Graph Attention Networks for Anti-Spoofing

Hemlata Tak[1], Jee-weon Jung[2], Jose Patino[1], Massimiliano Todisco[1] and Nicholas Evans[1]

[1]EURECOM, Sophia Antipolis, France
[2]Naver Corporation, South Korea

`lastname@eurecom.fr`, `jeeweon.jung@navercorp.com`

Abstract

The cues needed to detect spoofing attacks against automatic speaker verification are often located in specific spectral sub-bands or temporal segments. Previous works show the potential to learn these using either spectral or temporal self-attention mechanisms but not the relationships between neighbouring sub-bands or segments. This paper reports our use of graph attention networks (GATs) to model these relationships and to improve spoofing detection performance. GATs leverage a self-attention mechanism over graph structured data to model the data manifold and the relationships between nodes. Our graph is constructed from representations produced by a ResNet. Nodes in the graph represent information either in specific sub-bands or temporal segments. Experiments performed on the ASVspoof 2019 logical access database show that our GAT-based model with temporal attention outperforms all of our baseline single systems. Furthermore, GAT-based systems are complementary to a set of existing systems. The fusion of GAT-based models with more conventional countermeasures delivers a 47% relative improvement in performance compared to the best performing single GAT system.

Index Terms: graph attention network, graph neural network, anti-spoofing, automatic speaker verification, ASVspoof

1. Introduction

The success of anti-spoofing solutions for automatic speaker verification (ASV) systems is dependent on the reliable identification of processing artefacts stemming from the manipulation or synthesis of speech signals [1]. These artefacts are known to reside within specific sub-bands or temporal segments [2–7]. Their detection hence calls for spoofing countermeasure systems with spectral and/or temporal attention.

Convolutional neural network (CNN) approaches have been applied extensively to the anti-spoofing problem. CNNs are particularly appealing because of their capacity to extract localised artefacts within spectro-temporal decompositions such as a spectrogram. For both the ASVspoof 2017 [8] and ASVspoof 2019 [9–11] challenges, CNN-based approaches were among the best performing systems. More elaborate systems, such as those based upon ResNet architectures, are now attracting greater interest, and enable the learning of deeper networks using residual blocks with skip connections [12–19].

Our own work [20] explored the use of a RawNet2 architecture [21]. Similar to the original RawNet architecture [22], RawNet2 adopts residual blocks with skip connections and feeds their output to a gated recurrent unit layer to extract utterance level representations. The first convolutional layer uses a bank of band-pass filters parametrised in the form of sinc functions in identical fashion to SincNet [23]. Other differences include the use of cosine similarity scoring and the application of feature map scaling (FMS) to residual block outputs. FMS acts to emphasise the most salient sinc filter outputs (or sub-bands) by applying different weights to aggregated features extracted from each band. The network can hence be optimised to apply greater attention to the most discriminant sub-bands. FMS can be interpreted as a form of spectral attention and is applied to residual block outputs which is an aggregated frequency response. However, it neither learns nor models the relationships between different filters or sub-bands (e.g. spoofing artefacts present simultaneously in two different sub-bands). Like CNNs, ResNet and RawNet-based architectures also lack the capacity to capture and use such information to help discriminate between bona fide and spoofed speech. Having observed that spoofing artefacts can be present across multiple sub-bands or temporal segments, our hypothesis is that an attention mechanism with the power to model the relationships between different them has potential to improve upon anti-spoofing performance.

Graph neural networks (GNNs) [24, 25], especially recent architectures such as graph convolution networks (GCNs) [26] or graph attention networks (GATs) [27] can be used to model these relationships. Instead of modelling frames or sub-band representations linearly, GNNs models the non-Euclidean data manifold spanning different sub-bands and temporal segments. While conventional attention mechanisms can be used in spatial or sequence-based tasks to focus on more relevant information, graph attention mechanisms have the capacity to learn which sub-bands or segments are the most informative with regard to their neighbours, and to assign weights to emphasise those that are the most discriminative. Accordingly, we have explored the use of GNN-based architectures to model spectral and temporal relationships for the spoofing detection task.

