Vocoder Detection of Spoofing Speech Based on GAN Fingerprints and Domain Generalization

FAN LI, YANXIANG CHEN, HAIYANG LIU, ZUXING ZHAO, and YUANZHI YAO, Hefei University of Technology, China
XIN LIAO, Hunan University, China

As an important part of the text-to-speech (TTS) system, vocoders convert acoustic features into speech waveforms. The difference in vocoders is key to producing different types of forged speech in the TTS system. With the rapid development of general adversarial networks (GANs), an increasing number of GAN vocoders have been proposed. Detectors often encounter vocoders of unknown types, which leads to a decline in the generalization performance of models. However, existing studies lack research on detection generalization based on GAN vocoders. To solve this problem, this study proposes vocoder detection of spoofed speech based on GAN fingerprints and domain generalization. The framework can widen the distance between real speech and forged speech in feature space, improving the detection model's performance. Specifically, we utilize a fingerprint extractor based on an autoencoder to extract GAN fingerprints from vocoders. We then weight them to the forged speech for subsequent classification to learn the forged speech features with high differentiation. Subsequently, domain generalization is used to further improve the generalization ability of the model for unseen forgery types. We achieve domain generalization using domain-adversarial learning and asymmetric triplet loss to learn a better generalized feature space in which real speech is compact and forged speech synthesized by different vocoders is dispersed. Finally, to optimize the training process, curriculum learning is used to dynamically adjust the contributions of the samples with different difficulties in the training process. Experimental results show that the proposed method achieves the most advanced detection results among four GAN vocoders. The code is available at https://github.com/multimedia-infomation-security/GAN-Vocoder-detection.

CCS Concepts: • **Security and privacy → Human and societal aspects of security and privacy;**

Additional Key Words and Phrases: Speech forgery, vocoder, GAN fingerprint, domain generalization, curriculum learning

ACM Reference format:
Fan Li, Yanxiang Chen, Haiyang Liu, Zuxing Zhao, Yuanzhi Yao, and Xin Liao. 2024. Vocoder Detection of Spoofing Speech Based on GAN Fingerprints and Domain Generalization. *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 6, Article 157 (March 2024), 20 pages.
https://doi.org/10.1145/3630751

This work was supported by the National Natural Science Foundation of China under Grants 61972127, 61972142, U22A2030.

Authors' addresses: F. Li, Y. Chen (Corresponding author), H. Liu, Z. Zhao, and Y. Yao, Hefei University of Technology, Hefei, Anhui, 230601, China; e-mails: 2021111020@mail.hfut.edu.cn, chenyx@hfut.edu.cn, 1405357533@qq.com, 2021171120@mail.hfut.edu.cn, yaoyz@hfut.edu.cn; X. Liao, Hunan University, Changsha, Hunan, 410082, China; e-mail: xinliao@hnu.edu.cn.

1551-6857/2024/03-ART157 $15.00
https://doi.org/10.1145/3630751

1 INTRODUCTION

Recent advances in deep learning have helped speech forgery technology mature. One of the key technologies of speech forgery is speech synthesis (**text-to-speech (TTS)**) [13, 21, 32, 34]. TTS can synthesize fake speech similar to natural speech. However, when used by criminals to spread false politics, create public opinions, and other undesirable purposes, it seriously threatens global economic, political, and social security. Therefore, it is important to develop a method to detect forged speech.

This issue has been previously addressed by several studies. Currently, synthetic speech detection systems primarily consist of two parts: front-end speech feature extraction and backend feature representation learning. For the front-end feature, the spectral feature is widely used in forged speech detection, such as **Mel-frequency cepstral coefficients (MFCCs)**, **linear frequency cepstral coefficients (LFCCs)**, and **cochlear filter cepstral coefficients (CFCCs)** [22, 23, 33]. Traditional backend networks are mostly feature-based machine learning methods that directly classify input manual features, such as the **Gaussian mixture model (GMM)** and **support vector machine (SVM)** [27]. With the development of deep learning technology, extensive **convolutional neural network (CNN)**-based approaches have been developed [15, 26, 41]. With the help of the learning ability of CNN, the front-end features of the input are learned to obtain the semantic information of high-dimensional features. However, although existing detection methods have achieved good detection performance, most of them rely on specific datasets or forgery methods and can only show good performance on known forgery methods while having a poor detection effect on unknown forgeries. Therefore, the generalization problem has received increasing attention.

