Interspeech 2024
1-5 September 2024, Kos, Greece

DGPN: A Dual Graph Prototypical Network for Few-Shot Speech Spoofing Algorithm Recognition

Zirui Ge[1], *Xinzhou Xu*[1], *Haiyan Guo*[1], *Tingting Wang*[1], *Zhen Yang**[1], *Björn W. Schuller*[2,3]

[1] School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, P. R. China
[2] CHI – Chair of Health Informatics, MRI, Technische Universität München, Germany
[3] GLAM – Group on Language, Audio, & Music, Imperial College London, U. K.

{2022010211, xinzhou.xu, guohy, tingting_wang, yangz}@njupt.edu.cn; schuller@ieee.org

Abstract

As synthetic speech technologies rapidly advance, accurately classifying these synthesis algorithms has become increasingly critical in the speech anti-spoofing. Nevertheless, in the incipient stage of emerging spoofing algorithms, the acquisition of ample generated speech samples is often constrained, impeding the efficacy of conventional models. To this end, we introduce a novel methodology within the realm of few-shot learning, named Dual Graph Prototypical Network (DGPN), in view of this limitation for the Speech Spoofing Algorithm Recognition (SSAR) task. The proposed method consists of intra-speech graph and inter-speech graph modules, where the former employs graph attention networks to model the low-level representations of an utterance, and the latter utilizes graph neural networks to depict high-level representations of different utterances. Experimental evaluations demonstrate that the proposed method outperforms existing models in classification accuracy, showcasing its effectiveness in addressing the challenge of the few-shot SSAR task.

Index Terms: Few-shot learning, speech spoofing algorithm recognition, graph neural networks, speech anti-spoofing.

1. Introduction

As a pivotal component in a biometric information confirmation system, *Automatic Speaker Verification* (ASV) technology aims to distinguish among different speakers through spoken signals [1]. However, the rapid advancements in *Text-To-Speech* (TTS) synthesis and *Voice Conversion* (VC) techniques [2, 3] have introduced security challenges in ASV systems, potentially threatening the privacy and security of speakers [4]. In response to these challenges, *Audio DeepFake Detection* (ADD) systems have been proposed to recognize fake speech [5]. Meanwhile, beyond focusing on the binary classification for true and fake speech, distinguishing the source of fake speech can not only enhance the performance of an ADD system [6, 7], but be helpful for forensics of malicious speech [8]. Consequently, the emerging task of *Speech Spoofing Algorithm Recognition* (SSAR) has garnered attention, aiming to trace the sources of spoofing speech [8, 9] .

The purpose of the SSAR task is to identify the categories of spoofing algorithms, under the condition of giving some arbitrary samples generated by a certain spoofing algorithm. Initially, the Audio Deepfake Detection Challenge 2023 [9] proposed an audio deepfake algorithm recognition task as its "track 3" sub-challenge. Further, [10] proposes a center-based similarity maximum method for determining the categories of spoofing algorithms, while, [11, 12] take the representations of pre-trained models as the speech features for the SSAR task. Afterwards, [13, 14, 15] fused the embedding features extracted by different models for SSAR tasks.

Nevertheless, existing methodologies still include two deficiencies for addressing the SSAR task. First, acquiring extensive utterance samples of the emerging spoofing algorithms poses a significant challenge, and conventional models may be overfitting to the extremely small size labeled examples. Second, although *Few-Shot Learning* (FSL) methods [16, 17] are proposed for this data scarcity problem, conventional FSL models [18, 19] often fall short by treating all examples as uniformly contributive to the class prototype. This may lead to overlooking the diverse information of different labeled examples and hindering the models' performance. In this regard, we introduce the *Dual Graph Prototypical Network* (DGPN) for these two deficiencies in the SSAR task. For the data scarcity problem, the proposed method, residing in the field of FSL, can efficiently learn the new classes given few labeled examples. To address the second problem, we consider the different importance of low-level representations of an utterance in characterizing the spoofing algorithms, and high-level representations of different utterances in forming the class prototypes.

The proposed approach first utilizes wav2vec 2.0 [20] to acquire low-level representations of a spoofing utterance. Then, we design an intra-speech graph module, containing a *Graph ATtention network* (GAT) [21] to model the obtained low-level components, simultaneously generating high-level utterance representations. Further, we introduce an inter-speech graph module to depict the relationship of high-level utterance representations, employing a *Graph Neural Network* (GNN) to form the graph class prototypes.

