), and Poincaré Feature Whitening (**c**) components.*

verify that our method outperforms state-of-the-art methods in terms of Equal Error Rate (EER).

2. Method

The overall architecture of the proposed Poin-HierNet is depicted in Fig. 2. Following prior studies [1, 16], we utilize the wav2vec 2.0 XLS-R (0.3B) [21] as the frontend feature extractor and AASIST [4] as the backend model. Bonafide and spoof audio samples are processed by wav2vec 2.0 XLS-R and AASIST, obtaining high-dimensional features in Euclidean space. We use paired inputs, specifically the original sample and its augmentation, both processed identically. These features are then mapped onto the Poincaré sphere via exponential mapping. To construct domain-invariant hierarchical representations in the Poincaré sphere, PPL represents the overall data distribution through prototype learning by enhancing prototype-feature and original-augmentation alignments; HSL introduces triplet construction to establish the hierarchical structure; and PFW calculates a mask matrix to suppress domain-sensitive features, ensuring that the hierarchical feature space remains domain-invariant. Finally, the overall loss is integrated for optimization.

2.1. The Poincaré ball model

Let $\mathbb{E}^n$ and $\mathbb{H}^n$ denote the n-dimensional Euclidean space and hyperbolic space, respectively. The Poincaré ball model $(\mathbb{H}^n_c, g^{\mathbb{H}})$ is given by the manifold:

$$\mathbb{H}^n_c = \{\mathbf{x} \in \mathbb{E}^n : c\|\mathbf{x}\| < 1\}, \tag{1}$$

equipped with the Riemannian metric:

$$g^{\mathbb{H}} = \left(\frac{2}{1 - c\|\mathbf{x}\|^2}\right)^2 g^{\mathbb{E}}, \tag{2}$$

where c is the curvature hyperparameter, $\|\cdot\|$ denotes the Euclidean norm, $g^{\mathbb{E}} = \mathbf{I}_n$ is the Euclidean metric.

The mapping function converting Euclidean embeddings to Poincaré embeddings via the exponential map is defined as:

$$\exp_{\mathbf{0}}^c(\mathbf{v}) = \tanh\sqrt{c}\|\mathbf{v}\|\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}. \tag{3}$$

The distance between points $\mathbf{u}, \mathbf{v} \in \mathbb{H}^n$ is given as:

$$d_H(\mathbf{u}, \mathbf{v}) = arcosh\left(1 + 2\frac{\|\mathbf{u} - \mathbf{v}\|^2}{(1 - \|\mathbf{u}\|^2)(1 - \|\mathbf{v}\|^2)}\right). \tag{4}$$

2.2. Poincaré prototype learning (PPL)

To facilitate the construction of a hierarchical structure within the Poincaré sphere, we introduce prototype learning. Distinct learnable prototype sets are employed for each category to represent their essential features. Specifically, K_b bonafide prototypes and K_s spoofing prototypes are established to represent the typical characteristics of genuine and manipulated features, respectively. These trainable prototypes are randomly initialized in the Poincaré sphere and formally defined as $P = \{p_j^r \in \mathbb{R}^D\}$, where $j \in \{0, 1\}$ denotes the class type (with 0 representing bonafide samples and 1 representing spoof samples). The index r ranges from 1 to K_j, where K_j corresponds to the number of prototypes for each class (i.e.,$K_j = K_b$ for bonafide samples when $j = 0$, and $K_j = K_s$ for spoof samples when $j = 1$). Here, D represents the dimension of the feature space.

Inspired by [22, 23], we introduce the following loss for Poincaré prototype learning:

$$\mathcal{L}_{proto} = -\sum_{n=1}^{N} \log \frac{\exp(-d_H(z_n, p_{y_n}^k))}{\sum_{r=1}^{K_j} \exp(-d_H(z_n, p_j^r))}, \tag{5}$$

where d_H is the hyperbolic distance defined in Eq.(4), $z_n \in \mathbb{R}^D$ represents the embedding of the n-th sample, y_n denotes its label, and $p_{y_n}^k$ is the prototype associated with the n-th sample, i.e., the closest prototype from $\{p_{y_n}^r\}$ to that sample within the Poincaré ball model. As shown in Fig. 2(a), this constraint brings the features close together with their corresponding prototypes while driving them far apart from non-corresponding prototypes.

