IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 March 2024, accepted 15 May 2024, date of publication 1 July 2024, date of current version 26 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3421281

RESEARCH ARTICLE

Toward Improving Synthetic Audio Spoofing Detection Robustness via Meta-Learning and Disentangled Training With Adversarial Examples

ZHENYU WANG AND JOHN H. L. HANSEN, (Fellow, IEEE)
Center for Robust Speech Systems (CRSS), The University of Texas at Dallas, Richardson, TX 75080, USA

Corresponding author: John H. L. Hansen (john.hansen@utdallas.edu)

This work was supported by The University of Texas at Dallas from the Distinguished University Chair in Telecommunications Engineering held by John H. L. Hansen.

ABSTRACT Advances in automatic speaker verification (ASV) promote research into the formulation of spoofing detection systems for real-world applications. The performance of ASV systems can be degraded severely by multiple types of spoofing attacks, namely, synthetic speech (SS), voice conversion (VC), replay, twins and impersonation, especially in the case of unseen synthetic spoofing attacks. A reliable and robust spoofing detection system can act as a security gate to filter out spoofing attacks instead of having them reach the ASV system. A weighted additive angular margin loss is proposed to address the data imbalance issue, and different margins has been assigned to improve generalization to unseen spoofing attacks in this study. Meanwhile, we incorporate a meta-learning loss function to optimize differences between the embeddings of support versus query set in order to learn a spoofing-category-independent embedding space for utterances. Furthermore, we craft adversarial examples by adding imperceptible perturbations to spoofing speech as a data augmentation strategy, then we use an auxiliary batch normalization (BN) to guarantee that corresponding normalization statistics are performed exclusively on the adversarial examples. Additionally, A simple attention module is integrated into the residual block to refine the feature extraction process. Evaluation results on the Logical Access (LA) track of the ASVspoof 2019 corpus provides confirmation of our proposed approaches' effectiveness in terms of a pooled EER of 0.87%, and a min t-DCF of 0.0277. These advancements offer effective options to reduce the impact of spoofing attacks on voice recognition/authentication systems.

INDEX TERMS Audio spoofing detection, simple attention module, additive angular margin loss, relation network, meta-learning, disentangled training, adversarial examples.

I. INTRODUCTION

In recent years, ASV has been used extensively for personal biometric authentication. An ASV system aims to verify an identity claim of an individual from their voice characteristics [1]. Spoofed voice attacks involve an attacker who masquerades as the target speaker to gain access into the ASV system [2], [3] for use of resources, services or devices. In most cases, the zero-effort imposters can be easily caught by a general ASV system, but more sophisticated spoofing attacks pose a significant threat to system robustness and credibility [4]. With easy access to biometric information of personal voices, spoofing attacks are inevitable [5]. Such a potential system security breach represents a key reliability concern of ASV systems. To address this, an audio spoofing detection system generates countermeasure scores for each audio sample to distinguish between genuine (bona-fide) and spoofed speech, which allows for deployment of the

The associate editor coordinating the review of this manuscript and approving it for publication was Antonio J. R. Neves.

ASV system into real-world situations where diverse audio spoofing attacks could occur.

Since 2015 [6], [7], [8], [9], the ASVspoof community has been at the forefront of anti-spoofing research with a series of biannual challenges. Their aims are to foster progress in development of audio spoofing detection to protect ASV systems from manipulation. Existing audio spoofing detection systems have been proposed to address two different mainstream use case scenarios: logical access (LA) and physical access (PA), which involves three major forms of spoofing attack, namely synthetic, converted, and replayed speech. Spoofing attacks on the physical access track are direct attacks at the transmission stage, where genuine audio samples are represented by a replay device to microphone input of the ASV system [10]. With continuous progress in speech synthesis [11] and voice conversion [12], such advanced techniques are able to impersonate a target speaker's voice, compromising ASV reliability. Logical access attacks, generated by the latest speech synthesis and voice conversion technologies, can be more challenging and perceptually indistinguishable from genuine speech.

Such spoofed speech generated by different attacking algorithms contains artefacts, which reside in specific sub-bands or temporal segments [13], [14], [15], [16], [17], [18], [19]. Specifically, artefacts serve as indicative cues to distinguish genuine speech from spoofed speech. Additionally, artefacts present in different attacks tend to be heterogeneous, which depend on the specific spoofing algorithm employed. Reliable detection often relies upon the ensemble system with multiple subsystems tuned to capture specific forms of artefacts. Here, we seek to develop a single system that delivers reliable detection performances across a spectrum of diverse spoofing attacks.

As in many related fields of speech processing, a growing number of researchers are adopting end-to-end model architectures that operate directly upon raw speech waveforms [20], [21], [22], [23], which bypass limitations introduced by the utilization of knowledge-based, hand-crafted acoustic features, (e.g., Mel-frequency cepstral coefficients, and Melfilterbank energy features [24], [25], [26], [27]). Following this trend, RawNet2 [28], combined with the merits of RawNet1 [29], takes in raw waveforms and tends to yield more discriminative representations compared to traditional spoofing detection solutions. To learn a meaningful filterbank structure, the first layer of RawNet2 is the same as that of SincNet [23], [30], which implements band-pass filters based on parametrized sinc functions. The upper layers are comprised of residual blocks [31] to extract frame-level representations, and the GRU [32] layer serves to aggregate utterance-level representations. Here, filter-wise feature map scaling (FMS) [28] is employed as an attention mechanism to derive more discriminative representations.

To further enhance the model's representation ability to construct informative features, the Squeeze-and-Excitation (SE) component has been extensively used in residual blocks, which recalibrates channel-wise feature maps by modelling the inter-dependencies between each channel [33]. Given an intermediate feature map in a residual block, the convolutional block attention module (CBAM) [34] sequentially infers attention maps along the channel and spectral-temporal dimensions, and then attention maps are used to refine the input features. In contrast to channel-wise and spatial-wise attention modules, a simple attention module (SimAM) [35] infers 3-dimensional attention weights for adaptive feature refinement. Inspired by neuroscience theories, they propose to optimize an energy function to attain the importance of each neuron. Attention modules noted here represent general plug-and-play modules, which can be injected into each residual block of any feed-forward convolutional neural network (CNN) architecture seamlessly with negligible additional parameters and is also end-to-end trainable along with CNNs.

In most cases, the spoofing detection classifier is trained using a cross-entropy loss with softmax (denoted by CE-Softmax loss). A reliable spoofing detection model should aggregate embeddings from the same identity and separate clusters for different identities. However, the spoofing detection model optimized by Softmax loss is not generalizable enough, and performance degradation is observed when evaluated on unseen spoofing attacks. As in some speaker verification tasks [36], [37], [38], [39], the end-to-end system is able to learn discriminative representations directly, however, it is time-consuming for training and requires complex data preparation (e.g., semi-hard example mining). To address this issue with negligible computational overhead, margin-based losses such as angular softmax loss (denoted by A-Softmax loss) [40], additive margin softmax loss (denoted by AM-Softmax loss) [41], and additive angular margin loss (denoted by AAM-Softmax loss) [42], can be considered to encourage intra-class compactness and inter-class segregation. Previous research has investigated the impact of margin-based losses for speaker embedding learning [43], [44], [45], It has been proven that margin serves as a vital factor in discriminative embeddings learning and leads to a significant overall performance improvement [46].

Due to the continuing evolution of voice conversion and speech synthesis techniques, a growing number of emerging unseen spoofing attacks poses a great threat to the reliability of spoofing detection systems. The generalization capability of existing solutions could be subject to a limited variety of known attacks. Meta-learning has recently become one research hotspot in deep-learning-based approaches. Several novel meta-learning approaches [47], [48], [49] propose to learn a shared metric space between the embeddings of unseen examples from a test set, and known classes in the training set. Ko et al. [47] employed prototypical networks (PN), a typical meta-learning architecture, to enhance the discriminative power of the speaker embedding extractor. While episodic optimization could be insufficient to obtain the optimal embedding distribution for unseen classes,

Kye et al. [48] perform global classification for each sample within every episode. By combining two learning schemes, significant improvement is observed for short-duration utterance speaker recognition. With consideration of unseen samples in the test set during the training phase, the model achieves a consistent framework across train and test, which boosts discriminative power for unseen samples.

Deep-learning-based approaches always require a large amount of data to tune model parameters during training, where the spoofing detection accuracy and model robustness can be subject to training data size. Data augmentation is a commonly-used method to obtain additional synthetically modified data. Adversarial examples can also be obtained by attacking model vulnerability, which has been adopted as a free resource for model training in different tasks [50], [51], [52]. Adversaries are crafted by adding imperceptible perturbations to original training data, and the modified data is used to mislead a well-trained model [53]. Adversarial examples are commonly viewed as a threat to neural network models, which behave in a similar manner to spoofing attacks. Inspired by this, adversaries can be treated as additional training examples to boost spoofing detection performance if harnessed in the proper manner. Although there probably exists a trade-off between model accuracy and robustness to adversarial perturbations [54]. Unexpected benefits can be observed when adversarial examples are involved in the model training (e.g., interpretable feature representations that align well with salient data characteristics [54] and improved robustness to corruptions concentrated in the high-frequency domain [55]). Considering different data distributions between original training data and adversaries, Xie et al. proposed an auxiliary batch normalization (BN) to disentangle model training for accurate statistics estimation [56].

In this study, we begin with a variant of RawNet2 [28] as our backbone model architecture. We develop an end-to-end robust spoofing detection system to reliably detect spoofing attacks on the LA track of the ASVspoof2019 corpus without score-level ensembles. We inject three different attention modules (e.g., SE, CBAM, SimAM) into each residual block, respectively, to enhance model representation ability. Zhang et al [57] proposed a one-class learning to improve detection performance on unknown synthetic spoofing attacks, which results from over-fitting of known attacks. Based on the AAM-Softmax loss [42], we assign different weights and margins to each class (e.g., genuine and spoofed) for alleviating the unbalanced data and over-fitting problem. To mitigating the adverse impact of unseen spoofing attacks, we further adopt the meta-learning loss to adaptively learn a shared metric space between unseen samples and known attacks. The relation network [58] is employed to compare samples in the support and query sets, where an additional neural network serves to parameterize the comparison metric. Previous research has explored taking adversarial examples as augmentation data to improve an attention-based keyword spotting system [59]. Here, Our interest lands on disentangled learning with adversarial examples to enhance system robustness, such that the complementarity across original training data and adversaries could be fully exploited. We hypothesize that the AAM-Softmax loss for global classification, the meta-learning episodic loss, and the loss from adversarial examples exhibit specific unexpected benefits during model training, thus bringing out an overall joint optimization as a powerful ensemble method for discriminative representations learning. The main contributions of this research study are:

1. Investigated three extensions to the RawNet2-based model, and analyzed the effectiveness of each attention module in improving model performance.

2. A weighted AAM-Softmax loss is employed for binary classification to encourage intra-class compactness and inter-class separability in the embedding space.

3. proposed a meta-learning framework to enhance model generalization capability to unseen spoofing attacks, and integrated episodic and global classification to encourage discriminative embedding learning.

4. Adversarial examples are treated as additional training samples, with an auxiliary BN used for adversaries to perform disentangled training.

5. Joint optimization with weighted AAM-Softmax loss, meta-learning loss, and adversarial loss are performed to boost the entire spoofing detection system performance for detection of LA-based spoofing attacks.

The rest of the paper is organized as follows. Sec. II investigates conventional and existing state-of-the-art spoofing detection approaches. Sec. III details each component of the proposed spoofing detection framework. Sec. IV comprises the specifics of data, evaluation metrics, and experimental configurations. Sec. V presents experimental results and corresponding analysis of observations. Lastly, conclusions are drawn in Sec. VI. The overview of this study is presented in Fig. 1.

II. RELATED WORK

This section presents a detailed investigation of existing state-of-the-art countermeasures for audio synthetic spoofing detection. The countermeasures are broadly classified into three categories: conventional handcrafted features with machine learning classifiers, enhanced deep learning approaches, and state-of-the-art end-to-end approaches.