In addition, we make a comparison between the proposed method and highly-related FSL works. In contrast to conventional FSL methods [18, 19, 22], our method assigns learnable weights for utterance representations. Compared with the other GNN-based methods [23, 24], we jointly consider the low-level and high-level components of different samples. We also observe that [25] consists of feature-level and instance-level attention blocks, while our approach models different level representations as graph signals, and outperforms the approach proposed in [25] on the SSAR task.

2. METHODOLOGY

The proposed DGPN mainly contains a speech representation extracting module, intra-speech graph module, and inter-speech graph module, as presented in Figure 1. Within the proposed approach, the speech representation extracting module aims to generate low-level representations of different utterances. Then,

* Corresponding author(s)

10.21437/Interspeech.2024-1724

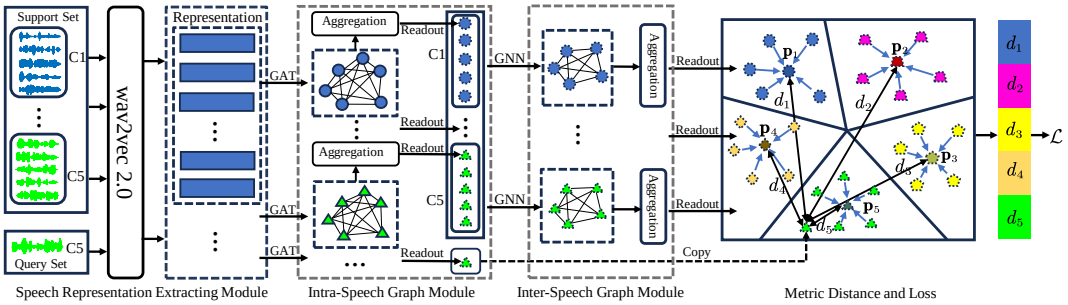

Figure 1: *The diagrammatic overview of the proposed DGPN approach, exemplified by a* 5*-way* 5*-shot task with* 1 *query sample in class C5, where C1, C2,* $\cdots$*, C5 represent the* 5 *classes, respectively, and* $d_1, d_2, \cdots, d_5$ *indicate the query-to-prototype distances.*

the intra-speech graph module models these low-level representations, through the first GNN to obtain the high-level utterance representations. Further, the second GNN utilizes the utterance representations within the class n to acquire the class prototype $\mathbf{p}_n$. Finally, the distance d_n between query sample and class prototypes $\mathbf{p}_n$ is fed into the loss function $\mathcal{L}$.

2.1. Preliminary

Let $\mathcal{D}_{base}$ and $\mathcal{D}_{novel}$ denote FSL training and test example sets, respectively. In the FSL framework, the FSL model learns distinguishing the novel classes with a few samples of $\mathcal{D}_{novel}$ in the test phase based on knowledge learned from $\mathcal{D}_{base}$ in the training phase. The FSL training task $\mathcal{T}$ typically takes the form of an N-way K-shot classification task, and each task $\mathcal{T}$ consists of the labeled support set $\mathcal{D}_s = \{(\mathbf{h}_i, y_i)\}_{i=1}^{NK}$ and unlabeled query set $\mathcal{D}_q = \{\mathbf{h}_i\}_{i=1}^{NM}$. Particularly, the support set $\mathcal{D}_s$ contains N classes with K samples per class, and the query set $\mathcal{D}_q$ contains samples from the same N classes with M samples per class, where $\mathbf{h}$ is the sample representation, and y is the sample label.

2.2. Intra-Speech Graph Module

The wav2vec 2.0 model[1] is employed as the upstream feature extracting module. For the kth speech sample in the nth class of task $\mathcal{T}$, the size of the extracted output representation $\mathbf{E}_{n,k}$ is (L, T, F), where $L = 13$ is the number of the hidden outputs in the wav2vec 2.0 model (including 12 Transformer layers), and F is the features' dimensionality for each frame (with a duration of 20 ms). With T representing the number of speech frames within the sample, the sample's feature matrix can be set to

$$\mathbf{R}_{n,k} = (\sum_{l=1}^{L} \alpha_l)^{-1} \sum_{l=1}^{L} \alpha_l \mathbf{E}_{n,k}(l, \cdot, \cdot) \tag{1}$$

through combining the representations $\mathbf{E}_{n,k}(l, \cdot, \cdot)$s ($l = 1, 2, \cdots, L$) corresponding to the L layers, respectively, where α_ls represent trainable weights initialized as 1, and the size of $\mathbf{R}_{n,k}$ is (T, F).