Recent studies [25, 38] have investigated vocoder artifacts for forgery speech detection generalization. Vocoders are one of the core components of speech synthesis and are responsible for synthesizing spectrograms into speech waveforms. Because vocoders are commonly the last step in speech synthesis models, it is improbable for authentic speech to undergo processing using vocoders. Consequently, the artifacts produced by vocoders can serve as crucial cues for distinguishing between genuine and synthetic speech.

Yan et al. [38] proposed mining vocoder fingerprints using multi-classification tasks. Sun et al. [25] attempted to develop a generalization ability by digging vocoder artifacts from multitask learning. They performed a vocoder recognition task as an auxiliary task to constrain feature extractors to focus on vocoder artifacts and provide discriminant features for the final binary classifier. However, they can only extract the vocoder artifacts with the help of auxiliary tasks and cannot fully mine forgery information hidden in vocoders. In addition, they can only generalize to the vocoders that have been seen during training but still exhibit poorer performance on unseen vocoders. In real scenarios, vocoders are updated quickly with the rapid development of GANs, particularly GAN-based vocoders. It is important to develop more generalized models that perform well for unseen vocoders. To address this problem, we focus on GAN fingerprints and domain generalization to improve the generalization ability of forged speech detection.

We first introduce the concept of GAN fingerprints in forged images [18, 39], which represents the distribution of the training dataset, network architecture, and optimization strategy unique to the GAN model. Marra et al. [18] demonstrated the existence of a GAN fingerprint in forged images. Inspired by this, we believe that forged speech synthesized by a GAN vocoder also contains fingerprint information associated with the GAN model. Thus, effectively extracting the GAN fingerprints of forged speech can help better discriminate between real and synthetic speech and improve the generalization ability of forged speech detection. In contrast to [25, 38], we directly extract GAN fingerprints from the GAN vocoder itself before training. We do not rely on the help of

other tasks, which reduced the training complexity while extracting high quality vocoder artifacts for our detection.

To highlight the distinct GAN fingerprints left by vocoders in synthetic speech, we directly synthesize our forged dataset using vocoders from real speech, ignoring the interference from other factors. To obtain the GAN fingerprints from the forged speech, we first reconstruct the forged speech based on an autoencoder. The autoencoder is trained only on real speech, forged speech reconstructed by it will lack the forged information synthesized by the GAN vocoder, because the autoencoder has only been exposed to real speech during the training stage and lacks prior knowledge about forged speech. Therefore, GAN fingerprints can be represented as the residual of forged speech and reconstructed-forged speech.

To further improve the generalization ability of the model for unseen forgery types, we use the **domain generalization (DG)** method. We view forgery speech synthesized by different GAN vocoders as different domains. Existing DG methods [14, 36] improve the generalization ability by extracting common distinguishing features from different data domains. However, owing to the joint of GAN fingerprints, there is a large distribution difference between forged speech in different domains, and it is difficult to find a compact and generalized feature space for forgery speech. Instead, there is little difference between the real speeches of different domains because they do not contain the process of vocoders, and it is easier for them to learn a compact space.

In our work, we achieved DG by domain-adversarial learning and asymmetric triples loss to address the problem above and form a better feature space, which can generalize well to the unseen domains. We first use domain-adversarial learning for real speech to make it indistinguishable among the domains. In addition, as mentioned above, the fake speech of different domains is rather diverse in feature space, it is easier to form a dispersed feature space for fake speech rather than a compact one. Therefore, to generate a better generalized feature space, we consider achieving asymmetric optimization between real and fake speeches. We label the real speech of all domains as one class and the fake speech of different domains as different classes and optimize the asymmetric triples loss according to the class. In this manner, we can learn a better generalized feature space in which the feature distribution of real speech is compact, but the distribution of forgery speech in different domains is separate. In the final classification loss, curriculum learning [1] is combined to dynamically adjust the contribution of samples of different difficulty levels in the training stage so that the model can easily find a better local optimum and speed up the training. The main contributions of this study are as follows:

- We focus on GAN-based vocoder artifacts for forged speech detection. To the best of our knowledge, this is the first study to directly extract GAN fingerprints from GAN vocoder itself as vocoder artifacts instead of mining them with the help of other tasks, which makes the real and forged speeches more dispersed in the feature space.
- The joint of GAN fingerprints causes a large distribution difference between the forged speech synthesized by different vocoders in feature space. To learn a better generalized feature space, we achieve DG by domain-adversarial learning for real speech and asymmetric triples loss, in which the feature distribution of real speech is compact, but the distribution of forgery speech in different domains is separated.
- Extensive experiments are conducted on four vocoder detection datasets constructed by us, and the experimental results demonstrate the effectiveness of the proposed method. Compared with existing works, this strategy can significantly improve the detection generalization of forged speech based on a GAN vocoder.

2 RELATED WORK

2.1 Speech Forgery Detection

In this section, two important components of speech forgery detection are reviewed: front-end feature extraction and backend classification models.

Front-end feature: The purpose of front-end feature extraction is to extract distinctive speech features to input to the backend for judgment. At present, the most widely used is the spectrum features. The constant-Q cepstral coefficients were proposed in [28]. Han et al. [5] proposed the use of the MFCC as a front-end feature for speech recognition. Todisco et al. [29] compared the performance of the MFCC features with that of the LFCC features when a GMM backend was used.

Backend model: The backend model classifies the features extracted from the front-end to distinguish the authenticity of speech. Most traditional methods are based on manual feature learning and directly classify and recognize manual input features. Kumar et al. [8] proposed a decision method for selecting specific frames to calculate the logarithmic likelihood ratio based on traditional GMM classifiers to reduce the influence of unmodified voiceless frames on decision scores in speech conversion. Lei et al. [12] proposed a twin convolutional network that comprehensively considered the GMM score and local relations between frames to improve the classification. With the development of deep learning, the backend of the latest forged speech detection systems is primarily based on **deep neural network (DNN)** classification. Chen et al. [2] proposed a forged speech- detection framework based on ResNet. Lavrentyeva et al. [10] proposed a **lightweight convolutional neural network (LCNN)**, which is the most widely used deep network model for speech forgery detection.

2.2 Detection Generalization

Recently, the generalization performance of forged-speech detection models has been studied from the perspectives of speech feature extraction, model optimization, and training strategies. Zhang et al. [40] improved generalization ability by adopting a one-class unsupervised training strategy. This strategy uses only real speech training to distinguish between real and fake speech, such that the model only needs to learn the real audio data distribution. Tzeng et al. [30] optimized a forged-speech recognition model using a **domain adaptation (DA)** method to improve the ability of the model to detect speech generated by unknown algorithms. Shao et al. [24] used DG to learn a generalized feature space. They achieved DG using adversarial learning to train multiple feature extractors. Based on the idea in [24], this study conducted domain-adversarial learning of real speech, gathered the feature distribution space of real speech, and widened the distance between fake and real speeches. Sun et al. [25] attempted to develop a generalization ability by digging vocoder artifacts in forged speech through multitask learning and adopting RawNet2 as a backend network. Ma et al. [17] proposed an end-to-end Dual-Branch system and introduced multitask learning into a Dual-Branch Network to learn the common forgery features from different speech forgery types. [17, 25] focused only on finding forgery cues during backend feature extraction, ignoring excavating discriminative forgery features left by the forged speech itself. In addition, these two works can only be generalized to vocoders that have been seen during training but still exhibit poorer performance on unseen vocoders.

2.3 GAN Fingerprints

Marra et al. [18] demonstrated the existence of GAN fingerprints that can be used for reliable forensic analysis. It was demonstrated that each GAN leaves its specific fingerprint in an image generated by extracting noise residuals. Yu et al. [39] proposed a multi-class classification method based on fingerprint matching. Each GAN model was believed to have a unique fingerprint under