A. CONVENTIONAL APPROACHES

Researchers in the spoofing detection community have worked on finding handcrafted features that reflect artefacts based on phase spectrum, magnitude spectrum, pitch, group delay, etc., to distinguish between spoofed and genuine speech [60], [61], [62], [63], [64], [65]. Since feature extraction and classifiers are two main components of spoofing detection systems, the Gaussian mixture model (GMM), its variants, and support vector machine (SVM) classifiers [60], [62], [66], [67], [68] have been extensively explored for

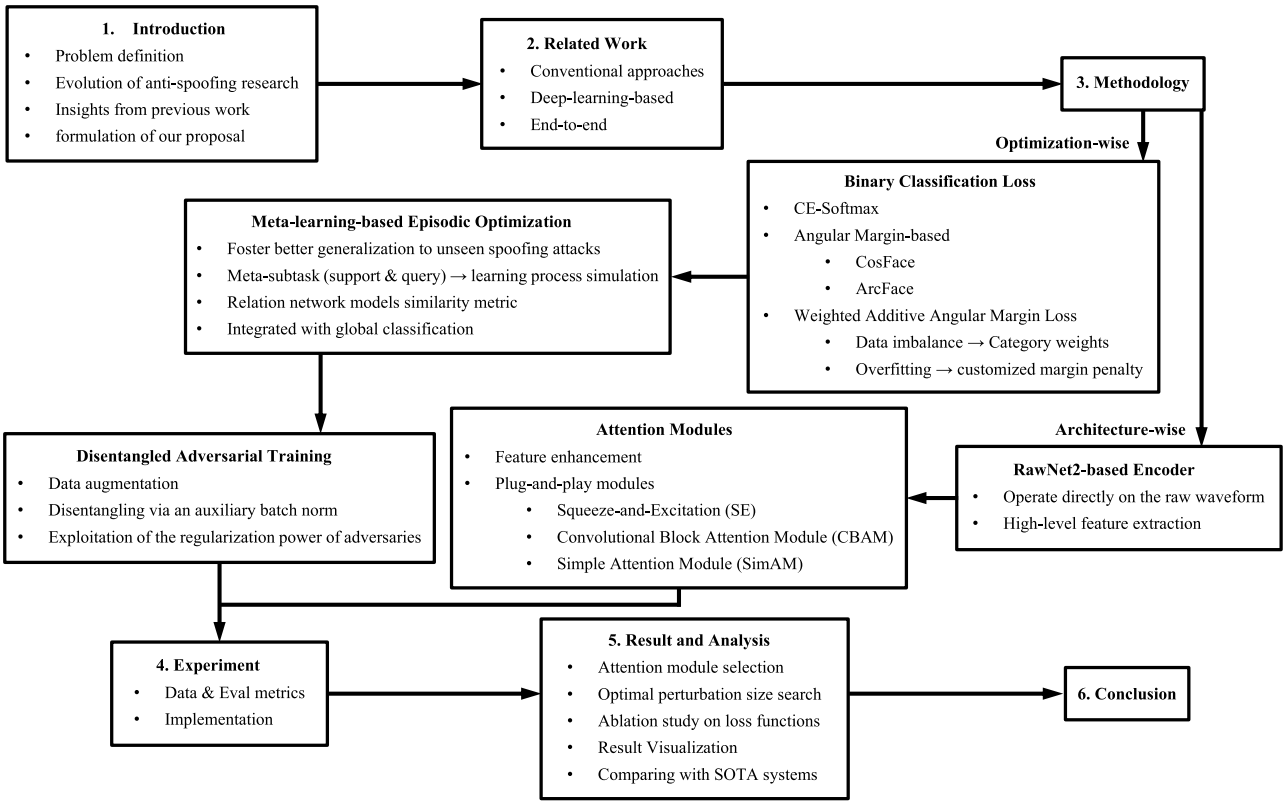

FIGURE 1. Study overview.

synthetic spoofing detection. However, it has been shown that efforts on complex machine-learning-based classifiers are less effective than crafting informative features [69].

The Constant-Q Cepstral Coefficients (CQCC) [61] are extracted with the constant-Q transform (CQT), which captures manipulation artefacts that are indicative of spoofing attacks. Patel et al. proposed a combination of cochlear filter cepstral coefficients (CFCC) and change in instantaneous frequency (IF) (i.e., CFCCIF) to detect genuine versus spoofed speech [70]. Additionally, improved classification performance was observed when CFCCIF was combined with Mel frequency cepstral coefficients (MFCC) [71] features. For other effective features, the high-dimensional magnitude-based features (i.e., log magnitude spectrum, and residual log magnitude spectrum) and phase-based features (i.e., group delay function, modified group delay function, baseband phase difference, pitch synchronous phase, instantaneous frequency derivative) have been introduced in [72].

Artefacts from synthetic speech reside in different subbands, therefore, subband processing is explored to extract discriminative features such as linear frequency cepstral coefficients (LFCC) [14], energy separation algorithm instantaneous frequency cepstral coefficients (ESA-IFCC) [69], and constant-Q statistics-plus-principal information coefficient (CQSPIC) [73]. Sriskandarajaet al. proposed another subband processing approach to perform a hierarchical scattering decomposition through a wavelet filterbank, then the absolute values of the filter outputs are used to yield a scalogram [74].

Previous studies in image processing have extensively explored the concept of texture. It has been found that texture descriptors such as local binary patterns (LBP) and local ternary patterns (LTP) are effective for image classification tasks. Next, a novel countermeasure based on the analysis of sequential acoustic feature vectors using Local Binary Patterns (LBPs) was presented to detect LA attacks [67]. Reference [65] also employed relative phase shift features and MGDF-based features to detect synthesized/converted speech. LBPs and MGDF [66] are less successful at differentiating between genuine and spoofed samples because they are susceptible to noise, which generates patterns that are similar for both classes.

B. DEEP-LEARNING-BASED APPROACHES

Recent efforts have witnessed a rise in utilization of deep-learning-based methods to detect synthesized/converted spoofing attacks. Alzantot et al. [75] built three variants of ResNet [31] that ingested different feature representations, namely, MFCC, log-magnitude STFT, and CQCC. The fusion of three variants of ResNet (i.e., MFCC-ResNet, CQCC-ResNet, and Spec-ResNet) has outperformed the spoofing detection baseline methods (i.e., LFCC-GMM, CQCC-GMM). Wang et al. [59] used a 135-layer deep dense convolutional network to detect voice transformation

spoofing. Similarly, Lai et al. [76] adopted two low-level acoustic features, namely, log power magnitude spectra (logspec) and CQCC as input, where the DNN models hinged on variants of all the Squeeze-Excitation (SE) network and residual networks were trained to detect spoofed speech. In [77], spectral log-filter-bank and relative phase shift features were taken as input to train DNN classifiers for synthetic spoofing detection. A five-layer DNN classifier with a novel human log-likelihoods (HLL) scoring method was proposed, which was mathematically proven to be more suitable for synthetic spoofing detection than the classical LLR scoring method [78].

Concerning feature engineering, it was found that utilizing the DNN-based model as a pattern classifier was less effective than using it for representation learning followed by traditional machine learning classifiers (i.e., GMM or SVM as the classifier). In [79], a spoofing-discriminant network was used to extract the representative spoofing vector (s-vector) at the utterance level. Next, the Mahalanobis distance, along with normalization, was applied to the computed s-vector for LA attack detection. In [80], bottleneck features with frame-level posteriors were extracted by the DNN-based model, followed by a standard GMM classifier built with acoustic-level features and bottleneck features. In [81], a light convolutional gated recurrent neural network was used to extract utterance-level representations, later with extracted deep features, back-end classifiers (i.e. linear discriminant analysis (LDA), and its probabilistic version (PLDA), and SVM) performed the final genuine/spoofed classification). A similar approach has been proposed to learn spoofing identity representations [82], where DNN-based frame-level features and RNN-based sequence-level features were incorporated in model training (i.e. LDA, gaussian density function (GDF), and SVM) for spoofing detection. Despite the extra computation costs introduced by feature engineering, deep-learning-based methods deliver better classification performances than traditional methods.

C. END-TO-END APPROACHES

Today, end-to-end approaches have achieved state-of-the-art performance in a variety of audio processing applications [83], [84]. Bypassing complex feature engineering, the end-to-end framework takes raw waveforms as input for representation learning and yields corresponding classification decisions, which encapsulate pre-processing and post-processing components within a single network [85], [86]. Muckenhirn developed a convolutional neural network-based approach to learn features and then built a classifier in an end-to-end manner [87]. A joint architecture called convolutional Long-Short Term Memory (LSTM) neural network (CLDNN) with raw waveform front-ends was proposed for spoofing detection [88], [89]. In the literature [90], an end-to-end system based on a variant of RawNet2 encoder [28] and spectro-temporal graph attention networks [91] was used to learn the relationship between cues spanning different sub-bands and temporal segments. Jung et al. developed an end-to-end architecture incorporated with a novel heterogeneous stacking graph attention layer, followed by a new max graph operation and readout scheme, to facilitate the concurrent modelling of temporal-spectral graph attention for improved spoofing detection [92]. Following previous work [93] based on a variant [94] of differentiable architecture search [95], Ge et al. explored how to learn automatically the network architecture towards a spoofing detection solution [96]. End-to-end approaches represent a new direction of anti-spoofing study.

III. METHODOLOGY

This section describes each of the relevant components for building our proposed synthetic spoofing detection architecture. This comprises the encoder for general representation learning, attention modules for feature enhancement, and three specific optimization/training schemes to improve model accuracy, generalization ability to unseen attacks, and robustness.

A. RAWNET2-BASED ENCODER

Instead of using hand-crafted features as inputs [97], the Rawnet2-based model operates directly upon the raw waveform without preprocessing techniques [90], [98]. A variant of the RawNet2 model was introduced in [29] for the speaker embedding learning and applied subsequently for building spoofing detection systems [90], [99]. Here, we adopt that model to extract high-level representations $F \in \mathbb{R}^{C\times S\times T}$ (C, S, and T are the number of channels, spectral bins, and the temporal sequence length, respectively) from raw waveforms. According to the literature [29], [30], [99], approaches equipped with a bank of sinc filters show superior effectiveness in terms of both convergence stability and performance. Therefore, a sinc convolution layer is employed for front-end feature learning. The sinc layer transforms the raw waveform in the time domain using a set of parameterized sinc functions that are analogous to rectangular band-pass filters [100], [101]. Each filter within the filterbank possesses its center frequencies based on a mel-scale. Cut-in and cutoff frequencies are fixed to alleviate over-fitting to training data due to training data sparsity or rather limited genres of different spoofing attacks (only 6 for the training and development partitions from the ASVspoof 2019 LA database).

The output of each filter is treated as a spectral bin, subsequently, the output of the sinc layer is transformed into a 2-dim time-frequency representation by adding a channel dimension. The result is fed into stacked 2-dim residual blocks [31] with pre-activation [102] for high-level feature learning. Each residual block is comprised of a batch normalization layer [103], a 2-dim convolution layer, scaled exponential linear units (SeLU) [104], and a max pooling layer for down-sampling. The specifics of our model configuration are summarized in Tab. 1.

TABLE 1. Model configuration.

Layer	Input: 64600 samples	Output shape
Sinc Layer	Conv-1D(129,1,70) add channel (TF representation) Maxpool-2D(3) BN & SeLU	(70,64472) (1,70,64472) (1,23,21490)
Res block × 2	Conv-2D(32,(2,3),(1,1),1) BN&SeLU Conv-2D(32,(2,3),(0,1),1) Maxpool-2D((1,3))	(32,23,2387)
Res block × 4	Conv-2D(64,(2,3),(1,1),1) BN&SeLU Conv-2D(64,(2,3),(0,1),1) Maxpool-2D((1,3))	(64,23,29)
AdaptivePooling	AdaptiveAvgPool2d((1,29))	(64,1,29)
GRU	GRU(64)	(64,)
Output	FC(2)	(2,)

B. ATTENTION MODULES

The fundamental building block of convolutional neural networks (CNNs) serve as the convolution operator, allowing networks to learn informative features by combining spatial and channel-wise information within the local receptive fields at each layer. Plug-and-play attention modules [33], [34], [105] as an effective component can refine the intermediate feature maps within a CNN block, so as to boost the model capacity. Researchers are of interest to formulate effective attention modules for feature enhancement, which enable networks to improve the quality of channel-wise or spatial encoding throughout the feature hierarchy.