The intra-speech graph module models speech frames as a complete graph, where the nodes are associated with speech frames in each speech, and the edges are given by the graph attention coefficients. We first map the obtained frame features into a space dominated by task-related information via linear transformation, written as

$$\mathbf{r}_{n,k}^i = \mathbf{W}\mathbf{R}_{n,k}^\top(i, \cdot) + \mathbf{o}, \tag{2}$$

where $i = 1, 2, \cdots, T$, and $\mathbf{W} \in \mathbb{R}^{F' \times F}$ is the projection matrix, with an F'-dimensional offset $\mathbf{o}$. Then, the ith-row and jth-column element ($i, j = 1, 2, \cdots, T$) of the frame-level adjacency matrix $\mathbf{A}_{n,k}^{intra}$ is represented as

$$(\mathbf{A}_{n,k}^{intra})_{i,j} = \frac{e^{\beta \cos\left(\mathbf{r}_{n,k}^i, \mathbf{r}_{n,k}^j\right)}}{\sum_{v_m \in \mathcal{N}_1(i)} e^{\beta \cos\left(\mathbf{r}_{n,k}^i, \mathbf{r}_{n,k}^{v_m}\right)}}, \tag{3}$$

where β is a learnable parameter. $\mathcal{N}_1(i)$ represents the node-index set containing the indexes of the ith node and its neighboring nodes, and v_m indicates the index of the mth node within the set. Then, we obtain the aggregation for the ith node as

$$\tilde{\mathbf{r}}_{n,k}^i = \sigma\left((\mathbf{A}_{n,k}^{intra})_{i,i}\mathbf{r}_{n,k}^i + \sum_{j \in \mathcal{N}_1(i), j \neq i} (\mathbf{A}_{n,k}^{intra})_{i,j}\mathbf{r}_{n,k}^j\right), \tag{4}$$

where $\sigma(\cdot)$ is the ReLU activation function, leading to the speech-level representation written as

$$\mathbf{h}_{n,k} = Readout(\{\tilde{\mathbf{r}}_{n,k}^1, \tilde{\mathbf{r}}_{n,k}^2, \cdots, \tilde{\mathbf{r}}_{n,k}^T\}), \tag{5}$$

where $Readout(\cdot)$ can be a simple permutation invariant function. Therefore, speech sample embeddings in the nth class can be represented as $\mathcal{H}_n = \{\mathbf{h}_{n,1}, \mathbf{h}_{n,2}, \cdots, \mathbf{h}_{n,K}\}$.

2.3. Inter-Speech Graph Module

We model all labeled examples in a class as graph nodes residing on a graph. FSL aims to learn the embedding of the entire graph given a few sampled graph nodes, and then determine whether a new node resides on this graph or not.

We employ a *Multi-Layer Perceptron* (MLP) to calculate the similarity of a pair of samples in the nth class, resulting in the k_1th-row and k_2th-column element ($k_1, k_2 = 1, 2, \cdots, K$) of the learned similarity matrix $\mathbf{S}_n$ expressed as

$$(\mathbf{S}_n)_{k_1,k_2} = \psi_\theta(|\mathbf{h}_{n,k_1} - \mathbf{h}_{n,k_2}|), \tag{6}$$

where $\psi_\theta(\cdot)$ is the MLP parameterized by θ, and the absolute difference of the two node features leads to a symmetric matrix. Then, the utterance-level weighted adjacency matrix is represented as $\mathbf{A}_n^{inter} = Softmax(\mathbf{S}_n)$, using a softmax mapping.

Finally, the same aggregation function as in Section 2.2 is employed on $\mathbf{A}_n^{inter}$ and $\mathcal{H}_n$, represented as

$$\tilde{\mathbf{h}}_{n,k} = \sigma\left((\mathbf{A}_n^{inter})_{k,k}\mathbf{h}_{n,k} + \sum_{k' \in \mathcal{N}_2(k), k' \neq k} (\mathbf{A}_n^{inter})_{k,k'}\mathbf{h}_{n,k'}\right), \tag{7}$$

in order to obtain the aggregated node $\tilde{\mathbf{h}}_{n,k}$, using the corresponding elements $(\mathbf{A}_n^{inter})_{\cdot,\cdot}$ from $\mathbf{A}_n^{inter}$ within the neighboring node-index set $\mathcal{N}_2(k)$ including the kth node itself.