The remainder of the paper is organised as follows. Section 2 describes the related work, Section 3 introduces the proposed GAT approach to anti-spoofing. The experimental setup and results are presented in Section 4. Finally, the paper is concluded in Section 5.

2. Related work

Velickovic et al. [27] introduced graph attention networks (GATs) for node classification in graph-structured data. By modelling the relationships between neighbouring nodes using a self-attention approach, GAT was used to learn the hidden representations of each node in the graph. The work demonstrated that GAT-based attention models can also be applied to arbitrarily structured graphs. Rather than modelling these relationships with identical weights as with GraphSage [28] or with pre-determined weights as with GCNs [26], GATs learn weights via a self-attention mechanism, such that nodes are weighted according to the information they provide relative to neighbouring nodes [27].
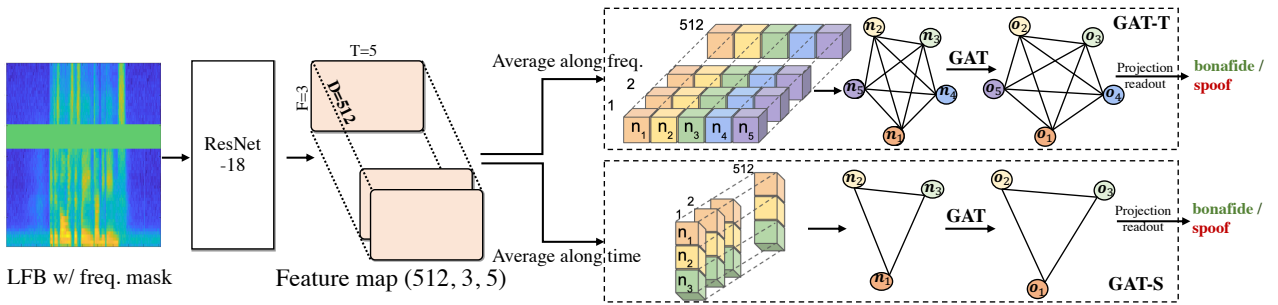
Figure 1: *Proposed GAT approach to anti-spoofing. Frequency-masked, high-resolution log-linear filterbank (LFB) features extracted from an input utterance are fed to a ResNet-18 network to extract high-level representations. The ResNet-18 feature map output is used to construct a graph with nodes n after averaging across time, (spectral attention, **GAT-S**) or frequency (temporal attention, **GAT-T**). We use a GAT layer followed by projection and readout to predict whether the input utterance is bona fide or spoofed.*

GNNs, including GCN and GAT variants, have been applied successfully in a variety of speech-related tasks. Zhang et al. [29] applied GCNs to few-shot audio classification. GNNs were used to predict an attention vector which helps to discriminate between different audio classes accordingly to their relative importance. Results showed the effectiveness of so-called attentional GNNs in transferring the metric representation learned from training classes to novel classes. Liu et al. [30] demonstrated the application of GNNs to neural speech synthesis, for which they are used to encode explicitly the syntactic relationship of the different elements within a sentence. Jung et al. [31] showed how GATs can be used to learn utterance-level relationships between speakers and how a GAT architecture with residual connections can be adapted to compute utterance-level similarity scores for speaker verification. To the best of our knowledge, this paper reports the first application of GATs to the anti-spoofing problem for which it is used to model the relationships between isolated spectral or temporal artefacts.

3. Anti-spoofing using GATs

In this section, we describe our approach to anti-spoofing using GATs. We describe the deep residual network (ResNet-18) architecture [32] which is used to extract high-level representation and then the GAT framework. Finally, we discuss the application of temporal and spectral attention. The architecture is illustrated in Figure 1.

3.1. Extraction of high-level representations

We use a ResNet-18 [32] system to learn high-level representations from acoustic features. The use of residual blocks facilitates the learning of a deeper network than is possible without them. The architecture is illustrated in Table 1. The network consists of a 3×3 convolutional layer and a 3×3 max pool layer to downsample the input feature map. Then follows four residual blocks and averaging either in time or frequency, depending on the domain in which attention is applied. The kernel, filter, stride and output sizes (no. of CNN filters × frequency × time) for each layer are illustrated in Table 1.