1) SQUEEZE-AND-EXCITATION

Squeeze-and-Excitation (SE) module can be integrated into residual blocks for learning informative representations by the insertion after a non-linearity following a convolution [33]. The module as a computational unit is comprised of two fully connected layers to learn the importance of each channel, which is built on transforming by first compressing and then expanding the full average channel vector. Given the intermediate feature map $\mathbf{x} \in \mathbb{R}^{C\times S\times T}$ of the Residual block as input, the SE module first calculates the channel-wise mean statistics $\mathbf{e} \in \mathbb{R}^{C}$. Here, the c-th element of $\mathbf{e}$ is

$$\mathbf{e}_c = \frac{1}{S\times T}\sum_{i=1}^{S}\sum_{j=1}^{T}\mathbf{x}_{c,i,j}, \tag{1}$$

where C, S, and T represent channel, frequency, and time dimensions. The SE module then scales this channel-wise mean by two fully connected layers to obtain the channel-wise attention weights s of the various channels:

$$\mathbf{s} = \sigma(\mathbf{W}_2 f(\mathbf{W}_1\mathbf{e} + \mathbf{b}_1) + \mathbf{b}_2), \tag{2}$$

where W and $\mathbf{b}$ denote the weight and bias of a linear layer. Also, $f(\cdot)$ is the activate function of the rectified linear unit (ReLU), and $\sigma(\cdot)$ is the sigmoid function.

2) CONVOLUTIONAL BLOCK ATTENTION MODULE

The convolutional block attention module (CBAM) [34] extends channel-wise attention into two separate dimensions, referred to as the channel and spatial (frequency-temporal) attention modules. Next, the input feature maps are multiplied by attention maps for adaptive feature refinement. With the merits of a lightweight and effective module, the CBAM can be integrated into any CNN-based architecture, which has previously been successfully applied for speaker verification [106]. Given the input feature map $\mathbf{x} \in \mathbb{R}^{C\times S\times T}$, the overall attention process sequentially infers a 1-dim channel attention map $\mathbf{M_c} \in \mathbb{R}^{C\times 1\times 1}$ and a 2-dim frequency-temporal attention map $\mathbf{M_{ft}} \in \mathbb{R}^{1\times S\times T}$. The feature refinement process is formulated as,

$$\begin{aligned}\mathbf{x}' &= \mathbf{M_c}(\mathbf{x}) \otimes \mathbf{x},\\ \mathbf{x}'' &= \mathbf{M_{ft}}(\mathbf{x}') \otimes \mathbf{x}',\end{aligned} \tag{3}$$

where $\otimes$ denotes element-wise multiplication. The final refined output $\mathbf{x}''$ is obtained by broadcasting the attention values (i.e., $\mathbf{M_c}$ and $\mathbf{M_{ft}}$) along with the frequency-temporal and channel dimensions accordingly.

3) SIMPLE ATTENTION MODULE (SIMAM)

Inspired by attention mechanisms in the human brain based on certain well-known neuroscience theories [107], the simple attention module (SimAM) [35] is proposed to optimize an energy function for encapsulating the relevance of each neuron. The parameter-free simple attention module (SimAM) has proven to be flexible and effective in enhancing the learning capabilities of convolution networks with negligible computational costs [35], and subsequently applied in speaker verification [108]. By optimizing an energy function to capture the significance of each neuron, it generates 3-dim attention weights for the feature map in a convolution layer.

$$e_t(W_t, b_t, \mathbf{y}, x_i) = (y_t - \hat{t})^2 + \frac{1}{M-1}\sum_{i=1}^{M-1}(y_o - \hat{x}_i)^2. \tag{4}$$

Given the feature map $\mathbf{x} \in \mathbb{R}^{C\times S\times T}$ in a single channel, t denotes the target neuron. x_i is other neurons, where i is the index over the frequency-temporal domain and $M = S \times T$ is the number of neurons for each channel. Here, $\hat{t} = W_t t + b_t$ and $\hat{x}_i = W_t x_i + b_t$ are linear transforms for t and x_i. Eq. 4 obtains its minimal value when $\hat{t} = y_o$ and $\hat{x}_i = y_t$. Considering y_o and y_t as two distinct values, for simplicity, binary labels (i.e., 1 and -1) are assigned to y_o and y_t in the final energy function with a regularizer,

$$\begin{aligned}e_t(W_t, b_t, \mathbf{y}, x_i) = \frac{1}{M-1}\sum_{i=1}^{M-1}(-1 - (W_t x_i + b_t))^2\\ + (1 - (W_t t + b_t))^2 + \lambda {W_t}^2.\end{aligned} \tag{5}$$

There are extensive computational resources needed to optimize each of the neuron's attention weights using a general optimizer such as SGD. Fortunately, a closed-form

solution can be leveraged to derive the transform's weight W_t and bias b_t with optimal energy. Specifically, the minimal energy of a neuron x in an input feature map $\mathbf{x} \in \mathbb{R}^{C\times H\times W}$ is formulated as:

$$e_x^* = \frac{4(\hat{\sigma}^2+\lambda)}{(x-\hat{u})^2+2\hat{\sigma}^2+2\lambda}, \tag{6}$$

where $\hat{\mu} = \frac{1}{H\times W}\sum_{i=1}^{H\times W} x_i$, $\hat{\sigma}^2 = \frac{1}{H\times W}\sum_{i=1}^{H\times W}(x_i-\hat{\mu})^2$, and λ is a hyper parameter. Each neuron within a channel shares the statistics μ and σ, which hence significantly lowers computation costs. Given that research in neuroscience demonstrates an inverse relationship between the energy of e_x^* and the significance of each neuron x [109], the refinement process of a feature map can be written as,

$$\hat{\mathbf{x}} = \sigma(\frac{1}{\mathbf{E}}) \otimes \mathbf{x}, \tag{7}$$

where $\mathbf{E}$ groups all energy values of e_x^*, with $\sigma(\cdot)$ denoting the sigmoid function. In this study, we inserted a SimAM after the first convolution layer in each residual block of the base model.

C. BINARY CLASSIFICATION LOSS

In this section, the fundamental cross-entropy loss with softmax and angular margin-based losses are discussed, and the weighted additive angular margin loss is proposed for our binary classification. During training, each mini-batch contains N utterances from either spoofed or genuine speech, whose feature embedding vectors are $\mathbf{x}_i \in \mathbb{R}^D$, with the corresponding spoofing identity labels being y_i, where $1 \le i \le N$ and $y \in \{0, 1\}$ (i.e., 0 denotes spoofed speech and 1 represents the genuine).

1) REVISITING CE-SOFTMAX LOSS

The Softmax loss is comprised of a softmax function integrated with a multi-class cross-entropy loss, which is formulated as,

$$\begin{aligned} L_S &= -\frac{1}{N}\sum_{i=1}^{N} w_{y_i} log \frac{e^{\mathbf{W}_{y_i}^T\mathbf{x}_i}}{e^{\mathbf{W}_{y_i}^T\mathbf{x}_i}+e^{\mathbf{W}_{1-y_i}^T\mathbf{x}_i}} \\ &= \frac{1}{N}\sum_{i=1}^{N} log(1+e^{(\mathbf{W}_{1-y_i}-\mathbf{W}_{y_i})^T\mathbf{x}_i}), \end{aligned} \tag{8}$$

where $\mathbf{W}$ represents the weight vector of the last layer of the encoder trunk, and $\mathbf{W}_0, \mathbf{W}_1 \in \mathbb{R}^D$ are the weight vectors of the spoofed class and genuine class, respectively. w_{y_i} is the weight of the i-th sample with label y_i. This loss function merely computes penalties for classification error and does not explicitly encourage intra-class compactness or inter-class separation.

2) ANGULAR MARGIN-BASED LOSS

The softmax loss can be reformulated so that the posterior probability only hinges on the cosine value of the angle between the weights and input vectors. With normalized unit vectors of $\widehat{\mathbf{W}}$ and $\hat{\mathbf{x}}$, the loss function termed as Normalized Softmax Loss (NSL), is written as,

$$\begin{aligned} L_N &= -\frac{1}{N}\sum_{i=1}^{N} log \times \frac{e^{|\widehat{\mathbf{W}}_{y_i}^T||\hat{\mathbf{x}}_i|cos(\theta_{y_i,i})}}{e^{|\widehat{\mathbf{W}}_{y_i}^T||\hat{\mathbf{x}}_i|cos(\theta_{y_i,i})}+e^{|\widehat{\mathbf{W}}_{1-y_i}^T||\hat{\mathbf{x}}_i|cos(\theta_{1-y_i,i})}} \\ &= \frac{1}{N}\sum_{i=1}^{N} log(1+e^{(cos(\theta_{1-y_i,i})-cos(\theta_{y_i,i}))}), \end{aligned} \tag{9}$$

where $cos(\theta_{y_i,i})$ denotes the dot product of normalized vector $\widehat{\mathbf{W}}$ ($|\ \widehat{\mathbf{W}}\ | = 1$) and $\hat{\mathbf{x}}_i$ ($|\ \hat{\mathbf{x}}_i\ | = 1$). Next, $\mathbf{x}_o = W_{y_i}^T\mathbf{x}_i + b_{y_i}$ describes the final linear transformation, where $x_i \in \mathbb{R}^D$ is the penultimate linear layer's output (i.e., D-dim embedding) of the i-th sample with label y_i and $x_o \in \mathbb{R}^2$ is the last linear layer's output. Finally, $W_{y_i} \in \mathbf{R}^D$ denotes the y_i-th column of the weight $W \in \mathbf{R}^{D\times 2}$ and b_{y_i} is the bias term. This bias term b_{y_i} is set to 0 here, therefore, the linear transformation is reformulated as $W_{y_i}^T\mathbf{x}_i = |\ W_{y_i}\ ||\ \mathbf{x}_i\ | cos\theta_{y_i,i}$, where $\theta_{y_i,i}$ is the angle between the weights and the input feature. Likewise, this loss function has the same issue as the Softmax loss in that it only computes penalties based on classification error. This results in embeddings which are learned by the NSL as not being sufficiently discriminative. Modifications are proposed here to mitigate this issue, where an additive margin is introduced with the AM-Softmax to make the embedding space of the two classes close to their weights $\mathbf{W}_0 - \mathbf{W}_1$ and $\mathbf{W}_1 - \mathbf{W}_0$. The formula for the AM-Softmax (CosFace) can now be written as,

$$\begin{aligned} L_C &= -\frac{1}{N}\sum_{i=1}^{N} log \\ &\quad \times \frac{e^{s(|\widehat{\mathbf{W}}_{y_i}^T||\hat{\mathbf{x}}_i|cos(\theta_{y_i,i})-m)}}{e^{s(|\widehat{\mathbf{W}}_{y_i}^T||\hat{\mathbf{x}}_i|cos(\theta_{y_i,i})-m)}+e^{s|\widehat{\mathbf{W}}_{1-y_i}^T||\hat{\mathbf{x}}_i|cos(\theta_{1-y_i,i})}} \\ &= \frac{1}{N}\sum_{i=1}^{N} log(1+e^{s(m-cos(\theta_{y_i,i})+cos(\theta_{1-y_i,i}))}), \end{aligned} \tag{10}$$

where s denotes a hyper-parameter that rescales up the gradient instead of the numerical values becoming too small within the training phase, which helps to expedite optimization. Feature maps are rescaled by s, where they are accordingly projected onto a hypersphere with radius s.