We further acquire the nth-class graph prototypical representation using the readout function as $\mathbf{p}_n = Readout(\{\tilde{\mathbf{h}}_{n,1}, \tilde{\mathbf{h}}_{n,2}, \cdots, \tilde{\mathbf{h}}_{n,K}\})$.

[1] https://huggingface.co/facebook/wav2vec2-base

Table 1: *Summary of the ASVSpoof 2019 LA training and evaluation sets. Bonafide represents the natural speech.*

Description \ Sets	Training Set	Evaluation Set
Class Description (Bonafide & Algorithms)	Bonafide & A01∼A06	Bonafide & A07∼A19
# Utterances	25 380	71 237
Total Duration	24.2 h	61.5h
Sampling Rate	16 kHz	16 kHz

2.4. Loss Function

The label of an arbitrary query $\mathbf{q}$ can be predicted through computing the metric distance between the query and the class prototypical representation of each class as

$$\hat{y} = \arg\max_{n} \; p(y = n \mid \mathbf{q}) = \frac{e^{-d(\mathbf{q},\mathbf{p}_n)}}{\sum_{n'=1}^{N} e^{-d\left(\mathbf{q},\mathbf{p}_{n'}\right)}}, \qquad (8)$$

where $d(\cdot)$ represents the square of Euclidean distance.

The objective of each FSL training task $\mathcal{T}$ is to minimize the classification loss, which is determined by the discrepancy between the predicted and ground-truth labels in the query set. Hence, the training loss is formulated as the average negative log-likelihood probability of assigning correct class labels as

$$\mathcal{L} = -\frac{1}{NM} \sum_{c=1}^{NM} \log p\left(y_c^* \mid \mathbf{q}_c\right), \qquad (9)$$

where $\mathbf{q}_c$ $(c = 1, 2, \cdots, NM)$ indicates the cth query in the query set when training, and y_c^* is the ground-truth label of $\mathbf{q}_c$.

3. Experimental Setups

3.1. The Datasets

In order to evaluate the performance, we implement the experiments on the ASVspoof 2019 *Logical Access* (LA) database [5]. In the LA database, the training and development sets share the same 6 speech spoofing algorithms (A01∼A06). In the evaluation set, there are 13 speech spoofing algorithms (A07∼A19), where A16 takes the same algorithm as A04, and A19 takes the same algorithm as A06. As presented in [5], upon the visualized speech feature distributions of A16 and A04, as well as A19 and A06, A16 and A04 exhibit a significant overlap in their feature spaces, whereas A19 and A06 demonstrate a more modest shared feature space. In light of this observation, we exclude algorithm A16 from the evaluation set in the subsequent experiments to avoid a serious data leakage. In this paper, we utilize the training and evaluation sets to implement our experiments, and Table 1 shows the details of these two subsets.

3.2. Implementation Details

In each experiment, we first denote the training set as the $\mathcal{D}_{base}$ and the evaluation set as the $\mathcal{D}_{novel}$ set, and name this pipeline as 'Task I', while exchanging the two sets to form a second task denoted as 'Task II'. We follow the setting in Section 2.1 to train the model in 5-way with 5-shot, 5-way with 10-shot, 5-way with 20-shot classification, respectively, and $M = 1$ query for both of the training tasks.

The weight parameters of the pre-trained models are frozen in the FSL training phase. In detail, the output feature dimensionality is $F = 768$ in the speech feature extracting module and the transformed dimensionality is $F' = 512$ in the intra-graph module. For the test phase, we report the mean of 6 000 randomly generated test tasks in terms of accuracy (%).

We set each sample's duration to 2 seconds sampling from the audio files and the output speech feature size from wav2vec 2.0 is $(13, 99, 768)$. Further, we take the mean over every 11 continuous frames along with the time dimension, and the output feature size is $(13, 9, 768)$. For the neighboring sets $\mathcal{N}_1(\cdot)$ and $\mathcal{N}_2(\cdot)$, we set them to all the nodes, resulting in complete graphs for the GNNs. The five-layer MLP is used and the hidden unit size is $(192, 192, 96, 48, 1)$ in Section 2.3. The $Readout(\cdot)$ is set to the mean function in Section 2.2 and 2.3. The batch size is set to 30. The optimizer is set to *Adaptive moment estimation* (Adam) with a learning rate 10^{-4}, and each experiment implements 20 epochs.