Moreover, to ensure that the learned prototypes effectively generalize to unknown encoding and transmission conditions, we implement the following loss function to enforce the augmented feature $z^{aug} \in \mathbb{R}^D$ aligns with the original feature z:

$$\mathcal{L}_{aug} = \sum_{n=1}^{N} (d_H(z_n, z_n^{aug}) + \|d_H(z_n, p_{y_n}^k) - d_H(z_n^{aug}, p_{y_n}^k)\|_2). \tag{6}$$

This loss minimizes the distance between z_n and z_n^{aug} as well as the discrepancy between their distances to the corresponding prototype, thus learning task-specific features and enhancing Poincaré prototype representations. The total loss for

Poincaré prototype learning is:

$$\mathcal{L}_{PPL} = \mathcal{L}_{proto} + \mathcal{L}_{aug}. \tag{7}$$

2.3. Hierarchical structure learning (HSL)

Based on the data prototypes learned in Poincaré sphere that capture the sample distribution, we subsequently explore various latent semantic hierarchies within the Poincaré prototypes by modeling a tree-like hierarchical structure. We draw inspiration from [20] and employ a series of learnable top prototypes, $P_{top} = \{\rho_k | k \in \{0, 1, \ldots, K_{top}\}\}$, as higher-level nodes. As shown in Fig. 2(b), HSL consists of three stages: triplet construction, ancestor nodes selection, and hierarchical distance refinement.

Triplet construction. To capture structural information beyond labels, we omit label details and flatten the learned Poincaré prototypes as $P = \{p_n | n \in \{0, 1, \ldots, (K_b + K_s)\}\}$. Subsequently, we randomly select a prototype as the anchor p_i. A positive pair is constructed by choosing a random sample p_j from its K-nearest neighbors $R_K(p_i)$, while a negative pair is formed by randomly selecting a prototype p_k from outside the neighborhood $R_K(p_i)$. The triplets are expressed as:

$$T = \{(p_i, p_j, p_k) | (p_j \in R_K(p_i)) \wedge (p_k \notin R_K(p_i))\}. \tag{8}$$

Ancestor nodes selection. Higher-level nodes are identified within the top-level prototypes P_{top}. As illustrated in Fig. 2(b), in the triplet, p_i and p_j encouraged to share the same lowest common ancestor (LCA) ρ_{ij}, while p_i and p_k have different LCAs. The selection of ρ_{ij} from P_{top} follows:

$$\rho_{ij} = \arg\max_{\rho}(e^{(-max\{d_H(p_i,\rho), d_H(p_j,\rho)\})} + g_{ij}), \tag{9}$$

where $e^{(-max\{d_H(p_i,\rho), d_H(p_j,\rho)\})}$ represents the probability of ρ being the ideal representative ρ_{ij}, and g_{ij} is a noise term sampled from a Gumbel(0,1) distribution to mitigate the risk of local optima. The same approach as in Eq. (9) is used to select the LCA of ρ_{ij} and ρ_k, denoted as ρ_{ijk}, as a higher-level node than ρ_{ij}.

Hierarchical distance refinement. The optimization based on the distance relationship between prototypes and their higher-level prototype ancestors is as follows:

$$\begin{aligned} \mathcal{L}_{HSL} = &[d_H(p_i, \rho_{ij}) - d_H(p_i, \rho_{ijk}) + \delta] \\ &+ [d_H(p_j, \rho_{ij}) - d_H(p_j, \rho_{ijk}) + \delta] \\ &+ [d_H(p_k, \rho_{ijk}) - d_H(p_k, \rho_{ij}) + \delta], \end{aligned} \tag{10}$$

where δ represents the margin.

2.4. Poincaré feature whitening (PFW)

To capture domain-invariant features at a finer granularity in the Poincaré sphere, we propose Poincaré Feature Whitening (PFW). PFW imposes explicit constraints on the correlations among feature dimensions, suppressing those sensitive to domain variations while accentuating the insensitive ones.