Furthermore, an additive angular margin penalty m between $\mathbf{W}_{y_i}$ and $\mathbf{x}_i$ is also incorporated into the equation in order to simultaneously enhance the intra-class compactness and inter-class separability, termed as the AAM-Softmax (ArcFace) loss [42], formulated as,

$$\begin{aligned} L_A &= -\frac{1}{N}\sum_{i=1}^{N} log \\ &\quad \times \frac{e^{s(|\widehat{\mathbf{W}}_{y_i}^T||\hat{\mathbf{x}}_i|cos(\theta_{y_i,i}+m))}}{e^{s(|\widehat{\mathbf{W}}_{y_i}^T||\hat{\mathbf{x}}_i|cos(\theta_{y_i,i}+m))}+e^{s|\widehat{\mathbf{W}}_{1-y_i}^T||\hat{\mathbf{x}}_i|cos(\theta_{1-y_i,i})}} \\ &= \frac{1}{N}\sum_{i=1}^{N} log(1+e^{s(cos(\theta_{1-y_i,i})-cos(\theta_{y_i,i}+m))}). \end{aligned} \tag{11}$$

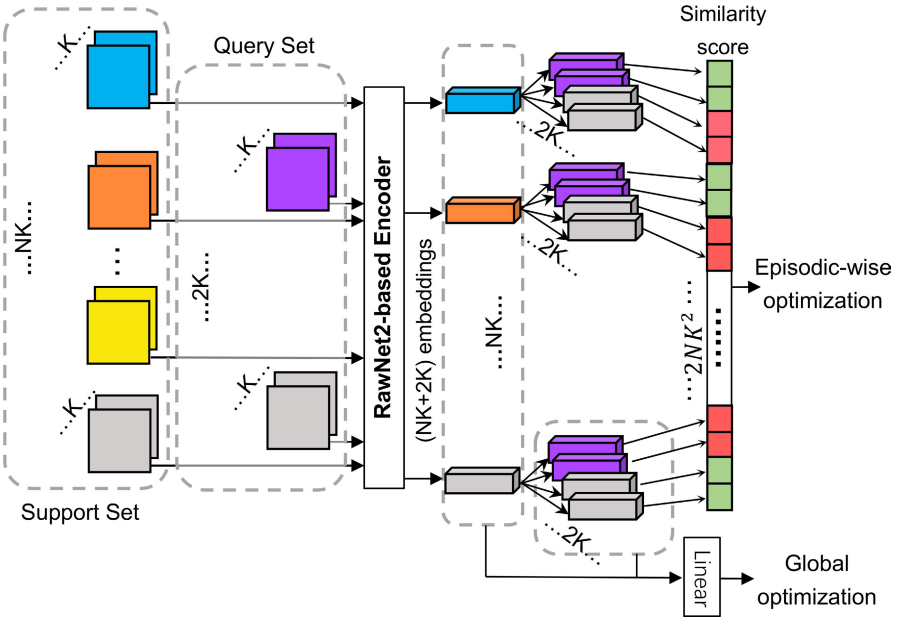

FIGURE 2. Joint optimization scheme. All spoofing samples and embeddings are color-coded to represent different types of spoofing attacks, while genuine speech is gray. The similarity score in green denotes a match: $r_{i,j} = 1$, likewise, those in red are unmatched: $r_{i,j} = 0$.

3) WEIGHTED ADDITIVE ANGULAR MARGIN LOSS FOR BINARY CLASSIFICATION

Since the dataset employed in this study is unbalanced (e.g., genuine versus spoofed), different classes are expected to possess individual weights for loss calculation. Here, w_{y_i} denotes a manual rescaling weight assigned to class y_i. By adding this weight factor into the equation, a benefit is possible for the scenario when the training set is unbalanced (e.g., more spoofing samples are included versus genuine samples). Inspired by earlier research [57], the different additive angular margin penalty m_{y_i} can be injected into the corresponding target angle, which prevents the model from overfitting unseen spoofing attacks to known attacks. Specifically, there exists a distribution mismatch for spoofing attacks in the training and evaluation partition. Two different margins are therefore assigned to the bona-fide speech and spoofing attacks, which encourages better compactness for bona-fide samples, and at the same time greater isolation of the spoofing attacks. The AAM-softmax (see Eq. 11) is hence reformulated as,

$$L_W = -\frac{1}{N}\sum_{i=1}^{N} log\frac{w_{y_i}\, e^{s(cos(\theta_{y_i,i}+m_{y_i}))}}{e^{s(cos(\theta_{y_i,i}+m_{y_i}))} + e^{s\, cos\theta_{(1-y_i,i)}}}$$
$$= \frac{1}{N}\sum_{i=1}^{N} log\, w_{y_i}(1 + e^{s(cos(\theta_{1-y_i,i})-cos(\theta_{y_i,i}+m_{y_i}))}). \quad (12)$$

D. META-LEARNING EPISODIC OPTIMIZATION

Meta-learning is focused on developing a task-oriented model to enhance the learning ability by conducting optimization within each subtask (i.e., an episode or a mini-batch), instead of overall engagement for a given problem. A meta-subtask is composed of a support set and a query set. Examples in the support set are used for learning how to directly solve a subtask, while the query set is used for subtask performance assessment. At each step in meta-learning, model parameters are updated based on a randomly selected subtask. Since the network is presented with various tasks at each iteration, this enforces learning to distinguish inhomogeneous examples in general, rather than a specific subset of examples. In realistic settings of the spoofing detection, training data would contain N different types of spoofing attacks manipulated by various spoofing techniques (e.g. A01-A06 in the ASVspoof 2019 logical access (LA) dataset [8], [110]), but the unseen attacks could still occur in the evaluation phase. To simulate this situation during training, we first randomly select K spoofing examples $\mathbf{x}^s$ from each spoofing type respectively, along with 2K bona-fide examples $\mathbf{x}^b$. Next, one spoofing type is randomly included in the query set while keeping all other types in the support set within each subtask. Here, 2K bona-fide examples are equally distributed between the query and support set. As a result, we obtain the following support set $\mathcal{S} = \{\mathbf{x}_i^s\}_{i=1}^{(N-1)\times K} \cup \{\mathbf{x}_i^b\}_{i=1}^{K}$ and query set $\mathcal{Q} = \{\mathbf{x}_j^s\}_{j=1}^{K} \cup \{\mathbf{x}_j^b\}_{j=1}^{K}$. Given this formulation of support and query pairs in each episode, with a finite number of spoofing types of spoofing attacks enrolled into the model, the spoofing attack types in the query set can now vary in each subtask.

To compare samples in the support set and query set, we use the relation network [58], which parameterizes the non-linear similarity metric using a neural network. Specifically, the relation network simultaneously models the feature representation and metric over a set of subtasks in order to generalize to unseen spoofing attacks. Given the input sample and its corresponding label in terms of $(\mathbf{x}, y)$,

samples from the support set $\mathcal{S}$ and query set $\mathcal{Q}$ are fed through the encoder f_θ (see Sec. III-A). Next, an embedding $f_\theta(\mathbf{x}_i)$ from the support set and an embedding $f_\theta(\mathbf{x}_j)$ from the query set are concatenated to formulate one pair. Considering the number of samples in $\mathcal{S}$ ($|\mathcal{S}| = NK$) and $\mathcal{Q}$ ($|\mathcal{Q}| = 2K$), each subtask/mini-batch is comprised of $2NK^2$ permutations as a set $\mathcal{P}$ of pairs for metric-based meta-learning. Finally, each pair is processed by the relation module f_ϕ, which yields a scalar relation output score representing the similarity between the feature representation pair,

$$r_{i,j} = f_\phi([f_\theta(\mathbf{x}_i), f_\theta(\mathbf{x}_j)]), \tag{13}$$

where [., .] denotes the concatenation operation, the network f_ϕ treats the relation score as a similarity measure [58], therefore $r_{i,j}$ is defined as,

$$r_{i,j} = \begin{cases} 1, & if\ y_i = y_j, \\ 0, & otherwise. \end{cases} \tag{14}$$

The network f_θ and f_ϕ are jointly optimized using mean square error (MSE) objective as in [58], where the relation network output is treated as the output of a linear regression model. The MSE loss for meta-learning here is written as,

$$L_M = \frac{1}{2NK^2} \sum_{i=1}^{NK} \sum_{j=1}^{2K} (r_{i,j} - 1(y_i == y_j))^2. \tag{15}$$

Additionally, we enforce the model to classify samples in both the support and query sets against the entire set of classes in the training set. The entire meta-learning scheme with global classification is depicted in Fig. 2. A hyper-parameter λ balances the weighted AAM loss (Eq. 12) and the MSE loss, where the fusion loss is hereby written as,

$$L_F = L_W + \lambda L_M. \tag{16}$$

E. DISENTANGLED ADVERSARIAL TRAINING

1) ADVERSARIAL EXAMPLES

Next, adversarial examples can be obtained by adding imperceptible but malicious perturbations to the original training data, which can compromise the accuracy of a well-trained neural network [111]. Adversarial examples are commonly treated as a threat to neural networks. Here, we leverage both original training data and corresponding adversarial examples to train networks for enhanced system performance. Consider the default adversary generation method, the Fast Gradient Sign Method (FGSM), which has random perturbation and has been used for maximizing the inner part of the saddle point formulation [112]. A more powerful multi-step attacker based on the projected gradient descent (PGD) (see Eq. 17) is adopted here to produce adversaries on the fly [51]. Given an input training sample $\mathbf{x} \in \mathbb{D}$ with a corresponding ground-truth label y, adversary generation is conducted in an iterative manner as follows,

$$\mathbf{x}_t^{adv} = \Pi_{\mathbf{x}+\mathbb{S}}(\mathbf{x}_{t-1}^{adv} + \alpha\ \mathrm{sgn}(\nabla_{\mathbf{x}} \mathrm{L}(\theta, \mathbf{x}, y))), \tag{17}$$

where Π denotes a projection operator, $\mathbb{S}$ represents the allowed perturbation size that formalizes the manipulative power of the adversary, α is the step size, $\mathrm{L}(\cdot, \cdot, \cdot)$ stands for the loss function, and θ indicates the model parameters. Eq. 17 then illustrates one step of a multi-step attacker to generate adversaries.

The adversarial training framework proposed in [51] only used maliciously perturbed samples to train networks. Here, the robust optimization objective illustrates a saddle point problem composed of an inner maximization problem and an outer minimization problem written as,

$$\arg\min_\theta \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{D}}(\max_{\delta\in\mathbb{S}} \mathrm{L}(\theta, \mathbf{x}+\delta, y)). \tag{18}$$

For each training data sample $\mathbf{x} \in \mathbb{D}$, a set of allowed perturbations $\delta \in \mathbb{S}$ are introduced to formalize adversaries. Such a training framework has merits as described in [54], [55], and [113], but cannot generalize well to original training data [51], [114].

To encourage full exploitation of the complementarity nature between original training data and corresponding adversarial examples, adversarial examples are treated as augmented data, and incorporated with the original data for model training. The learning objective is formulated as,

$$\begin{aligned} &\arg\min_{\theta,\phi} \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{D}}(\mathrm{L}_F(\theta, \phi, \mathbf{x}, y)) \\ &\quad + \arg\min_\theta \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{D}}(\max_{\delta\in\mathbb{S}} \mathrm{L}_W(\theta, \mathbf{x}+\delta, y), \end{aligned} \tag{19}$$

where L_F and L_W is referred to as Eq. 16 and Eq. 12, respectively.

2) DISENTANGLING VIA AN AUXILIARY BN

Earlier studies on adversarial attacks have demonstrated that training using adversarial examples can cause label leaking (i.e., the neural network overfits to the specific adversary distribution), which leads to compromised model performance [50], [111]. Under the assumption that adversarial examples and original data come from different underlying distributions, Xie et al. proposed disentangled training via an auxiliary batch norm (BN) to decouple the batch statistics between original and adversarial data in the normalization layers during model training [56]. This approach would allow for better exploitation of the regularization power of adversarial data. For the original mini-batch training data and corresponding adversarial data at each training step, we hereby utilize two BNs (i.e., one main BN and one auxiliary BN) for specific data partitions while the remaining model parameters are jointly tuned. Corresponding data flows in different architectures (i.e., with conventional BN and with Auxiliary BN) are illustrated in Fig. 3. At the evaluation phase, we maintain only the main BN for data distribution normalization while bypassing the auxiliary one.

3) ADVERSARIAL TRAINING SCHEME

Compared to adversarial training [50], [111], disentangled learning can fully exploit the complementarity nature

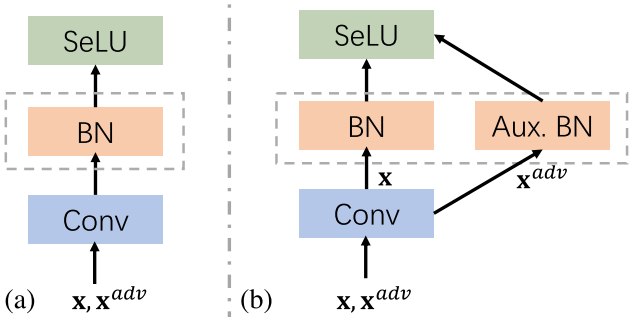

FIGURE 3. Alternate data flow options between architectures with conventional BN (a) and with auxiliary BN (b).

between original training data and corresponding adversarial examples. Adversarial examples are generated during model training. In each training iteration, we treat the original data mini-batch as adversarial examples at the initial step (i.e., $t = 0$). Multiple steps for attacking are performed using the auxiliary BNs. We then derive the adversaries for the current mini-batch. The objective of incorporating adversarial examples into training is to improve the model generalization ability to unseen spoofing attacks. We thereby substitute adversaries corresponding to the bona-fide label (i.e., $y^{adv} = 1$) with original training samples to maintain an identical data distribution of genuine samples in the adversarial mini-batch. Subsequently, we submit the original mini-batch and adversarial mini-batch to the same network, while calculating the loss via main BNs and auxiliary BNs for specific mini-batches, respectively. Finally, we minimize the total loss for the network parameter updates (Eq. 19). We present the complete training scheme with adversarial examples in Algorithm 1.