Table 2: *Comparison with the baseline methods in terms of average classification accuracy with standard variation (%), where N_1 and N_2 are the number of classes for different $\mathcal{D}_{novel}$.*

Approaches	# Ways in Test	5-Shot	10-Shot	20-Shot
Task I:				
ProtoNet [18]	5-Way	78.8 ± 2.4	81.5 ± 2.5	81.3 ± 2.5
	N_1-Way	59.9 ± 0.9	62.7 ± 0.6	62.3 ± 0.7
MatchingNet [22]	5-Way	78.8 ± 2.5	78.8 ± 2.2	78.6 ± 2.2
	N_1-Way	58.7 ± 1.1	58.9 ± 0.8	59.2 ± 0.6
RelationNet [19]	5-Way	74.7 ± 1.2	74.5 ± 0.9	75.3 ± 1.0
	N_1-Way	50.2 ± 0.6	50.4 ± 0.5	49.9 ± 0.6
Attention Based [25]	5-Way	79.6 ± 0.6	80.5 ± 0.8	80.7 ± 1.1
	N_1-Way	60.7 ± 0.9	61.7 ± 0.9	61.6 ± 1.0
DGPN (Proposed)	5-Way	$\mathbf{83.8} \pm 1.0$	$\mathbf{85.4} \pm 1.4$	$\mathbf{85.4} \pm 1.3$
	N_1-Way	$\mathbf{65.9} \pm 1.3$	$\mathbf{66.4} \pm 1.2$	$\mathbf{66.8} \pm 1.2$
Task II:				
ProtoNet [18]	5-Way	95.2 ± 1.1	95.5 ± 0.7	95.0 ± 0.8
	N_2-Way	93.1 ± 0.8	94.0 ± 0.8	93.4 ± 0.7
MatchingNet [22]	5-Way	94.8 ± 1.8	94.1 ± 2.3	94.4 ± 1.4
	N_2-Way	92.4 ± 1.3	91.7 ± 1.1	92.2 ± 0.9
RelationNet [19]	5-Way	93.5 ± 1.2	93.8 ± 1.3	93.8 ± 1.2
	N_2-Way	91.1 ± 1.4	91.2 ± 1.4	91.6 ± 1.1
Attention Based [25]	5-Way	94.8 ± 1.1	96.1 ± 1.2	95.5 ± 1.1
	N_2-Way	92.6 ± 1.5	$\mathbf{95.1} \pm 1.1$	94.1 ± 1.3
DGPN (Proposed)	5-Way	$\mathbf{96.6} \pm 0.9$	$\mathbf{96.9} \pm 0.9$	$\mathbf{97.0} \pm 0.8$
	N_2-Way	$\mathbf{94.9} \pm 1.1$	95.0 ± 0.9	$\mathbf{95.3} \pm 0.7$

4. Experimental Results

4.1. Comparison Results with Different Methods

In the experiments, we aim to make comparisons between the proposed DGPN and existing approaches. The compared metric-based approaches include ProtoNet [18], MatchingNet [22], RelationNet [19], and the attention-based method (noted as 'Attention Based') [25]. Note that these approaches keep the same setup for processing the L-layer outputs in the wav2vec 2.0 model, afterwards using a two-layer *Gated Recurrent Unit* (GRU) based *Recurrent Neural Network* (RNN) with the hidden size 128 to perform temporal encoding. We also consider a full training model with a joint optimization strategy [26] in comparison, which employs RawNet2 [27] as the speech encoder.

The results, summarized in Table 2 and Figure 2, present a comprehensive comparison of various methods for the few-shot SSAR task. In Table 2, the test task is constructed in two styles. The first style is sampling randomly 5 classes in $\mathcal{D}_{novel}$ to construct the test task (the first row in each method), named the 5-way style, which is the same as that in the training phase, and the second style is taking all the classes in $\mathcal{D}_{novel}$ to construct

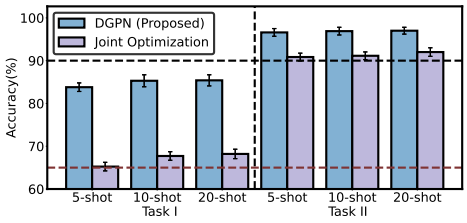

Figure 2: *Comparison between the proposed DGPN and the joint optimization strategy [26] in terms of average classification accuracy and standard variation (%), with the 5-way test.*