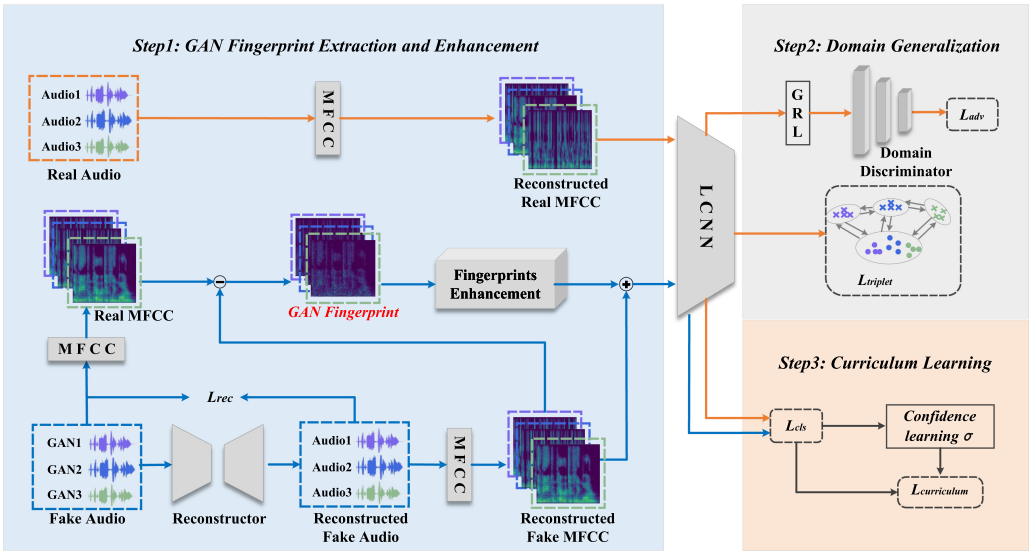


Fig. 1. Method overview. The figure shows the three modules of the method. The blue lines exhibit the flow of fake speech, and the orange lines show the flow of real speech. Step 1: Fingerprint extraction and enhancement. The forged voice is synthesized by the GAN vocoder, and the reconstructed forged voice is obtained by the reconstruction module. The residual between the forged voice and the reconstructed voice is the GAN fingerprint, which is enhanced by the fingerprint enhancement module. The enhanced GAN fingerprint is attached to the reconstructed fake voice and sent to the LCNN network. Step 2: Domain Generalization. The DG module includes two parts: domain-adversarial loss and triplet loss. The domain-adversarial learning is used to gather the feature space of real speech, and the triplet loss is used to gather real speech and make forged speech feature dispersed among domains. Step 3: Curriculum learning. The learning strategy of the classification model is dynamically adjusted through curriculum learning, and the contribution of different samples is adjusted by the confidence score.

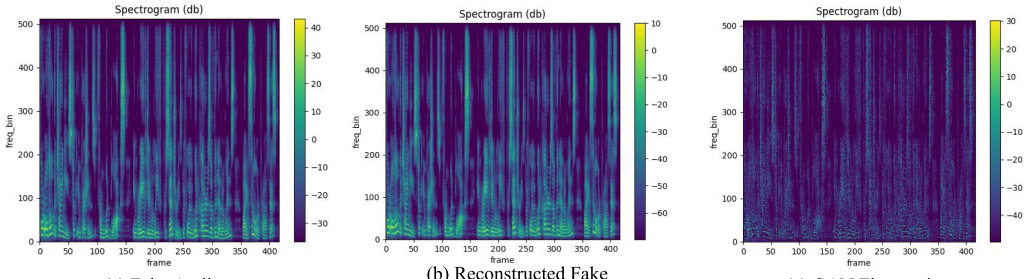
the influence of training data, network structure, loss function, parameter setting, and other factors. Wang et al. [31] performed data enhancement in a GAN fingerprint domain to improve the extensibility of an image detector generated by a GAN for forged image detection.

2.4 Curriculum Learning

Curriculum learning is a technique used to improve model performance and generalization ability. Its idea is that simple samples should precede difficult samples during training. For instance, Graves et al. [4] improved the learning efficiency to a maximum by automatically selecting the path or method followed by neural networks in the curriculum. Ma et al. [16] proposed a **curriculum comparison fake detection model (CCFD)** for fake news detection that automatically selects and trains negative samples of different difficulty levels in different training stages. Castells et al. [1] proposed a simple generic loss function that can be applied to various losses and tasks without altering the learning process.