Algorithm 1 Adversarial Training Scheme

Input: Original training data with labels $\{\mathbf{x}^{org}, \mathbf{y}^{org}\} \sim \mathbb{R}$
Output: Encoder parameter θ; relation module parameter ϕ
Given S training iterations, T attacking steps, and batch size N
for s in $1 : S$ **do**
  sample mini-batch $\{\mathbf{x}_i^{org}, y_i^{org}\}_{i=1}^m$
  Suppose $\{\mathbf{x}_{i,0}^{adv}, y_{i,0}^{adv}\}_{i=1}^m = \{\mathbf{x}_i^{org}, y_i^{org}\}_{i=1}^m$
  for t in $1 : T$ **do**
    Generate adversarial examples $\{\mathbf{x}_{i,t}^{adv}, y_{i,t}^{adv}\}_{i=1}^m \sim \mathbb{R} \cup \mathbb{S}$ at current time step t using auxiliary BNs w/ Eq. 17
  end for
  for n in $1 : N$ **do**
    if $y_n^{adv} = 1$ **then** ▷ label 1 represents the bona-fide
      $\mathbf{x}_n^{adv} = \mathbf{x}_n^{org}$ ▷ maintain original data for genuine samples
    end if
  end for
  Compute $\mathrm{L}_F(\mathbf{x}^{org}, \mathbf{y})$ w/ Eq. 16
  Compute $\mathrm{L}_W(\mathbf{x}^{adv}, \mathbf{y})$ w/ Eq. 12
  Update θ and ϕ w/ Eq. 19
end for

IV. EXPERIMENT

A. DATASET AND EVALUATION METRICS

The ASVspoof 2019 corpus on the Logical Access (LA) track [8], [110] is adopted in this work to train and test models. The corpus consists of three partitions, namely, training, development, and evaluation subsets, with each subset containing genuine and spoofed samples. Different spoofing methods (i.e., voice conversion and speech synthesis) are employed to create spoofing attacks [115]. The evaluation partition features 13 different attacking genres (A07-A19); the training and development subsets contain 6 different spoofing attacks (A01-A06). Model selection and gauging emergence of the over-fitting are dependent on the development subset. Given the 13 algorithms used for generating evaluation data, 2 algorithms are also used in training and development subsets, while the other 11 algorithms are unseen/uninvolved for train and development data. Bona-fide samples are collected from 107 speakers. The number of audio samples in each subset are 25,380, 24,986, 71933 for training, development, and evaluation, respectively. The durations of each speech sample ranges from 1-2 sec, with all audio samples in each subset stored in flac format.

We adopt the equal error rate (EER) and the minimum normalized tandem detection cost function (min t-DCF) [116], [117] as the metrics for assessing system performance. Wang et al. [118] found that spoofing detection systems initialized with different random seeds can deliver different results by a substantial margin. As such, all results reported here are the best results from three runs with different random seeds.

B. IMPLEMENTATION DETAILS

The currently proposed spoofing detection system is implemented using Pytorch toolkit. Each input segment is approximately 4 sec in duration, and processed by a RawNet2-based encoder [28]. The RawNet2-based encoder consists of a sinc-convolution layer [30] and six stacked residual blocks with pre-activation [102]. The sinc-convolution layer is initialized with a bank of 70 mel-scaled filters. Each residual block is stacked with a batch normalization layer [103], a scaled exponential linear unit (SeLU) activation [104], a 2D convolution layer, and a max pooling layer. The first two residual blocks are equipped with 32 filters, while the remaining four have 64 filters. After the encoder, there is an adaptive average pooling layer to aggregate frequency-wise information. Next, a gated recurrent unit (GRU) with 64 hidden units is used to aggregate sequential features within the temporal domain. The intermediate features are then processed using a fully connected layer with 64 units. The 64-dim embeddings extracted at the final layer are subsequently used for calculating similarity scores and estimating classification loss. The relation network has two fully connected layers with 64 units each. Additionally, we employ Projected Gradient Descent (PGD) [51] under an L_∞ norm as the default attacker for crafting adversarial examples on-the-fly. The perturbation size δ (see Eq. 18) is set to 0.002. The number of attacking iterations is set to 12. The attack step size (see Eq. 17) is fixed to $\alpha = 0.0001$, and the balance hyper-parameter λ in Eq. 16 is set to 0.8.

We conducted extensive experiments using multiple setup combinations with loss functions, attention modules, and disentangled training with adversarial examples. The baseline system employs the RawNet2-based encoder to learn the spoofing identity, which minimizes a cross-entropy loss w.r.t the network parameters for gradient updates. The ASVspoof 2019 corpus is data-unbalanced with a 1:9 ratio of genuine samples to spoofing samples, thereby assigning specific weights to genuine and spoofing classes with 0.1 and 0.9, respectively. Likewise, category-wise weights w_{y_i} in Eq. 12 are designated in the same way. Also, there are two hyper-parameters in Eq. 12, where the scale s is fixed to 32, while the margin m_0, m_1 are set to 0.2 and 0.9, respectively. The batch size in each experiment is set to 16. During the meta-learning sampling phase in one episode/mini-batch, we randomly select 2 ($K = 2$, see Sec. III-D) samples from each attacking type (A01-A06) and 4 samples from genuine samples (4 genuine samples are equally split into the support and query sets). With regard to our model optimization strategy, we utilize the Adam optimizer [119] with a learning rate of 0.0001 using a cosine annealing learning rate decay. The model in each experiment was trained for 100 epochs. For the SimAM attention module (see Sec. III-B3), the hyper-parameter λ in Eq. 6 is set to 0.0001.

V. RESULT AND ANALYSIS

To thoroughly evaluate our proposed methods, we assess the feature enhancement effectiveness for different attention modules, then search for a rational perturbation size to craft adversarial examples, and present an ablation study on loss functions, and a comparison of our results to the state-of-the-art systems in this section.

A. ATTENTION MODULE SELECTION

The RawNet2-based encoder model learns high-level representations for spoofing identities, while there are several attention modules that can be leveraged to refine the intermediate feature maps. As noted in Sec. IV-B, the baseline system uses a RawNet2-based encoder, which is trained with a cross-entropy (CE) loss (see Sec. III-C1). We compare spoofing detection system performances derived from the encoder (see Sec. III-A) equipped with different attention modules. The results are presented in terms of min t-DCF and EER in Tab. 2.

As results shown in Tab. 2, the baseline system achieves acceptable results in terms of min t-DCF and pooled EER, which is probably owing to the interpretation of single-channel 2-dim feature map generated by the sin-convolution layer, thereby enhancing the feature representation ability. Each attention module improves system performance to varying degrees, which means they contribute to refining the intermediate feature maps. The system with CBAM yields a lower EER, but the SE module outperforms CBAM in terms of min t-DCF. The system with CBAM delivers a better spoofing detection performance than that with SE while resulting in a higher expected detection cost. Compared to the previous two attention modules, the SimAM encourages learning the informative feature maps along with superior system performance.

TABLE 2. The effectiveness for different attention modules.

System	min t-DCF	EER (%)
RawNet2 + CE (baseline)	0.0566	1.67
RawNet2 w/ SE before BN	0.0492	1.64
RawNet2 w/ SE after BN	0.0412	1.62
RawNet2 w/ CBAM before BN	0.0514	1.55
RawNet2 w/ CBAM after BN	0.0456	1.52
RawNet2 w/ SimAM before BN	**0.0406**	**1.41**
RawNet2 w/ SimAM after BN	0.0458	1.43

TABLE 3. The effectiveness for different perturbation size.

System	min t-DCF	EER (%)
RawNet2 + CE (baseline)	0.0566	1.67
RawNet2 + CE + Adv. ($\delta = 0.1$)	0.0649	2.06
RawNet2 + CE + Adv. ($\delta = 0.01$)	0.0564	1.69
RawNet2 + CE + Adv. ($\delta = 0.001$)	0.0372	1.33
RawNet2 + CE + Adv. ($\delta =$**0.002**)	**0.0356**	**1.142**
RawNet2 + CE + Adv. ($\delta = 0.003$)	0.376	1.43
RawNet2 + CE + Adv. ($\delta = 0.004$)	0.468	1.55
RawNet2 + CE + Adv. ($\delta = 0.0001$)	0.0514	1.63

We found that different insertion positions of each attention module can result in varied spoofing detection performances. Inserting the attention module right after/before the BN in the residual block improves distinctive feature learning. In each residual block, either the SE or CBAM encourages feature refinement more significantly, while inserted right after BN, and SimAM conducts the more effective feature enhancement while inserted after the first convolutional layer and before the BN. The best result in this section is delivered with the RawNet2-based encoder equipped with SimAM, while the module is inserted before the BN in each residual block. The EER is reduced to 1.41% with + 15.57% relative reduction compared to the result of the baseline system, and min t-DCF improves to 0.0406 with a + 28.27% relative reduction.

B. SEARCH FOR OPTIMAL PERTURBATION SIZE

During adversaries generation, a set of allowed perturbations $\delta \in \mathbb{S}$ (see Eq. 18) formalize the manipulative power of adversarial examples. We investigate multiple orders of magnitude of perturbation size on the effectiveness of enhancing model robustness/accuracy.

As shown in Tab. 3, spoofing detection performances are compromised while training with slightly larger perturbation (i.e., $\delta = 0.1/0.01$), potentially due to the fact that excessive perturbations could in fact blur the distinctive original identity pattern, causing misclassification. In contrast, a slightly small perturbation size (i.e., $\delta = 0.0001$) is trivial to generate strong enough adversaries in order to improve robustness. Additionally, [51] found that increasing the capacity of the network when training using only original

TABLE 4. Breakdown EER performance of 13 attacks in the ASVspoof 2019 LA evaluation partition, pooled min t-DCF, and pooled EER,The best performance for each column is marked in boldface.

System	A07	A08	A09	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	min t-DCF	EER (%)
Baseline	1.79	0.69	0.06	2.42	1.28	2.73	0.38	0.75	1.12	0.61	1.81	3.52	1.33	0.0566	1.67
RawNet2 w/ SimAM + CE	**0.57**	0.41	0.04	0.91	0.34	1.96	0.12	0.34	**0.42**	0.67	1.85	3.58	1.23	0.0406	1.41
Replace CE w/ AAM	0.92	0.27	0.02	1.27	0.33	1.83	0.15	0.26	0.69	0.68	2.09	2.67	1.08	0.0399	1.36
Replace CE w/ WAAM	1.30	**0.14**	**0.00**	1.65	**0.31**	1.70	0.19	0.14	0.96	0.65	2.34	1.77	0.91	0.0389	1.29
+ MSE	0.91	0.29	0.02	1.39	**0.31**	1.30	**0.08**	0.18	0.61	**0.24**	2.11	2.25	0.97	0.0289	0.99
+ Adv.	0.63	0.22	**0.00**	**0.85**	0.35	**0.91**	0.34	**0.12**	0.89	0.75	**1.79**	**1.58**	**0.63**	**0.0277**	**0.87**

training data improves robustness against adversaries, and this effect is greater when considering adversaries with small perturbations. Moreover, performance on the original training samples can be degraded by the small capacity of the network, providing some form of robustness against adversaries [51]. We observe that adversarial examples with a spectrum of perturbation sizes (i.e., $\delta \in [0.001, 0.004]$) are exerting varying degrees of influence on boosting system performance. This is especially the case for adversaries generated with perturbation size $\delta = 0.002$, which maximize contributions to system performance improvement.

C. ABLATION STUDY ON LOSS FUNCTIONS

We perform an ablation study on diverse loss functions based on the RawNet2-based architecture. An ablation study serves to understand the contribution of each component to overall system performance. The base model (see Sec. III-A) equipped with SimAM (see Sec. III-B3) exhibits a satisfying encoding ability to absorb distinctive information from input features. Firstly, we minimize a cross-entropy (CE) loss (see Eq. 8) w.r.t. the network parameter for gradient updates. To encourage better binary classification, we replace the CE loss with the weighted additive angular margin (WAAM) loss (see Eq. 12). Subsequently, we incorporate the meta-learning mean square error (MSE) loss (see Eq. 15) into a fused total loss function (see Eq. 16). Additionally, to leverage the regularization power of adversarial examples, we conduct disentangled training (see Sec. III-E) with a mixture of original training data and corresponding spoofing adversaries under a combined learning objective (see Eq. 19).