Table 3: *Performance difference in terms of average classification accuracy (%) of the results in ResNet-18 as the pre-training model and wav2vec 2.0 as the pre-training model, i. e., the accuracy of wav2vec 2.0 subtracts that of ResNet-18.*

Approaches	5-Shot	10-Shot	20-Shot
Task I:			
ProtoNet [18]	+26.4	+29.3	+26.7
MatchingNet [22]	+24.7	+24.8	+24.0
RealtionNet [19]	+25.1	+24.3	+26.2
Attention Based [25]	+27.8	+27.8	+28.6
DGPN (Proposed)	**+32.2**	**+32.7**	**+32.9**
Task II:			
ProtoNet [18]	+40.7	+40.2	+40.3
MatchingNet [22]	+40.2	+36.2	+38.8
RealtionNet [19]	+34.5	+36.7	+34.8
Attention Based [25]	+41.8	+41.5	+40.5
DGPN (Proposed)	**+42.8**	**+42.0**	**+41.4**

the test task (the second row in each method), named the all-way style. Specifically, in the all-way style, Task I employs $N_1 = 13$ classes (excluding A16, but including Bonafide) to construct a test task, and Task II contains $N_2 = 7$ (including Bonafide) classes. Through Table 2 and Figure 2, we observe that our proposed DGPN outperforms the baseline approaches. The joint optimization method [26] fails to achieve the best performance among these approaches. This is possibly because this method is mainly used for the binary classification ADD task, with weak adaptation for multi-class classification. Besides, Table 2 shows that using all the classes to construct a test task will lead to decreasing performance. This declined performance may be primarily attributed to the discrepancy in the number of classes between the test and training tasks, from the perspective of an episodic training manner [28, 29].

The enhanced performance in Task II, as detailed in Table 2 and Figure 2, underscores the advantage of having a larger number of classes in $\mathcal{D}_{base}$ (comprising 7 and 13 classes in Task I and Task II, respectively) for effectively distinguishing novel classes, consistent with [30]. However, increasing the number of shots in the FSL tasks fails consistently leading to a corresponding enhancement in model performance, which may attribute to the pre-training models, whose parameters remain static during the training phase. Hence, the sample embedding plays a crucial role in the success of metric-based FSL methods [31, 32].

4.2. Ablation Studies

As in existing ADD tasks, the compared feature extracting module is the pre-trained ResNet-18 [33] on the 60-dimensional linear frequency cepstral coefficients [34]. As presented in Table 3, it is seen that the performance on the output features of the pre-trained ResNet-18 is not comparable to that on wav2vec 2.0. This means that the output features of the pre-trained ResNet-18 may mainly contain the discrimination information between the

Table 4: *Ablation results in terms of average accuracy with standard variation (%) for DGPN in Task I, with the 5-way test.*

Intra-Speech Module	Inter-Speech Module	5-Shot	10-Shot	20-Shot
×	×	78.8 ± 2.4	81.5 ± 2.5	81.3 ± 2.5
√	×	81.4 ± 1.7	82.4 ± 1.9	82.1 ± 1.8
×	√	82.0 ± 0.8	82.5 ± 1.1	82.9 ± 1.0
√	√	$\mathbf{83.8} \pm 1.0$	$\mathbf{85.4} \pm 1.4$	$\mathbf{85.4} \pm 1.3$

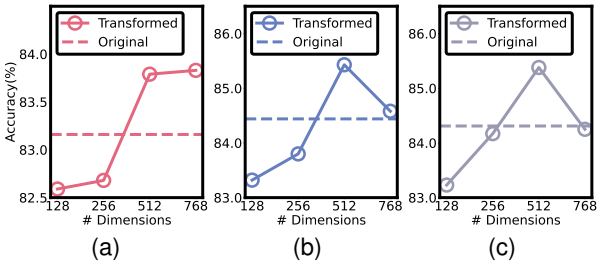

Figure 3: *Ablation results in terms of average classification accuracy (%) when using a different transformed dimension F' for Task I, with the 5-way setup for test.*

bonafide and spoofing speech, and ignore the category information within the spoofing speech.

We further present the results in Table 4 through removing inter-speech or intra-speech graph modules. They indicate an improvement in classification accuracy upon employing the intra-speech module or inter-speech graph module compared to the original ProtoNet model. Furthermore, the integration of both intra-speech and inter-speech graph modules yields the highest performance. These results underscore the contribution of each module in enhancing the overall performance.