3.2. Graph Attention Network

High-level feature representations are fed to the GAT whose architecture is identical to that in [31]. A graph $\mathcal{G}$ is first formed from the ResNet-18 output e where $e \in \mathbb{R}^{N \times D}$, where N is the number of nodes (N=3 or 5 depending on

Table 1: *The details of ResNet-18 architecture. Convolutional layers are followed by batch normalisation and scaled exponential linear unit (selu) activation function. The output from average layer is use for graph formulation for GAT where averaging is applied along time or frequency domain depending on the configuration. Numbers denoted in output size are refers to (no. of CNN filters × frequency × time).*

Layer	Kernel	Filters	Stride	Output size
Freq. masking	-	-	-	60 ×202
Convolutional	3×3	64	1×2	64×64×103
Max pooling	3×3	-	2×2	64×32×52
Res. block (1)	3×3	64	2×2	64×32×52
Res. block (2)	3×3	128	2×2	128×16×26
Res. block (3)	3×3	256	2×2	256×8×13
Res. block (4)	3×3	512	2×2	512×3×5
Avg. along freq.	-	-	-	512×5
Avg. along time	-	-	-	512×3

whether the ResNet-18 output is averaged across time or frequency) and where D=512 is the feature/node dimension. The graph is *fully-connected* with edges between every pair of nodes, including self-connections. A GAT layer aggregates neighboring nodes using weights learned with a self-attention mechanism. Through this process, nodes are projected into another representation learned from the minimisation of a training loss. Using a GAT, more informative nodes are aggregated using greater weights, where the weight reflects the strength of the relationship between a given node pair.

The GAT output is denoted as:

$$GAT(\mathcal{G}) = \frac{1}{N} \sum_{n \in \mathcal{G}} \boldsymbol{o}_n W_{out}, \quad (1)$$

where W_{out} is the projection matrix which maps each node vector to a scalar and o_n is the output feature for node n which is determined according to GAT node propagation:

$$o_n = BN(W_{att}(\boldsymbol{m}_n) + W_{res}(\boldsymbol{e}_n)), \quad (2)$$

where BN is batch normalisation [33], W_{att} is a matrix which projects the aggregated information for each node n to the target dimensionality, whereas W_{res} projects the residual to match the target dimensionality.

The information from neighboring nodes is aggregated via self-attention according to:

$$m_n = \sum_{v \in \mathcal{M}(n) \cup \{n\}} \alpha_{v,n} e_v, \qquad (3)$$

where $\mathcal{M}(n)$ refers to the neighbouring nodes of node n, and $\alpha_{v,n}$ refers to the attention weight between nodes v and n. We consider the neighbouring nodes for node n to be the full set of nodes within the graph, including the node itself. The attention weight is calculated differently that in the original work [27] according to:

$$\alpha_{v,n} = \frac{\exp(W_{map}(e_n \odot e_v))}{\sum_{w \in \mathcal{M}(n) \cup \{n\}} \exp(W_{map}(e_n \odot e_w))}, \qquad (4)$$

where $W_{map} \in \mathcal{R}^D$ is the learnable map applied to the dot product and where $\odot$ denotes element-wise multiplication. Full details are available in [31].

3.3. Spectral and temporal attention

Psychoacoustics research [34] shows that the human auditory system can select the most informative spectral bands and acts to perform an auto-correlation corresponding to the temporal correlation between adjacent frames. In order to capture such cues, we apply GATs with attention in either spectral or temporal domains. Temporal attention (GAT-T, top-right of Figure 1) is applied to model the temporal relationships between adjacent frames and can help to capture complex nonlinear temporal artefacts. Spectral attention (GAT-S, bottom-right of Figure 1) is used to model the relationships between different sub-bands.

4. Experimental setup

Our work was performed using the ASVspoof 2019 Logical Access (LA) database [35] and default metrics. We report results for our specific implementation of GAT solutions with either temporal or spectral attention and compare these to results for competing, state-of-the-art systems.