A breakdown of the results on the evaluation partition with unknown attacks is illustrated in Tab. 4. Each proposed loss function boosts system performance with incremental improvement. The WAAM loss outperforms cross-entropy loss, specifically, the relative reduction in EER is up to + 8.5%, and + 4.2% on min t-DCF. Additionally, the parameter w_{y_i} in Eq. 12 is beneficial to the system performance improvement, which is designed to alleviate the impact by data imbalance. The joint optimization of the WAAM loss and meta-learning MSE loss promotes better generalization to unseen spoofing attacks, yielding a better spoofing detection result. In addition, disentangled learning further facilitates distinctive embedding learning, leading to enhanced system accuracy/robustness. Revisiting the baseline system (also see in Tab. 2), the best solution outperforms the baseline system by + 47.9% relative EER reduction, and + 51.1% relative min t-DCF reduction.

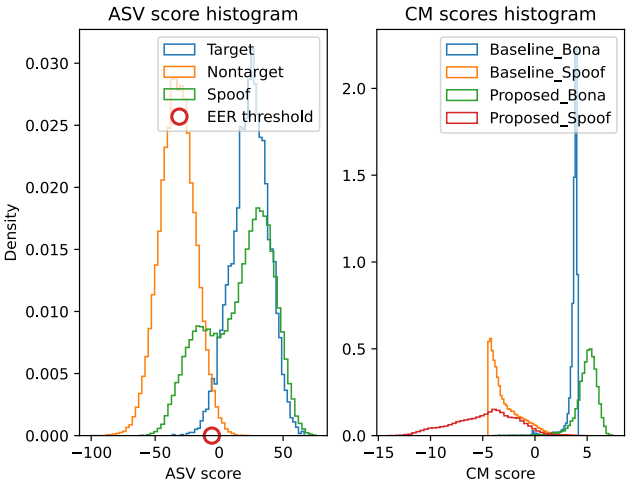

FIGURE 4. Distribution of ASV scores and countermeasure scores.

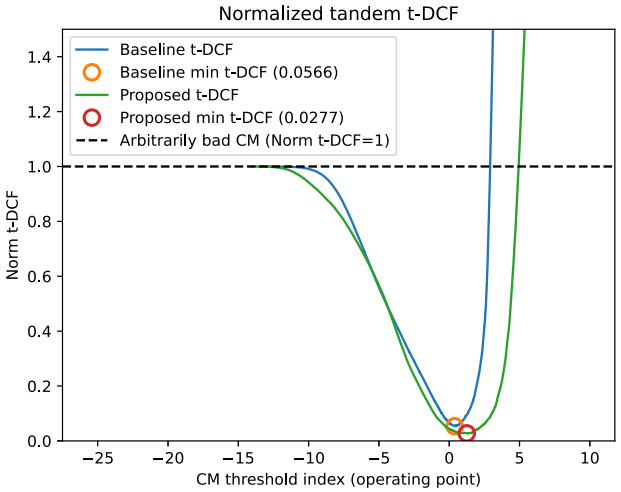

FIGURE 5. Normalised ASV-constrained t-DCFs plot for the baseline CM system and proposed CM system on ASVspoof 2019 LA track.

D. VISUALIZATION OF IMPROVEMENT BY PROPOSED APPROACH

Aligning with the definition of a tandem system as a cascade of countermeasure (CM) and ASV systems in [117], the CM

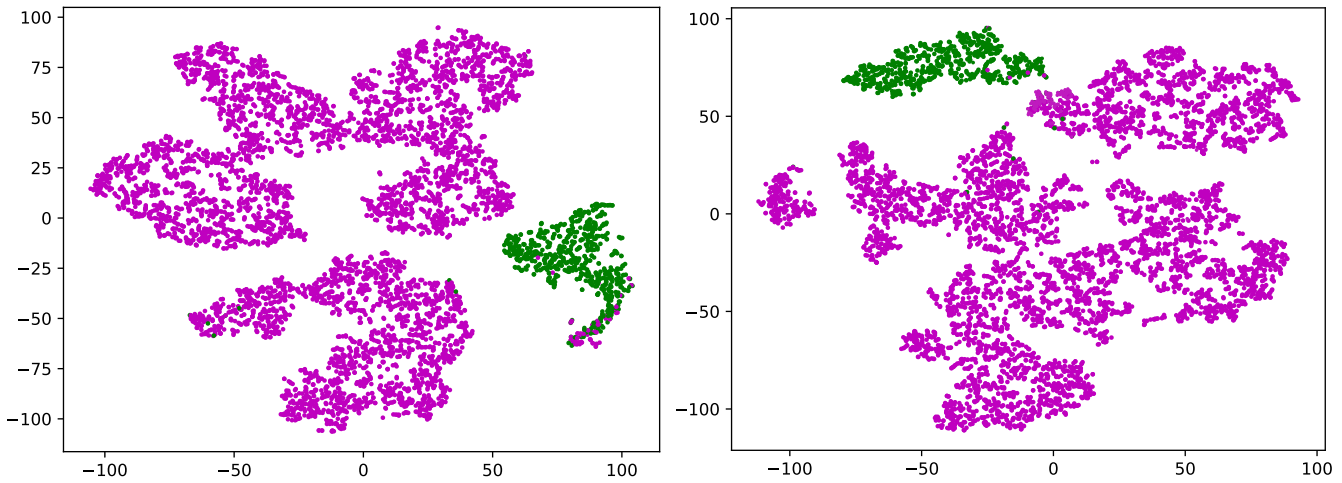

FIGURE 6. t-SNE visualization of feature embedding for the baseline CM system and proposed CM system on ASVspoof 2019 LA track (genuine samples are color-coded as green, spoofed samples are present in purple).

acts as a gate to filter out spoofing attacks before reaching the ASV system. The tandem system can encounter three types of trials: (i) target, (ii) non-target and (iii) spoof. Only the target trials should be accepted while both non-target and spoof trials should be rejected. Fig. 4 presents score density distributions, which comprises ASV scores (on the left panel) and CM scores (on the right panel) for both baseline CM system and proposed CM system. In Fig. 4, ''Bona'' and ''Spoof'' ''Spoof'' mean bona fide speech and spoofing attacks, respectively. The proposed CM system tends to yield a wider score consistent with the idea of better generalization by potentially encompassing scores from unseen samples. Additionally, a larger margin is observed between the genuine speech space and the spoofing space in the proposed approach, which further forces inter-class separability.

Fig. 5 shows the t-DCFs plot which provides insight into the observation made from Fig. 4. The tandem detection cost function (t-DCF) [116] metric reflects the overall performance of a combined ASV and CM system. Here, the ASV-constrained normalised t-DCF curves are shown for the baseline CM system and proposed CM system, while evaluated on the ASVspoof 2019 LA track, when varying the CM threshold. The proposed CM system reached a lower minimum t-DCF than the baseline CM system, indicating a better overall spoofing detection performance.

In order to further prove effectiveness of the proposed approach, the dimension-reduced feature embedding of the baseline CM and proposed CM are visualized in Fig. 6. We utilize t-distributed Stochastic Neighbor Embedding (t-SNE) [120] to visualize feature embedding for the evaluation partition on ASVspoof 2019 LA track. As shown in Fig. 6, the proposed CM system can distinguish genuine speech and spoof attacks better than the baseline CM system with fewer misclassified spoofed samples, which indicates a better generalization ability to unseen spoofing attacks is present in the proposed CM system.

E. PERFORMANCE COMPARISON AGAINST EXISTING SYSTEMS

As illustrated in Tab. 5, a comparison of system performance between our proposed CM system and competing single state-of-the-art systems is also presented. The classical machine-learning-based method uses a common GMM back-end with LFCC as the front-end, which shows satisfying classification performance [19]. Comprehensive results show that our introduced simple attention module outperforms alternative approaches from previous works such as Convolutional Block Attention Module (CBAM) [121]; Squeeze-and-Excitation (SE) [122], [123]; and Dual attention module with pooling and convolution operations [121]. The WAAM loss employed in this work inherited the merits of cross-entropy loss and one-class softmax loss in [57]. Subsequently, the system trained with WAAM loss for binary classification outperforms the OC-Softmax, and AM-Softmax losses proposed in [57]. Both RawGAT-ST system [90] and Raw PC-DARTS system [96] operate directly on the raw speech data, while the former system is based upon graph attention networks, the latter system suggests an interesting approach to learn the network architecture automatically. Recent works [124], [125], [126] employed pre-trained Wav2Ves 2.0 as the front-end to extract speech embedding which are already learned from another task [127], embeddings are then mapped to the latent feature via proposed networks. More recently, the Rawformer [128] is proposed to leverage positional-related local-global dependency for synthetic audio spoofing detection. Those approaches with pre-trained embedding extractor show good performance in low-resource and

TABLE 5. Performance on the ASVspoof 2019 LA evaluation partition in terms of min t-DCF and pooled EER for state-of-the-art systems and our proposed best system.

Architecture	Front-end	min t-DCF	EER (%)
Ours	SincNet	0.0277	0.87
Dual-Branch Network [132]	CQT,LFCC	0.021	0.80
WavLM+ [129]	HuBERT	N/A	0.23
SE-Rawformer [128]	SincNet	0.0184	0.59
SBNLCNN [126]	wav2vec 2.0	N/A	0.258
XLSR-ASP [125]	XLSR-53	0.0088	0.31
VIB [124]	wav2vec 2.0	0.0107	0.40
RawGAT-ST [90]	SincNet	0.0335	1.06
SENet [123]	FFT	0.0368	1.14
Raw PC-DARTS [96]	SincNet	0.0517	1.77
MCG-Res2Net50+CE [133]	CQT	0.0520	1.78
ResNet18-LMCL-FM [134]	LFB	0.0520	1.81
Res18-OC-Softmax [57]	LFCC	0.0590	2.19
SE-Res2Net50 [122]	CQT	0.0743	2.50
LCNN-Dual attention [121]	LFCC	0.0777	2.76
Res18-AM-Softmax [57]	LFCC	0.0820	3.26
GMM [19]	LFCC	0.0904	3.50
LCNN-4CBAM [121]	LFCC	0.0939	3.67

cross-dataset settings. An interesting research work has been done in [129], they found that the attacker could also benefit from self-supervised learning (SSL) models (i.e., wav2vec 2.0 [127], HuBERT [130], and WavLM [131]), thereby eliminating most of the benefits the defender gains from them. Overall, our proposed system delivers the competitive results of all systems evaluated on the ASVspoof 2019 LA track.

VI. CONCLUSION

This study has considered the formulation of an effective approach to improve robustness of synthetic audio spoofing detection. We employed the RawNet2-based encoder, equipped with a simple attention module for feature refinement, to strengthen the distinctive feature representation power. Subsequently, we extensively explored multiple loss functions and their fusion to calibrate an embedding space for enhanced generalization to unseen spoofing attacks. First, we put forward the weighted additive angular margin loss, which served to alleviate any data imbalance and refine the embedding distribution. Next, we proposed a meta-learning episodic optimization scheme to adaptively learn a shared metric space between unseen samples and known attacks. Next, we developed a disentangled adversarial learning via an auxiliary batch norm to leverage both original training data and corresponding adversarial examples to train networks. Finally, the best-performing system updates the network parameters according to an integrated learning objective. Performance evaluation on the ASVspoof 2019 LA dataset confirms that our proposed approach effectively improves robustness/accuracy for spoofing detection system operation, which delivers results in terms of a pooled EER of 0.87%, and a min t-DCF of 0.0277.

REFERENCES

[1] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 74–99, Nov. 2015.

[2] N. W. Evans, T. Kinnunen, and J. Yamagishi, "Spoofing and countermeasures for automatic speaker verification," in *Proc. ISCA Interspeech*, 2013, pp. 925–929.

[3] S. K. Ergünay, E. Khoury, A. Lazaridis, and S. Marcel, "On the vulnerability of speaker verification to realistic voice spoofing," in *Proc. IEEE 7th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Sep. 2015, pp. 1–6.

[4] M. R. Kamble, H. B. Sailor, H. A. Patil, and H. Li, "Advances in anti-spoofing: From the perspective of ASVspoof challenges," *APSIPA Trans. Signal Inf. Process.*, vol. 9, no. 1, p. e2, 2020.