Finally, we investigate the influence of a different transformed dimension F' on the experimental results in Figure 3. We observe that different values of F' can lead to distinct performances. Specifically, $F' = 512$ achieves the most favorable results, whereas smaller values of F' result in inferior performance compared to the original features. The performance with $F' = 768$ is nearly comparable to that with the original features. This suggests that smaller F' values may lead to the loss of critical information pertinent to spoofing algorithms, while large F' values may fail to focus on task-relevant information.

5. Conclusion

We proposed a *Dual Graph Prototypical Network* (DGPN) for a few-shot speech spoofing algorithm recognition in this work, which comprises intra-speech and inter-speech modules, focusing on the relationships among low-level and high-level representations of different utterances. The experiments on the ASVspoof 2019 LA dataset show that our method improves classification accuracy over the baselines. In future work, one may research domain generalization for the speech spoofing algorithm classification task, in view of the domain gaps between speech spoofing algorithms in real-world cases.

6. Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant 62071242), the China Postdoctoral Science Foundation (Grant 2022M711693), the Postgraduate Research and Practice Innovation Program of Jiangsu Province (KYCX23_1034), and the DFG Project AUDI0NOMOUS (Grant 442218748).

7. References

[1] Z. Bai and X.-L. Zhang, "Speaker Recognition Based on Deep Learning: An Overview," *Neural Networks*, vol. 140, pp. 65–99, 2021.

[2] B. Sisman, J. Yamagishi, S. King, and H. Li, "An Overview of Voice Conversion and Its Challenges: From Statistical Modeling to Deep Learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 132–157, 2020.

[3] A. Triantafyllopoulos, B. W. Schuller, and *et al.*, "An Overview of Affective Speech Synthesis and Conversion in the Deep Learning Era," *Proceedings of the IEEE*, pp. 1355–1381, 2023.

[4] A. Khan, K. M. Malik, J. Ryan, and M. Saravanan, "Battling Voice Spoofing: A Review, Comparative Analysis, and Generalizability Evaluation of State-of-the-art Voice Spoofing Counter Measures," *Artificial Intelligence Review*, pp. 1–54, 2023.

[5] X. Wang, J. Yamagishi, and *et al.*, "ASVspoof 2019: A Large-scale Public Database of Synthesized, Converted and Replayed Speech," *Computer Speech & Language*, vol. 64, p. 101114, 2020.

[6] K. Ma, Y. Feng, B. Chen, and G. Zhao, "End-to-End Dual-Branch Network Towards Synthetic Speech Detection," *IEEE Signal Processing Letters*, vol. 30, pp. 359–363, 2023.

[7] R. Li, M. Zhao, Z. Li, L. Li, and Q. Hong, "Anti-Spoofing Speaker Verification System with Multi-Feature Integration and Multi-Task Learning," in *Proc. International Speech Communication Association (INTERSPEECH)*, Graz, Austria, 2019, pp. 1048–1052.

[8] X. Yan, J. Yi, J. Tao, C. Wang, H. Ma, Z. Tian, and R. Fu, "System fingerprints detection for deepfake audio: An initial dataset and investigation," vol. abs/2208.10489, 2022.

[9] J. Yi, J. Tao, and *et al.*, "ADD 2023: the Second Audio Deepfake Detection Challenge," *CoRR*, vol. abs/2305.13774, 2023.

[10] X. Qin, X. Wang, Y. Chen, Q. Meng, and M. Li, "From Speaker Verification to Deepfake Algorithm Recognition: Our Learned Lessons from ADD2023 Track3," in *Proc. IJCAI Workshop on Deepfake Audio Detection and Analysis*, Macao, China, 2023, pp. 107–112.

[11] Z. Wang, Q. Wang, J. Yao, and L. Xie, "The NPU-ASLP System for Deepfake Algorithm Recognition in ADD 2023 Challenge," in *Proc. IJCAI Workshop on Deepfake Audio Detection and Analysis*, Macao, China, 2023, pp. 64–69.

[12] X.-M. Zeng, J.-T. Zhang, K. Li, Z.-L. Liu, W.-L. Xie, and Y. Song, "Deepfake Algorithm Recognition System with Augmented Data for ADD 2023 Challenge," in *Proc. IJCAI Workshop on Deepfake Audio Detection and Analysis*, Macao, China, 2023, pp. 31–36.

[13] S. Han, T. Kang, and *et al.*, "CAU KU Deep Fake Detection System for ADD 2023 Challenge," in *Proc. IJCAI Workshop on Deepfake Audio Detection and Analysis*, Macao, China, 2023, pp. 23–30.