4.1. Database and evaluation metric

The ASVspoof 2019 LA database has three independent subsets: train; development; evaluation. Spoofed speech in each dataset is generated using a set of different speech synthesis, voice conversion and hybrid algorithms [35]. Attacks in the training and development set were created with a set of 6 different algorithms (A01-A06), whereas those in the evaluation set were created with a set of 13 algorithms (A07-A19). We used the minimum normalised tandem detection cost function (t-DCF) [36] as a primary metric but also report results in terms of the pooled equal error rate (EER).

4.2. Baselines

We implemented three baselines: a high-spectral resolution linear frequency cepstral coefficient system with a conventional Gaussian mixture model classifier [7] (LFCC-GMM); a ResNet-18 system (the same as used in our GAT system, but with different attention mechanisms); a RawNet2 system [20]. We used the ResNet-18 systems in order to compare the benefit of GAT-based attention to alternative attention mechanisms: statistics pooling (SP) [37]; self-attentive pooling (SAP) which assigns different weights to different frames using a weighted mean; attentive statistical pooling (ASP) [38] which generates different weights for different frames according to both weighted means and weighted standard deviations.

4.3. Implementation details

While we obtained similar results using alternative representations such as linear frequency cepstral coefficients (LFCCs) [7, 39], all work reported in this paper was performed with 60-dimensional log linear filter bank (LFB) features extracted from 30 ms windows with a 10 ms frame shift and from audio waveforms which are truncated or concatenated to ≈ 4 second segments (64600 samples). To improve generalisation, we applied frequency masking augmentation [17, 40] to mask a random selection of contiguous frequency bands during training. The same frequency mask is applied to all training data within the same mini-batch. The maximum number of masked frequency bands was set to 12.

The ResNet-18 system operates upon the LFB features to produce high-level feature representations which are fed to the GAT after either temporal or spectral averaging. These are used to form an input graph where each node represents sub-bands or temporal segments with a 512-dimensional nodes/features. The GAT operates upon the input graph to generate the weighted output graph where the nodes/features dimension is 128. The latter are projected into a one-dimensional space (scalar) using a affine transforms (i.e. dense layer). Node features are finally aggregated using a readout layer to predict the output score. The entire system (both ResNet-18 and GAT) is trained using the ASVspoof 2019 LA training partition with binary cross-entropy (BCE) loss with a sigmoid activation function, a fixed learning rate of 0.0001 and a weight decay parameter of 0.0001. We used the standard Adam optimiser with a mini-batch size of 64 and train for 300 epochs.

4.4. Results

Results in terms of the min t-DCF are illustrated in Table 2 for development data (A01-06) and evaluation data (A07-A19). Columns labelled P1 and P2 show pooled min t-DCF and pooled EER results for each partition. For the development set, the baseline LFCC-GMM, GAT-T, GAT-S and ResNet-18 systems all yield min t-DCF values of close to zero, whereas the RawNet-2 system compares poorly. For almost all attacks in the evaluation set, GAT-T and GAT-S systems perform as well as, or better than the baseline systems. While pooled results show only a modest improvement for the GAT-T system, they are dominated by results for the A17 attack, for which the best results are obtained by the RawNet2 system. A comparison of results for other attacks and for ResNet-18, GAT-T and GAT-S systems show the benefit of graph-based approaches to model temporal or spectral relationships. The attention mechanisms of the three ResNet-18 systems are less effective, with pooled results being worse than for the LFCC-GMM system without any attention mechanism. We also note substantial differences in performance at the attack level. Whereas the use of spectral attention results in better performance for some attacks, temporal attention works better for others, and vice versa. These observations imply that different attacks exhibit different artefacts, none of which can be captured with a single classifier on its own.