[5] J. Galbally, S. Marcel, and J. Fierrez, "Biometric antispoofing methods: A survey in face recognition," *IEEE Access*, vol. 2, pp. 1530–1552, 2014.

[6] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge," in *Proc. Interspeech*, 2015, pp. 2037–2041.

[7] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," in *Proc. Interspeech*, Aug. 2017, pp. 2–6.

[8] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. H. Kinnunen, and K. A. Lee, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," in *Proc. INTERSPEECH*, Sep. 2019, pp. 1008–1012.

[9] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, and N. Evans, "ASVspoof 2021: Accelerating progress in spoofed and deepfake speech detection," in *Proc. ISCA Interspeech*, 2021, pp. 47–54.

[10] Z. Wu, S. Gao, E. S. Cling, and H. Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *Proc. Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA), Asia–Pacific*, Dec. 2014, pp. 1–5.

[11] T. Masuko, T. Hitotsumatsu, K. Tokuda, and T. Kobayashi, "On the security of hmm-based speaker verification systems against imposture using synthetic speech," in *Proc. Eurospeech*, 1999, pp. 1223–1226.

[12] B. L. Pellom and J. H. L. Hansen, "An experimental study of speaker verification sensitivity to computer voice-altered imposters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 2, 1999, pp. 837–840.

[13] 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 Lang. Recognit. Workshop (Odyssey)*, 2020, pp. 333–340.

[14] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," in *Proc. ISCA Interspeech*, 2015, pp. 2087–2091.

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

[16] J. Yang, R. K. Das, and H. Li, "Significance of subband features for synthetic speech detection," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2160–2170, 2020.

[17] S. Garg, S. Bhilare, and V. Kanhangad, "Subband analysis for performance improvement of replay attack detection in speaker verification systems," in *Proc. IEEE 5th Int. Conf. Identity, Secur., Behav. Anal. (ISBA)*, Jan. 2019, pp. 1–7.

[18] B. Chettri, T. Kinnunen, and E. Benetos, "Subband modeling for spoofing detection in automatic speaker verification," in *Proc. Speaker Lang. Recognit. Workshop (Odyssey)*, Nov. 2020, pp. 341–348.

[19] H. Tak, J. Patino, A. Nautsch, N. Evans, and M. Todisco, "Spoofing attack detection using the non-linear fusion of sub-band classifiers," in *Proc. ISCA Interspeech*, 2020, pp. 1106–1110.

[20] M. Hajibabaei and D. Dai, "Unified hypersphere embedding for speaker recognition," 2018, *arXiv:1807.08312*.

[21] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Comput. Speech Lang.*, vol. 60, Mar. 2020, Art. no. 101027.

[22] J.-W. Jung, H.-S. Heo, I. Yang, H.-J. Shim, and H.-J. Yu, "Avoiding speaker overfitting in end-to-end DNNs using raw waveform for text-independent speaker verification," in *Proc. ISCA Interspeech*, 2018, vol. 8, no. 12, pp. 23–24.

[23] M. Ravanelli and Y. Bengio, "Learning speaker representations with mutual information," in *Proc. ISCA Interspeech*, 2019, pp. 1153–1157.

[24] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2014, pp. 4052–4056.

[25] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5329–5333.

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

[27] Y. Jung, Y. Kim, H. Lim, Y. Choi, and H. Kim, "Spatial pyramid encoding with convex length normalization for text-independent speaker verification," in *Proc. Interspeech*, Sep. 2019, pp. 4030–4034.

[28] J. W. Jung, S. B. Kim, H. J. Shim, J. H. Kim, and H. J. Yu, "Improved RawNet with feature map scaling for text-independent speaker verification using raw waveforms," in *Proc. ISCA Interspeech*, 2020, pp. 3583–3587.

[29] 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*, Sep. 2019, pp. 1268–1272.

[30] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with SincNet," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 1021–1028.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[32] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. Neural Inf. Process. Syst. (NIPS) Workshop Deep Learn.*, 2014.

[33] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[34] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[35] L. Yang, R. Zhang, L. Li, and X. Xie, "SimAM: A simple, parameter-free attention module for convolutional neural networks," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 11863–11874.

[36] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," in *Proc. IEEE Spoken Lang. Tech. Workshop (SLT)*, 2016, pp. 171–178.

[37] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Proc. Interspeech*, Aug. 2017, pp. 1487–1491.

[38] M. Sang, H. Li, F. Liu, A. O. Arnold, and L. Wan, "Self-supervised speaker verification with simple Siamese network and self-supervised regularization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 6127–6131.

[39] M. Sang, W. Xia, and J. H. Hansen, "DEAAN: Disentangled embedding and adversarial adaptation network for robust speaker representation learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2021, pp. 6169–6173.

[40] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6738–6746.

[41] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 926–930, Jul. 2018.

[42] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4690–4699.

[43] M. Sang, W. Xia, and J. H. L. Hansen, "Open-set short utterance forensic speaker verification using teacher–student network with explicit inductive bias," 2020, *arXiv:2009.09556*.

[44] M. Sang and J. H. L. Hansen, "Multi-frequency information enhanced channel attention module for speaker representation learning," 2022, *arXiv:2207.04540*.

[45] M. Sang, Y. Zhao, G. Liu, J. H. Hansen, and J. Wu, "Improving transformer-based networks with locality for automatic speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.

[46] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Proc. Speaker Lang. Recognit. Workshop (Odyssey)*, 2018, pp. 74–81.

[47] T. Ko, Y. Chen, and Q. Li, "Prototypical networks for small footprint text-independent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6804–6808.

[48] S. M. Kye, Y. Jung, H. B. Lee, S. J. Hwang, and H. Kim, "Meta-learning for short utterance speaker recognition with imbalance length pairs," in *Proc. Interspeech*, 2020, pp. 2982–2986.

[49] J. Wang, K.-C. Wang, M. T. Law, F. Rudzicz, and M. Brudno, "Centroid-based deep metric learning for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 3652–3656.

[50] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. ICLR*, 2017, pp. 1–17.

[51] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[52] T. Pang, X. Yang, Y. Dong, K. Xu, J. Zhu, and H. Su, "Boosting adversarial training with hypersphere embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 7779–7792.

[53] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 1–7.

[54] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "There is no free lunch in adversarial robustness (but there are unexpected benefits)," 2018, *arXiv:1805.12152*.

[55] D. Yin, R. G. Lopes, J. Shlens, E. D. Cubuk, and J. Gilmer, "A Fourier perspective on model robustness in computer vision," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 13276–13286.

[56] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 819–828.

[57] Y. Zhang, F. Jiang, and Z. Duan, "One-class learning towards synthetic voice spoofing detection," *IEEE Signal Process. Lett.*, vol. 28, pp. 937–941, 2021.

[58] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[59] X. Wang, S. Sun, C. Shan, J. Hou, L. Xie, S. Li, and X. Lei, "Adversarial examples for improving end-to-end attention-based small-footprint keyword spotting," in *Proc. IEEE Interface Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 6366–6370.

[60] P. L. De Leon, B. Stewart, and J. Yamagishi, "Synthetic speech discrimination using pitch pattern statistics derived from image analysis," in *Proc. ISCA Interspeech*, 2012, pp. 370–373.

[61] M. Todisco, H. Delgado, and N. Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Comput. Speech Lang.*, vol. 45, pp. 516–535, Sep. 2017.

[62] M. J. Alam, P. Kenny, G. Bhattacharya, and T. Stafylakis, "Development of CRIM system for the automatic speaker verification spoofing and countermeasures challenge 2015," in *Proc. ISCA Interspeech*, 2015, pp. 2072–2076.

[63] Z. Wu, X. Xiao, E. S. Chng, and H. Li, "Synthetic speech detection using temporal modulation feature," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 7234–7238.

[64] P. L. De Leon, I. Hernaez, I. Saratxaga, M. Pucher, and J. Yamagishi, "Detection of synthetic speech for the problem of imposture," in *Proc. IEEE Interface Conf. Acoust., Speech Signal Process. (ICASSP)*, 2011, pp. 4844–4847.

[65] I. Saratxaga, J. Sanchez, Z. Wu, I. Hernaez, and E. Navas, "Synthetic speech detection using phase information," *Speech Commun.*, vol. 81, pp. 30–41, Jul. 2016.

[66] Y. Liu, Y. Tian, L. He, J. Liu, and M. T. Johnson, "Simultaneous utilization of spectral magnitude and phase information to extract super-vectors for speaker verification anti-spoofing," in *Proc. ISCA Interspeech*, 2015, pp. 2082–2086.

[67] F. Alegre, R. Vipperla, A. Amehraye, and N. Evans, "A new speaker verification spoofing countermeasure based on local binary patterns," in *Proc. ISCA Interspeech*, 2013, pp. 940–944.

[68] Z. Wu, E. S. Chng, and H. Li, "Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition," in *Proc. ISCA Interspeech*, 2012, pp. 1700–1703.

[69] M. R. Kamble and H. A. Patil, "Novel energy separation based instantaneous frequency features for spoof speech detection," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2017, pp. 106–110.

[70] T. B. Patel and H. A. Patil, "Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech," in *Proc. ISCA Interspeech*, 2015, pp. 2062–2066.

[71] F. Zheng, G. Zhang, and Z. Song, "Comparison of different implementations of MFCC," *J. Comput. Sci. Technol.*, vol. 16, no. 6, pp. 582–589, Nov. 2001.

[72] X. Xiao, X. Tian, S. Du, H. Xu, E. Chng, and H. Li, "Spoofing speech detection using high dimensional magnitude and phase features: The NTU approach for ASVspoof 2015 challenge," in *Proc. ISCA Interspeech*, 2015, pp. 2052–2056.

[73] J.-C. Yang, C. You, and Q. He, "Feature with complementarity of statistics and principal information for spoofing detection," in *Proc. ISCA Interspeech*, 2018, pp. 651–655.

[74] K. Sriskandaraja, V. Sethu, E. Ambikairajah, and H. Li, "Front-end for antispoofing countermeasures in speaker verification: Scattering spectral decomposition," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 4, pp. 632–643, Jun. 2017.

[75] M. Alzantot, Z. Wang, and M. B. Srivastava, "Deep residual neural networks for audio spoofing detection," in *Proc. ISCA Interspeech*, 2019, pp. 1078–1082.

[76] C.-I. Lai, N. Chen, J. Villalba, and N. Dehak, "ASSERT: Anti-spoofing with squeeze-excitation and residual networks," in *Proc. INTERSPEECH*, Sep. 2019, pp. 1013–1017.

[77] J. Villalba, A. Miguel, A. Ortega, and E. Lleida, "Spoofing detection with DNN and one-class SVM for the ASVspoof 2015 challenge," in *Proc. ISCA Interspeech*, 2015, pp. 2067–2071.

[78] H. Yu, Z.-H. Tan, Z. Ma, R. Martin, and J. Guo, "Spoofing detection in automatic speaker verification systems using DNN classifiers and dynamic acoustic features," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4633–4644, Oct. 2018.

[79] N. Chen, Y. Qian, H. Dinkel, B. Chen, and K. Yu, "Robust deep feature for spoofing detection—The SJTU system for ASVspoof 2015 challenge," in *Proc. ISCA Interspeech*, 2015, pp. 2097–2101.

[80] M. J. Alam, P. Kenny, V. Gupta, and T. Stafylakis, "Spoofing detection on the ASVspoof2015 challenge corpus employing deep neural networks," in *Proc. Speaker Lang. Recognit. Workshop (Odyssey)*, 2016, pp. 270–276.

[81] A. Gomez-Alanis, A. M. Peinado, J. A. Gonzalez, and A. M. Gomez, "A light convolutional GRU-RNN deep feature extractor for ASV spoofing detection," in *Proc. Interspeech*, Sep. 2019, pp. 1068–1072.

[82] Y. Qian, N. Chen, and K. Yu, "Deep features for automatic spoofing detection," *Speech Commun.*, vol. 85, pp. 43–52, Dec. 2016.

[83] Z. Zhang, N. Cummins, and B. Schuller, "Advanced data exploitation in speech analysis: An overview," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 107–129, Jul. 2017.

[84] T. Heittola, E. Çakır, and T. Virtanen, "The machine learning approach for analysis of sound scenes and events," in *Proc. Comput. Anal. Sound Scenes Events*, 2018, pp. 13–40.

[85] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2721–2725.

[86] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, and E. Gonina, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE Interface Conf. Acoust., Speech Signal Process. (ICASSP)*, 2018, pp. 4774–4778.