[14] Y. Tian, Y. Chen, Y. Tang, and B. Fu, "Deepfake Algorithm Recognition Through Multi-model Fusion Based on Manifold Measure," in *Proc. IJCAI Workshop on Deepfake Audio Detection and Analysis*, Macao, China, 2023, pp. 76–81.

[15] J. Lu, Y. Zhang, Z. Li, Z. Shang, W. Wang, and P. Zhang, "Detecting Unknown Speech Spoofing Algorithms with Nearest Neighbors," in *Proc. IJCAI Workshop on Deepfake Audio Detection and Analysis*, Macao, China, 2023, pp. 89–94.

[16] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *Proc. International Conference on Learning Representations (ICLR)*, New Orleans, LA, 2019, p. no pagination.

[17] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, "A Comprehensive Survey of Few-Shot Learning: Evolution, Applications, Challenges, and Opportunities," *ACM Computing Surveys*, vol. 55, no. 13s, 2023.

[18] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. International Conference on Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, 2017, p. 4080–4090.

[19] F. Sung, Y. Yang, and *et al.*, "Learning to Compare: Relation Network for Few-Shot Learning," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, 2018, pp. 1199–1208.

[20] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in *Proc. International Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, 2020, p. 12449–12460.

[21] K. K. Thekumparampil, C. Wang, S. Oh, and L.-J. Li, "Attention-based Graph Neural Network for Semi-supervised Learning," *ArXiv*, vol. abs/1803.03735, 2018.

[22] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching Networks for One Shot Learning," in *Proc. International Conference on Neural Information Processing Systems (NeurIPS)*, Barcelona, Spain, 2016, p. 3637–3645.

[23] K. Ding, J. Wang, J. Li, K. Shu, C. Liu, and H. Liu, "Graph prototypical networks for few-shot learning on attributed networks," in *Proc. ACM International Conference on Information & Knowledge Management (CIKM)*, Virtual Event, 2020, p. 295–304.

[24] S. Zhang, Y. Qin, K. Sun, and Y. Lin, "Few-Shot Audio Classification with Attentional Graph Neural Networks," in *Proc. International Speech Communication Association (INTERSPEECH)*, Graz, Austria, 2019, pp. 3649–3653.

[25] Y. Wang and D. V. Anderson, "Hybrid Attention-Based Prototypical Networks for Few-Shot Sound Classification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, 2022, pp. 651–655.

[26] Z. Wang and J. H. Hansen, "Audio Anti-spoofing Using Simple Attention Module and Joint Optimization Based on Additive Angular Margin Loss and Meta-learning," in *Proc. International Speech Communication Association (INTERSPEECH)*, Incheon, Korea, 2022, pp. 376–380.

[27] J. Jung, S. Kim, H. Shim, J. Kim, and H. Yu, "Improved RawNet with Feature Map Scaling for Text-Independent Speaker Verification Using Raw Waveforms," in *Proc. International Speech Communication Association (INTERSPEECH)*, Virtual Event, 2020, pp. 1496–1500.

[28] E. Triantafillou, T. Zhu, and *et al.*, "Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples," in *Proc. International Conference on Learning Representations (ICLR)*, Virtual Event, 2020, p. no pagination.

[29] C.-C. Lin, H.-L. Chu, Y.-C. F. Wang, and C.-L. Lei, "Joint Feature Disentanglement and Hallucination for Few-Shot Image Classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 9245–9258, 2021.

[30] X. Luo, H. Wu, J. Zhang, L. Gao, J. Xu, and J. Song, "A Closer Look at Few-Shot Classification Again," in *Proc. International Conference on Machine Learning (ICML)*, Honolulu, HI, 2023, pp. 23 103–23 123.

[31] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking Few-Shot Image Classification: A Good Embedding is All You Need?" in *Proc. European Conference on Computer Vision (ECCV)*, Virtual Event, 2020, pp. 266–282.

[32] Z. Yang, J. Wang, and Y. Zhu, "Few-Shot Classification with Contrastive Learning," in *Proc. European Conference on Computer Vision (ECCV)*, Tel Aviv, Israel, 2022, p. 293–309.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 770–778.

[34] Y. Zhang, F. Jiang, and Z. Duan, "One-Class Learning Towards Synthetic Voice Spoofing Detection," *IEEE Signal Processing Letters*, vol. 28, pp. 937–941, 2021.