Prior works [24, 25, 26] have demonstrated that the feature covariance matrix captures domain-specific features in Euclidean space; however, due to the distinct geometry and metric of Poincaré sphere, the conventional covariance matrix cannot be directly computed. To address this issue, we leverage the hyperbolic metric by replacing the covariance matrix with a similarity matrix computed across different feature dimensions. The similarity matrix is defined as:

$$\Sigma = d_H(Z, Z^\top) \in \mathbb{R}^{D \times D}, \tag{11}$$

where $Z^\top \in \mathbb{R}^{B \times D}$ represents the output features, with B indicating the batch size and D denoting the feature dimension.

As shown in Fig. 2(c), PFW is computed as follows. For the original and augmented features, the similarity matrices Σ_{org} and Σ_{aug} are computed using eq. (11), respectively. Next, we derive a mask M to identify positions in the similarity matrix that are sensitive to domain-specific styles. Concretely, given Σ_{org} and Σ_{aug}, we compute the element-wise variance as follows:

$$\begin{aligned} \mu_\Sigma &= \frac{1}{2}(\Sigma_{org} + \Sigma_{aug}) \in \mathbb{R}^{D \times D}, \\ \sigma^2_\Sigma &= \frac{1}{2}((\Sigma_{org} - \mu_\Sigma)^2 + (\Sigma_{aug} - \mu_\Sigma)^2) \in \mathbb{R}^{D \times D}. \end{aligned} \tag{12}$$

We then sort all the elements of σ^2_Σ, and select the top k_c fraction, setting them to 1 to obtain the mask $M \in \mathbb{R}^{D \times D}$:

$$M_{i,j}(k_c) = \begin{cases} 1, & \text{if index}(\sigma^2_\Sigma(i,j)) < \text{len}(\sigma^2_\Sigma) \times k_c \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where the selection ratio $k_c = k_b$ for bonafide utterances and $k_c = k_s$ for spoof ones.

Finally, we adopt this mask to perform PFW, which forces the selected features to be suppressed:

$$\mathcal{L}_{\text{PFW}} = \sum_{k_c \in \{k_b, k_s\}} \sum_{t \in \{org, aug\}} E\left[\|\Sigma_t \odot M(k_c)\|\right], \tag{14}$$

where E is the arithmetic mean.

2.5. Overall training and optimization

Building on the previous learning process, Poincaré prototypes precisely locate instance features. Consequently, Poin-HierNet predicts the final task label by classifying the similarity between sample features z_n and prototypes P, using Binary Cross Entropy (BCE) as the prediction loss:

$$\mathcal{L}_{cls} = \text{BCE}(W \cdot d_H(z_n, P), y_n), \tag{15}$$

where y_n is the label for z_n. The overall training loss is:

$$\mathcal{L}_{all} = \mathcal{L}_{cls} + \mathcal{L}_{PPL} + \mathcal{L}_{HSL} + \mathcal{L}_{PFW}. \tag{16}$$

During the evaluation phase, only the original samples x and $\mathcal{L}_{cls}$ are used.

3. Experiments

3.1. Datasets and metrics

The ASVspoof 2019 LA dataset [27] is utilized for training, comprising a total of 25k utterances, which include both bonafide and spoofed samples. The spoofed samples are generated through six different attack types, including voice conversion and speech synthesis techniques. To assess generalization performance, four test datasets from diverse domains are employed: the ASVspoof 2019 LA evaluation set (19LA) [27], which includes 71k utterances with 13 distinct spoofing attacks; the ASVspoof 2021 LA set (21LA) [7], containing 181k utterances with methods similar to those in 19LA, while also considering encoding and transmission effects from telephony systems; the ASVspoof 2021 DF set (21DF) [7], with over 600k utterances and more than 100 spoofing attacks processed with various lossy codecs; and the In-The-Wild dataset (ITW) [8], which includes 32k utterances collected under real-world, non-controlled conditions, making it a more challenging dataset. Performance is measured with EER.