[87] H. Muckenhirn, M. Magimai-Doss, and S. Marcel, "End-to-end convolutional neural network-based voice presentation attack detection," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 335–341.

[88] H. Dinkel, N. Chen, Y. Qian, and K. Yu, "End-to-end spoofing detection with raw waveform CLDNNS," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 4860–4864.

[89] H. Dinkel, Y. Qian, and K. Yu, "Investigating raw wave deep neural networks for end-to-end speaker spoofing detection," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 26, no. 11, pp. 2002–2014, Nov. 2018.

[90] H. Tak, J.-W. Jung, J. Patino, M. Kamble, M. Todisco, and N. Evans, "End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection," in *Proc. Autom. Speaker Verification Spoofing Countermeasures Challenge*, 2021, pp. 1–8.

[91] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, vol. 1050, no. 20, 2017, pp. 48539–48550.

[92] 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 *Proc. IEEE Interfaces Conf. Acoust., Speech Signal Process. (ICASSP)*, 2022, pp. 6367–6371.

[93] W. Ge, M. Panariello, J. Patino, M. Todisco, and N. Evans, "Partially-connected differentiable architecture search for deepfake and spoofing detection," in *Proc. ISCA Interspeech*, 2021, pp. 4319–4323.

[94] Y. Xu, L. Xie, X. Zhang, X. Chen, G.-J. Qi, Q. Tian, and H. Xiong, "PC-darts: Partial channel connections for memory-efficient architecture search," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–13.

[95] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–13.

[96] W. Ge, J. Patino, M. Todisco, and N. Evans, "Raw differentiable architecture search for speech deepfake and spoofing detection," in *Proc. Autom. Speaker Verification Spoofing Countermeasures Challenge*, 2021, pp. 22–28.

[97] H. Tak, J.-W. Jung, J. Patino, M. Todisco, and N. Evans, "Graph attention networks for anti-spoofing," in *Proc. ISCA Interspeech*, 2021, pp. 2356–2360.

[98] G. Hua, A. B. J. Teoh, and H. Zhang, "Towards end-to-end synthetic speech detection," *IEEE Signal Process. Lett.*, vol. 28, pp. 1265–1269, 2021.

[99] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with RawNet2," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2021, pp. 6369–6373.

[100] T. Quatieri, *Discrete-time Speech Signal Processing: Principles and Practice*. London, U.K.: Pearson Education, 2006.

[101] J. R. Deller, J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals*. Piscataway, NJ, USA: IEEE Press, 2000.

[102] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 630–645.

[103] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[104] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 972–981.

[105] Z. Yang, L. Zhu, Y. Wu, and Y. Yang, "Gated channel transformation for visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11791–11800.

[106] S. Yadav and A. Rai, "Frequency and temporal convolutional attention for text-independent speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6794–6798.

[107] X. Lin, L. Ma, W. Liu, and S.-F. Chang, "Context-gated convolution," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 701–718.

[108] X. Qin, N. Li, C. Weng, D. Su, and M. Li, "Simple attention module based speaker verification with iterative noisy label detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 6722–6726.

[109] B. S. Webb, N. T. Dhruv, S. G. Solomon, C. Tailby, and P. Lennie, "Early and late mechanisms of surround suppression in striate cortex of macaque," *J. Neurosci.*, vol. 25, no. 50, pp. 11666–11675, Dec. 2005.

[110] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, and K. A. Lee, "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Comput. Speech Lang.*, vol. 64, Nov. 2020, Art. no. 101114.
[111] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–11.
[112] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," 2017, *arXiv:1704.03453*.
[113] T. Zhang and Z. Zhu, "Interpreting adversarially trained convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 7502–7511.
[114] C. Xie, Y. Wu, L. V. D. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 501–509.
[115] A. Nautsch, X. Wang, N. Evans, T. H. 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 Trans. Biometrics, Behav., Identity Sci.*, vol. 3, no. 2, pp. 252–265, Sep. 2021.
[116] T. Kinnunen, K. A. Lee, H. Delgado, N. Evans, M. Todisco, M. Sahidullah, J. 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 Lang. Recognit. Workshop (Speaker Lang. Recognit. Workshop (Odyssey)*, 2018, pp. 312–319.
[117] T. Kinnunen, H. Delgado, N. Evans, K. A. Lee, V. Vestman, A. Nautsch, M. Todisco, X. Wang, M. Sahidullah, J. Yamagishi, and D. A. Reynolds, "Tandem assessment of spoofing countermeasures and automatic speaker verification: Fundamentals," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 28, pp. 2195–2210, 2020.
[118] X. Wang and J. Yamagishi, "A comparative study on recent neural spoofing countermeasures for synthetic speech detection," in *Proc. ISCA Interspeech*, 2021, pp. 4259–4263.
[119] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–12.
[120] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.
[121] X. Ma, T. Liang, S. Zhang, S. Huang, and L. He, "Improved lightCNN with attention modules for ASV spoofing detection," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2021, pp. 1–6.
[122] X. Li, N. Li, C. Weng, X. Liu, D. Su, D. Yu, and H. Meng, "Replay and synthetic speech detection with Res2Net architecture," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6354–6358.
[123] Y. Zhang, W. Wang, and P. Zhang, "The effect of silence and dual-band fusion in anti-spoofing system," in *Proc. ISCA Interspeech*, 2021, pp. 4279–4283.
[124] Y. Eom, Y. Lee, J. S. Um, and H. Kim, "Anti-spoofing using transfer learning with variational information bottleneck," 2022, *arXiv:2204.01387*.
[125] J. Woo Lee, E. Kim, J. Koo, and K. Lee, "Representation selective self-distillation and wav2vec 2.0 feature exploration for spoof-aware speaker verification," 2022, *arXiv:2204.02639*.
[126] H. Lin, Y. Ai, and Z. Ling, "A light CNN with split batch normalization for spoofed speech detection using data augmentation," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2022, pp. 1684–1689.
[127] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12449–12460.
[128] X. Liu, M. Liu, L. Wang, K. A. Lee, H. Zhang, and J. Dang, "Leveraging positional-related local–global dependency for synthetic speech detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.
[129] A. Ito and S. Horiguchi, "Spoofing attacker also benefits from self- supervised pretrained model," in *Proc. ISCA Interspeech*, 2023, pp. 5346–5350.
[130] W.-N. Hsu, B. Bolte, Y. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 29, pp. 3451–3460, 2021.
[131] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1505–1518, Oct. 2022.
[132] K. Ma, Y. Feng, B. Chen, and G. Zhao, "End-to-end dual-branch network towards synthetic speech detection," *IEEE Signal Process. Lett.*, vol. 30, pp. 359–363, 2023.
[133] X. Li, X. Wu, H. Lu, X. Liu, and H. Meng, "Channel-wise gated Res2Net: Towards robust detection of synthetic speech attacks," in *Proc. ISCA Interspeech*, 2021, pp. 4314–4318.
[134] T. Chen, A. Kumar, P. Nagarsheth, G. Sivaraman, and E. Khoury, "Generalization of audio deepfake detection," in *Proc. Speaker Lang. Recognit. Workshop (Odyssey)*, Nov. 2020, pp. 132–137.

ZHENYU WANG received the B.S. degree majoring in digital media technology from Hangzhou Dianzi University, Hangzhou, China, in 2015, and the M.S. degree in engineering computer system architecture from Beijing Language and Culture University, Beijing, China, in 2019. He is currently pursuing the Ph.D. degree in computer engineering with The University of Texas at Dallas (UTD), Richardson, TX, USA. He works with Prof. John H. L. Hansen with the Center for Robust Speech Systems. Since then, he has been a Graduate Research Assistant with the Center for Robust Speech Systems (CRSS), UTD. He has authored over ten journals and conference papers in the field of speech processing and language technology. His research interests include mispronunciation verification, computer-assisted language learning, forensic audio analysis and model adaptation for open-set speaker recognition system, representation learning used for acoustic modeling, keyword spotting, anti-spoofing, audio generation, and LLM.

JOHN H. L. HANSEN (Fellow, IEEE) received the B.S.E.E. degree from the College of Engineering, Rutgers University, and the M.S. and Ph.D. degrees in electrical engineering from Georgia Institute of Technology. He joined University of Texas at Dallas (UTDallas), Erik Jonsson School of Engineering, in 2005, where he presently serves as Jonsson School Associate Dean for Research, Professor of Electrical and Computer Engineering (ECE) and holds the Distinguished University Chair in Telecommunications Engineering. He previously served as Department Head of ECE (2005–2012), growing UTDallas to the 8th largest EE program from ASEE rankings in terms of degrees awarded. He also holds a joint appointment as a Professor with the School of Behavioral and Brain Sciences (Speech & Hearing). At UTDallas, he established the Center for Robust Speech Systems (CRSS). Previously, he served as Department Chair and Professor of Department of Speech, Language and Hearing Sciences (SLHS), and Professor with the Department of Electrical and Computer Engineering, University of Colorado Boulder (1998–2005), where he co-founded and served as Associate Director of the Center for Spoken Language Research. In 1988, he established the Robust Speech Processing Laboratory (RSPL) at Duke University Department of Electrical Engineering, and continues to direct research activities in CRSS at UTDallas. His research interests span the areas of digital speech processing, analysis and modeling of speech and speaker traits, speech enhancement, machine learning, applications in diarization and spoken document retrieval. He has been named IEEE Fellow (2007) for contributions in "Robust Speech Recognition in Stress and Noise," International Speech Communication Association (ISCA) Fellow (2010) for contributions on "research for speech processing of signals under adverse conditions," and received The Acoustical Society of America's 25 Year Award (2010) – in recognition of his contributions, leadership, service, and membership to the Acoustical Society

of America. He recently completed his second term as President of ISCA and member of ISCA Board. He also continues to serve as Vice-Chair on U.S. Office of Scientific Advisory Committees (OSAC) for OSAC-Speaker in the voice forensics domain (2015–2024). Previously he served as IEEE Technical Committee (TC) Chair and Member of IEEE Signal Processing Society: Speech & Language Processing Technical Committee (SLTC) (2005–2008; 2010–2014; elected IEEE SLTC Chair 2011–2014), and elected ISCA Distinguished Lecturer (2011/2012). He has served as member of IEEE Signal Processing Society Educational Technical Committee (2005–2010); Technical Advisor to the U.S. Delegate for NATO (IST/TG-01); IEEE Signal Processing Society Distinguished Lecturer (2005/2006), an Associate Editor for IEEE TRANSACTIONS SPEECH AND AUDIO PROCESSING (1992–1999), an Associate Editor for IEEE SIGNAL PROCESSING LETTERS (1998–2000), Editorial Board Member for IEEE *Signal Processing Magazine* (2001–2003); and guest editor (October 1994) for special issue on Robust Speech Recognition for IEEE TRANSACTION SPEECH AND AUDIO PROCESSING. He has served on Speech Communications Technical Committee for Acoustical Society of America (2000–2003). He has supervised 107 Ph.D./M.S. thesis candidates (64 PhD, 43 M.S./M.A.), was recipient of 2020 University of Texas – Dallas Provost's Award for Graduate Student Mentorship, 2005 University of Colorado Teacher Recognition Award as voted on by the student body, author/co-author of +912 journal and conference papers including 14 textbooks in the field of speech processing and language technology, coauthor of textbook Discrete-Time Processing of Speech Signals, (IEEE Press, 2000), and lead author of the report "The Impact of Speech Under 'Stress' on Military Speech Technology," (NATO RTO-TR-10, 2000). He also organized and served as General Chair for ISCA INTERSPEECH-2002, September 2016–2020, 2002; General Co-Chair for ISCA INTERSPEECH-2022, September 2018–2022, 2022; Co-Organizer and Technical Program Chair for IEEE ICASSP-2010, March 2015–2019, 2010, Co-Chair and Organizer for IEEE SLT-2014, December 2007–2010, 2014. He also served as Co-Organizer and Technical Program Chair for IEEE ICASSP-2024 (5950 CP submissions; 4350 attendees), Seoul, South Korea, 2024. In 2022, he was awarded the IEEE Signal Processing Society Leo L. Beranek Meritorious Service Award: "for exemplary service to and leadership in the Signal Processing Society." He was awarded the honorary degree "Doctor Technices Honoris Causa" from Aalborg University (Aalborg, DK) (2016) in recognition of his contributions to speech signal processing and speech/language/hearing science.