Table 2: *Results for the ASVspoof 2019 logical access (LA) database in terms of min t-DCF for each attack in the development (A01-A06) and evaluation (A07-A19) partitions. Pooled min t-DCF (P1) and pooled EER (P2) are also shown for each partition. Results shown for the baseline systems and the proposed GAT systems with temporal attention (GAT-T) and spectral attention (GAT-S).*

System	A01	A02	A03	A04	A05	A06	P1	P2	A07	A08	A09	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	P1	P2
LFCC-GMM	.000	.000	.000	.001	.000	.000	.000	.00	.001	.001	.000	.154	.005	.115	.080	.069	.069	.006	.352	.074	.008	.090	**3.50**
RawNet2	.030	.020	.015	.043	.036	.042	.036	1.1	.098	.179	.073	.089	.042	.088	.020	.013	.073	.046	.240	.629	.058	.155	5.54
GAT-T	.000	.000	.000	.000	.000	.000	.000	.00	.000	.009	.001	.012	.009	.011	.016	.010	.009	.000	.715	.073	.001	**.089**	4.71
GAT-S	.000	.000	.000	.000	.000	.000	.000	.00	.000	.012	.000	.009	.006	.006	.009	.008	.009	.000	.642	.088	.000	.091	4.48
ResNet18-SP	.001	.000	.000	.002	.000	.002	.002	.07	.001	.025	.001	.009	.009	.006	.011	.010	.010	.001	.966	.183	.005	.114	6.82
ResNet18-SAP	.001	.000	.000	.001	.001	.000	.000	.03	.002	.080	.011	.023	.015	.029	.053	.037	.040	.001	.944	.272	.005	.138	7.11
ResNet18-ASP	.000	.000	.000	.001	.000	.000	.000	.01	.000	.037	.001	.009	.008	.007	.010	.010	.009	.001	.809	.291	.006	.127	6.22

4.5. Fusion

We performed fusion experiments using the support vector machine (SVM) based fusion approach described in [7] using different combinations of LFCC-GMM, GAT-T, GAT-S and RawNet2 systems. Since the GAT-T and GAT-S systems outperform ResNet-18 systems and since the GAT systems are in any case built on top of the ResNet-18 systems, we discounted the latter in our fusion experiments. Fusion results are presented in boldface in Table 3. Also included in Table 3 are results for a selection of top-performing, primary systems reported in the literature. These results further demonstrate the effectiveness of GNN-based attention approaches. Fusion results for different combinations of LFCC-GMM, RawNet2 and GAT-based systems all lead to improvements in performance. The best result comes from the fusion of all four systems which represents a 47% relative reduction in terms of min t-DCF over the best single, GAT-T system. Our results are also competitive with those of competing, top-performing systems. Only one achieves a lower min t-DCF than our best fused system. Even then, the gap is modest, while our system achieves a lower EER corresponding to a relative improvement of over 10%.

5. Conclusions

Graph attention networks (GATs) apply a self-attention mechanism to graph convolutional networks in order to model graph structured data. Each node in the graph is weighted according to its relevance to other nodes. The weights reflect the relationships between connected nodes, which here represents either a specific sub-band or temporal segment. Our work shows how GATs can be used to model these relationships using high-level representations extracted from deep residual networks and how this improves spoofing detection performance. Our experiments, performed on the ASVspoof 2019 Logical Access dataset, show that the GAT solution outperforms ResNet-18 and RawNet2 baseline systems by a substantial margin and that the GAT system with temporal attention also outperforms the high resolution LFCC-GMM system. GAT-based systems also outperform all other systems for 9 out of 13 spoofing attacks. Fusion experiments show that GAT-based systems are complementary to the baselines, with an ensemble system producing a 47% relative reduction in the t-DCF over the best, single GAT-T system.

While the RawNet2 baseline operates directly upon the raw signal, our GAT solution operates upon filterbank outputs. We are now working to improve computational efficiency such that our GAT solution can also be applied directly to the raw signal. Another target for our future work will be to determine the na-

Table 3: *A performance comparison for the evaluation partition of the ASVspoof 2019 logical access (LA) database in terms of pooled min t-DCF and pooled EER. Results shown for a set of state-of-the-art countermeasures reported in the literature (regular, roman font) and all systems reported in this paper (illustrated in italics). Also shown are results for different fusions of the high-spectral resolution LFCC-GMM baseline [7], RawNet2 system and the two GAT-based systems (illustrated in boldface).*