3.2. Implementation details

All input utterances are randomly chunked into 4-second segments, with the fifth algorithm of the Rawboost [18] applied as the basic augmentation. These samples x are further augmented by adding stationary, signal-independent additive noise [18] to obtain the augmented samples x_{aug}. For the ASVspoof 2021 LA evaluation set, we additionally incorporate noises from the RIR corpus [28], and apply the codec augmentation for better performance. The codec types include adts, mp3, ogg, a-law, and μ-law [7].

The Adam optimizer is utilized, with a learning rate of 1e-6 for the backbone model and 1e-3 for the prototypes. The feature dimension in the Poincaré sphere is set to 160, with a curvature parameter c of 0.01. The number of bonafide prototypes K_b and spoofing prototypes K_s is set to 10 and 6, respectively. For $\mathcal{L}_{HSL}$, we select three nearest neighbors to construct positive prototype pairs. The number of top prototypes K_{top} is set to 256, with the margin δ established at 0.1. For $\mathcal{L}_{PFW}$, the values of k_b and k_s are empirically set at 0.3% and 0.06%,respectively. The batch size B is 256, with the bonafide batch size defined as $B * K_b/K_s$ and the spoofing batch size calculated as $B - (B * K_b/K_s)$. This approach ensures balanced learning of the data distribution during the Poincaré prototype learning process.

3.3. Comparison with existing methods

We evaluate the performance of Poin-HierNet against state-of-the-art (SOTA) methods in terms of the EER. These baselines consist of a variety of network architectures and domain generalization techniques. All methods, including Poin-HierNet, are trained on the ASVspoof2019 LA training set and assessed on four datasets. The results are presented in Table 1.

Table 1: *Comparison of EER (%) performance for different methods across multiple datasets. All systems are trained on the ASVspoof2019 LA training set.*

System	19LA	21LA	21DF	ITW
WavLM+AttM [29]	0.65	3.50	3.19	-
Wav2Vec+LogReg [30]	0.50	-	-	7.20
WavLM+MFA [2]	0.42	5.08	2.56	-
FTDKD [10]	0.30	2.96	2.82	-
OCKD [31]	0.39	<u>0.90</u>	2.27	7.68
GFL-FAD [13]	0.25	-	-	-
WavLM+ASP [14]	0.23	3.31	4.47	-
Wav2Vec+Linear [32]	0.22	3.63	3.65	16.17
OC+ACS [12]	0.17	1.30	<u>2.19</u>	-
Wav2Vec+AASIST [1]	-	**0.82**	2.85	-
Wav2Vec+AASIST2 [33]	<u>0.15</u>	1.61	2.77	-
LSR+LSA [16]	<u>0.15</u>	1.19	2.43	<u>5.92</u>
Poin-HierNet(Ours)	**0.11**	0.94	**1.40**	**4.91**

As presented in Table 1, Poin-HierNet outperforms all existing methods, achieving SOTA results on the 19LA, 21DF, and ITW datasets with EERs of 0.11%, 1.40%, and 4.91%, respectively. Notably, the 21DF and ITW datasets present significant challenges, and the substantial reduction in EER achieved on these datasets further illustrates the robust generalization capabilities of our approach. On the 21LA dataset, it also achieves a competitive EER of 0.94%. Unlike existing baselines that learn data representations in Euclidean space, Poin-HierNet models domain-invariant features in the Poincaré sphere and builds a hierarchical structure upon them. This fundamental difference contributes to the superior performance of Poin-HierNet.

3.4. Ablation study

This section conducts ablation studies to assess the impact of each loss function and the number of prototypes on the performance of the proposed method.