System	min-tDCF	EER
Spec+LFCC+CQT+SE-Res2Net [18]	0.0452	1.89
LFCC-GMM+GAT-S+GAT-T+RawNet2	**0.0476**	**1.68**
LFCC+LFCC-CMVN+CQT+FFT+LCNN+LFCC-GMM [9]	0.0510	1.86
ResNet18+LMCL+FM [17]	0.0520	1.81
GAT-S+GAT-T+RawNet2	**0.0635**	**2.21**
LFCC-GMM+RawNet2	**0.0643**	**2.33**
GAT-S+RawNet2	**0.0692**	**2.29**
Ensemble model [10]	0.0755	2.64
GAT-S+GAT-T	**0.0844**	**4.30**
GAT-T+RawNet2	**0.0854**	**2.61**
GAT-T	0.0894	4.71
LFCC-GMM [7]	0.0904	3.50
GAT-S	0.0914	4.48
Siamese CNN [11]	0.0930	3.79
FG-CQT+LCNN+CE [41]	0.1020	4.07
LFB-ResNet18 [17]	0.1090	4.04
ResNet-SP	0.1140	6.82
ResNet-ASP	0.1269	6.22
ResNet-SAP	0.1377	7.11
RawNet2	0.1547	5.54

ture and origins of the artefacts being detected with spectral and temporal attention and then to link these to specific spoofing attacks and the algorithmic origins.

6. Acknowledgements

7. References

[1] A. Nautsch, X. Wang, N. Evans, T. Kinnunen, V. Vestman, M. Todisco, H. Delgado, M. Sahidullah, J. Yamagishi, and K. A. Lee, "Asvspoof 2019: spoofing countermeasures for the detection of synthesized, converted and replayed speech," *IEEE Transactions on Biometrics, Behavior, and Identity Science (TIBOM)*, 2021.

[2] K. Sriskandaraja, V. Sethu, P. N. Le, and E. Ambikairajah, "Investigation of sub-band discriminative information between spoofed and genuine speech," in *Proc. INTERSPEECH*, San Francisco, USA, 2016, pp. 1710–1714.

[3] J. Yang, R. K. Das, and H. Li, "Significance of subband features for synthetic speech detection," *IEEE Transactions on Information Forensics and Security*, 2019.

[4] S. Garg, S. Bhilare, and V. Kanhangad, "Subband analysis for performance improvement of replay attack detection in speaker verification systems," in *International Conference on Identity, Security, and Behavior Analysis (ISBA)*, 2019, pp. 1–7.

[5] B. Chettri, T. Kinnunen, and E. Benetos, "Subband modeling for spoofing detection in automatic speaker verification," *Speaker Odyssey Workshop*, 2020.

[6] H. Tak, J. Patino, A. Nautsch, N. Evans, and M. Todisco, "An explainability study of the constant Q cepstral coefficient spoofing countermeasure for automatic speaker verification," in *Proc. Speaker Odyssey Workshop*, 2020, pp. 333–340.

[7] H. Tak, J. Patino, A. Nautsch *et al.*, "Spoofing Attack Detection using the Non-linear Fusion of Sub-band Classifiers," in *Proc. INTERSPEECH*, 2020, pp. 1106–1110.

[8] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks." in *Proc. Interspeech*, 2017, pp. 82–86.

[9] G. Lavrentyeva, S. Novoselov, A. Tseren *et al.*, "STC antispoofing systems for the ASVspoof2019 challenge," in *Proc. INTERSPEECH*, 2019, pp. 1033–1037.

[10] B. Chettri, D. Stoller, V. Morfi, M. A. M. Ramírez, E. Benetos, and B. L. Sturm, "Ensemble models for spoofing detection in automatic speaker verification," in *Proc. INTERSPEECH*, 2019, pp. 1118–1112.

[11] Z. Lei, Y. Yang, C. Liu, and J. Ye, "Siamese convolutional neural network using gaussian probability feature for spoofing speech detection," in *Proc. Interspeech*, 2020, pp. 1116–1120.

[12] C.-I. Lai, N. Chen, J. Villalba, and N. Dehak, "Assert: Antispoofing with squeeze-excitation and residual networks," in *Proc. Interspeech*, 2019.

[13] M. Alzantot, Z. Wang, and M. B. Srivastava, "Deep residual neural networks for audio spoofing detection," in *Proc. Interspeech*, 2019.

[14] P. Parasu, J. Epps, K. Sriskandaraja, and G. Suthokumar, "Investigating light-resnet architecture for spoofing detection under mismatched conditions," in *Proc. Interspeech*, 2020, pp. 1111–1115.

[15] P. Aravind, U. Nechiyil, N. Paramparambath *et al.*, "Audio spoofing verification using deep convolutional neural networks by transfer learning," *arXiv preprint arXiv:2008.03464*, 2020.

[16] Y. Zhang, F. Jiang, and Z. Duan, "One-class learning towards synthetic voice spoofing detection," *arXiv preprint arXiv:2010.13995*, 2020.

[17] T. Chen, A. Kumar, P. Nagarsheth, G. Sivaraman, and E. Khoury, "Generalization of audio deepfake detection," in *Proc. Speaker Odyssey Workshop*, 2020, pp. 132–137.

[18] X. Li, N. Li, C. Weng, X. Liu, D. Su, D. Yu, and H. Meng, "Replay and synthetic speech detection with res2net architecture," *arXiv preprint arXiv:2010.15006*, 2020.

[19] X. Wang and J. Yamagishi, "A comparative study on recent neural spoofing countermeasures for synthetic speech detection," *arXiv preprint arXiv:2103.11326*, 2021.

[20] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with rawnet2," in *Proc. ICASSP*, 2021.

[21] J.-w. Jung, S.-b. Kim, H.-j. Shim, J.-h. Kim, and H.-J. Yu, "Improved RawNet with Filter-wise Rescaling for Text-independent Speaker Verification using Raw Waveforms," in *Proc. INTERSPEECH*, 2020, pp. 1496–1500.

[22] J.-w. Jung, H.-S. Heo, J.-h. Kim, H.-j. Shim, and H.-J. Yu, "Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," in *Proc. INTERSPEECH*, 2019, pp. 1268–1272.

[23] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *Proc. SLT*, 2018, pp. 1021–1028.

[24] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2. IEEE, 2005, pp. 729–734.

[25] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2009.

[26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, 2018.

[28] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. NIPS*, 2017, pp. 1025–1035.

[29] S. Zhang, Y. Qin, K. Sun, and Y. Lin, "Few-shot audio classification with attentional graph neural networks." in *INTERSPEECH*, 2019, pp. 3649–3653.

[30] R. Liu, B. Sisman, and H. Li, "Graphspeech: Syntax-aware graph attention network for neural speech synthesis," in *Proc. ICASSP*, 2021.

[31] J.-w. Jung, H.-S. Heo, H.-J. Yu, and J. S. Chung, "Graph attention networks for speaker verification," in *Proc. ICASSP*, 2021.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. ICVPR*, 2016, pp. 770–778.

[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, vol. 1. IMLS, 2015, pp. 448–456.

[34] J. Brown, "Calculation of a constant Q spectral transform," *Journal of the Acoustical Society of America (JASA)*, vol. 89, no. 1, pp. 425–434, 1991.

[35] 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 synthetized, converted and replayed speech," *Computer Speech & Language (CSL)*, vol. 64, 2020, 101114.

[36] T. Kinnunen, K. Lee, H. Delgado, N. Evans, M. Todisco, J. Sahidullah, M.and Yamagishi, and D. A. Reynolds, "t-DCF: a detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification," in *Proc. Speaker Odyssey Workshop*, 2018, pp. 312–319.

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

[38] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. INTERSPEECH*, 2018, pp. 2252–2256.

[39] M. Sahidullah, T. Kinnunen, and C. Hanilci, "A comparison of features for synthetic speech detection," in *Proc. INTERSPEECH*, 2015, pp. 2087–2091.

[40] 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," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[41] Z. Wu, R. K. Das, J. Yang, and H. Li, "Light convolutional neural network with feature genuinization for detection of synthetic speech attacks," in *Proc. INTERSPEECH*, 2020.