Table 2: *Ablation study on the loss functions across multiple datasets in terms of EER(%). The checked items (✓) indicate the corresponding loss is used for optimization.*

$\mathcal{L}_{PPL}$	$\mathcal{L}_{HSL}$	$\mathcal{L}_{PFW}$	19LA	21LA	21DF	ITW
✓			0.24	1.45	1.95	5.93
✓	✓		0.16	1.30	1.88	5.56
✓	✓	✓	0.11	0.94	1.40	4.91

As shown in Table 2, we present the results to evaluate the effects of different loss functions by progressively adding the three loss functions across multiple datasets. The results demonstrate that using only $\mathcal{L}_{PPL}$ achieves competitive performance, as PPL effectively aligns prototypes with feature distributions in the Poincaré space while minimizing information loss during alignment. After incorporating $\mathcal{L}_{HSL}$, performance improves significantly, as hyperbolic structure learning leverages the advantages of the Poincaré sphere to establish a more meaningful hierarchical feature space. Finally, adding $\mathcal{L}_{PFW}$ yields the best performance on all four datasets, since PFW compels the model to focus on learning domain-invariant features.

Table 3: *Ablation study on the number of bonafide prototypes K_b and spoofing prototypes K_s across multiple datasets in terms of EER(%).*

K_b	K_s	19LA	21LA	21DF	ITW
12	4	0.14	1.55	1.83	5.03
10	6	0.11	0.94	1.40	4.91
8	8	0.17	2.13	2.15	5.65
6	10	0.21	2.96	2.02	5.82
4	12	0.31	2.92	2.44	7.96

Table 3 presents the influence of the values of K_b and K_s on the performance of our method. As observed, the best performance is achieved with $K_b = 10$ and $K_s = 6$. Excessive spoofing prototypes introduce redundancy and degrade performance, while an insufficient number of bonafide prototypes weakens the model's generalization capability. These findings highlight the importance of prototype selection in hyperbolic representation learning for anti-spoofing tasks.

4. Conclusions

In this paper, we propose Poin-HierNet, a novel framework for constructing a domain-invariant hierarchical feature space using the Poincaré ball model. Within Poin-HierNet, PPL introduces a set of data prototypes to represent the overall data distribution; HSL explores the correlations between data prototypes and top prototypes to establish a tree-like hierarchical structure; and PFW applies feature whitening to enhance the model's ability to achieve domain invariance. Experimental results demonstrate the superiority of our framework over existing methods.

5. Acknowledgments

This work was supported by GuangDong Science and Technology Foundation (2023A0505050116), Guangdong Natural Science Foundation (2022A1515011687), the National Natural Science Foundation of China (62371195), and the National Key Research and Development Program of China (2022YFC3301703).

6. References

[1] H. Tak, M. Todisco, X. Wang, J.-w. Jung, J. Yamagishi, and N. Evans, "Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation," in *The Speaker and Language Recognition Workshop*, 2022.

[2] Y. Guo, H. Huang, X. Chen, H. Zhao, and Y. Wang, "Audio deepfake detection with self-supervised wavlm and multi-fusion attentive classifier," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 12 702–12 706.

[3] Q. Zhang, S. Wen, and T. Hu, "Audio deepfake detection with self-supervised xls-r and sls classifier," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 6765–6773.

[4] J.-w. Jung, H.-S. Heo, H. Tak, H.-j. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2022, pp. 6367–6371.

[5] Y. Chen, J. Yi, J. Xue, C. Wang, X. Zhang, S. Dong, S. Zeng, J. Tao, L. Zhao, and C. Fan, "Rawbmamba: End-to-end bidirectional state space model for audio deepfake detection," in *Interspeech*, 2024, pp. 2720–2724.

[6] H. Wu, W. Guo, S. Peng, Z. Li, and J. Zhang, "Adapter learning from pre-trained model for robust spoof speech detection," in *Proc. Interspeech*, 2024, pp. 2095–2099.

[7] X. Liu, X. Wang, M. Sahidullah, J. Patino, H. Delgado, T. Kinnunen, M. Todisco, J. Yamagishi, N. Evans, A. Nautsch *et al.*, "Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2507–2522, 2023.

[8] N. M. Müller, P. Czempin, F. Dieckmann, A. Froghyar, and K. Böttinger, "Does audio deepfake detection generalize?" *arXiv preprint arXiv:2203.16263*, 2022.

[9] Y. Xie, H. Cheng, Y. Wang, and L. Ye, "Domain generalization via aggregation and separation for audio deepfake detection," *IEEE Transactions on Information Forensics and Security*, 2023.

[10] B. Wang, Y. Tang, F. Wei, Z. Ba, and K. Ren, "Ftdkd: Frequency-time domain knowledge distillation for low-quality compressed audio deepfake detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing(ICASSP)*, 2024.

[11] Y. Ren, H. Peng, L. Li, and Y. Yang, "Lightweight voice spoofing detection using improved one-class learning and knowledge distillation," *IEEE Transactions on Multimedia*, 2023.

[12] H. M. Kim, K. Jang, and H. Kim, "One-class learning with adaptive centroid shift for audio deepfake detection," *arXiv preprint arXiv:2406.16716*, 2024.

[13] X. Wang, R. Fu, Z. Wen, Z. Wang, Y. Xie, Y. Liu, J. Tao, X. Liu, Y. Li, X. Qi *et al.*, "Genuine-focused learning using mask autoencoder for generalized fake audio detection," *arXiv preprint arXiv:2406.03247*, 2024.

[14] H. M. Tran, D. Guennec, P. Martin, A. Sini, D. Lolive, A. Delhay, and P.-F. Marteau, "Spoofed speech detection with a focus on speaker embedding," in *Proc. Interspeech*, 2024.

[15] T.-P. Doan, L. Nguyen-Vu, K. Hong, and S. Jung, "Balance, multiple augmentation, and re-synthesis: A triad training strategy for enhanced audio deepfake detection," in *Proc. Interspeech*, 2024, pp. 2105–2109.

[16] W. Huang, Y. Gu, Z. Wang, H. Zhu, and Y. Qian, "Generalizable audio deepfake detection via latent space refinement and augmentation," *arXiv preprint arXiv:2501.14240*, 2025.

[17] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[18] H. Tak, M. Kamble, J. Patino, M. Todisco, and N. Evans, "Rawboost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6382–6386.

[19] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," *Advances in neural information processing systems*, vol. 30, 2017.

[20] S. Kim, B. Jeong, and S. Kwak, "Hier: Metric learning beyond class labels via hierarchical regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 903–19 912.

[21] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. Von Platen, Y. Saraf, J. Pino *et al.*, "Xls-r: Self-supervised cross-lingual speech representation learning at scale," *arXiv preprint arXiv:2111.09296*, 2021.

[22] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Robust classification with convolutional prototype learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3474–3482.

[23] C. Hu, K.-Y. Zhang, T. Yao, S. Ding, and L. Ma, "Rethinking generalizable face anti-spoofing via hierarchical prototype-guided distribution refinement in hyperbolic space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1032–1041.

[24] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," *Advances in neural information processing systems*, vol. 30, 2017.

[25] S. Roy, A. Siarohin, E. Sangineto, S. R. Bulo, N. Sebe, and E. Ricci, "Unsupervised domain adaptation using feature-whitening and consensus loss," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9471–9480.

[26] Q. Zhou, K.-Y. Zhang, T. Yao, X. Lu, R. Yi, S. Ding, and L. Ma, "Instance-aware domain generalization for face anti-spoofing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 453–20 463.

[27] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee *et al.*, "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, p. 101114, 2020.

[28] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[29] Z. Pan, T. Liu, H. B. Sailor, and Q. Wang, "Attentive merging of hidden embeddings from pre-trained speech model for anti-spoofing detection," in *Interspeech*, 2024.

[30] O. Pascu, A. Stan, D. Oneata, E. Oneata, and H. Cucu, "Towards generalisable and calibrated audio deepfake detection with self-supervised representations," in *Interspeech*, 2024, pp. 4828–4832.

[31] J. Lu, Y. Zhang, W. Wang, Z. Shang, and P. Zhang, "One-class knowledge distillation for spoofing speech detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 11 251–11 255.

[32] X. Wang and J. Yamagishi, "Spoofed training data for speech spoofing countermeasure can be efficiently created using neural vocoders," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[33] Y. Zhang, J. Lu, Z. Shang, W. Wang, and P. Zhang, "Improving short utterance anti-spoofing with aasist2," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 11 636–11 640.