3 METHOD

In this study, a forged-speech detection framework is proposed based on GAN fingerprints and DG, as shown in Figure 1. The framework consists of three modules: First, the GAN fingerprint artifacts left by vocoders are extracted through an autoencoder. Because each GAN vocoder contains



(a) Fake Audio (b) Reconstructed Fake Audio (c) GAN Fingerprint

Fig. 2. (a) Spectrogram of the GAN-forged speech, (b) spectrogram of the speech reconstructed from the GAN speech, and (c) spectrogram of the GAN fingerprint information, which is represented as the residual between the reconstructed speech (b) and original speech (a).

specific fingerprint information [18], it can be used to widen the difference between real speech and forged speech. For more details, please refer to Section 3.1. Second, to further improve the generalization ability of the model on unseen forgery types, we use the domain generalization method. Since we add GAN fingerprints in the front-end, it is harder to find a compact feature space for forgery speech. Instead, it is easier to learn a compact space for real speech because they do not contain the process of vocoders. Thus, we utilize domain-adversarial learning for real speech and add an asymmetric triples loss to learn a generalized feature space where the feature distribution of the real speech is compact but the distribution of forgery speech in different domains is separated. (See Section 3.2). Finally, the dynamic curriculum learning method [1] is used to add a confidence score ω based on the classification loss, which can adjust the contribution of samples of different difficulty levels to model training, that is, automatically reduce the importance of samples with large losses to improve the classification performance of the model, as shown in Section 3.3.

3.1 GAN Fingerprint Extraction and Enhancement

In this section, we aim to obtain the GAN fingerprints left by vocoders of the GAN-forged speech. Because the GAN fingerprint is only related to the GAN vocoders, to eliminate the interference of other factors, the text analysis and acoustic model modules of the TTS are not considered; that is, the speech is not generated from text. Instead, the forged speech is directly synthesized [38] from real speech, and an important GAN fingerprint is extracted from the GAN-synthesized speech.

3.1.1 GAN Fingerprints Extraction. According to [18], each GAN model has its own training data distribution, network architecture, and optimization strategy. Owing to the nonconvexity of the objective function and the instability of the adversarial balance between the generator and discriminator in GANs, the value of the model weight is sensitive to its random initialization and does not converge to the same value during each training. Hence, although two well-trained GAN models can perform equivalently, they produce different high-quality speech. This indicates the existence and uniqueness of GAN fingerprints. Because GAN fingerprints are found only in GAN-forged speech, they can be used to magnify the gap between the GAN-forged and real speech.

We utilize an autoencoder trained only on real speech to extract the GAN fingerprints. The autoencoder has a strong learning ability and can fully learn the characteristics of real speech. If an autoencoder trained entirely by real speech is used to reconstruct forged speech, the reconstructed forged speech will lack the forged information synthesized by the GAN vocoder because the autoencoder has only been exposed to real speech during the training stage and lacks prior knowledge about the forged speech. Therefore, we represent the GAN fingerprint as the residual

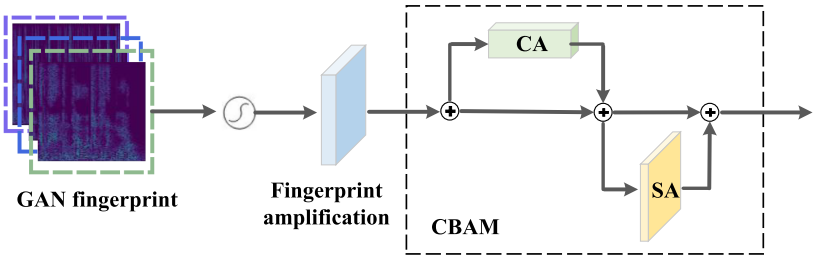


Fig. 3. GAN fingerprint enhancement module. First, the fingerprint is amplified through a sigmoid function and a 1×1 convolutional layer. Then, the CBAM module is used to adjust the amplification amplitude of the amplified fingerprint, focusing on the important information in the fingerprint.

of the forged speech and the reconstructed-forged speech, as shown in Figure 2(c). Specifically, we represent the autoencoder as E_{rec}, A_r denotes real speech, and A_f denotes fake speech, and we train E_{rec} on A_r by optimizing the following objectives:

$$\mathcal{L}_{rec} = \|E_{rec}(A_r) - A_r\|_2 \tag{1}$$

We used E_{rec} to reconstruct A_f, the reconstructed forgery speech is represented as $\hat{A}_f$; we then extracted the MFCC features of A_f and $\hat{A}_f$ respectively:

$$\hat{F}_f = MFCC(\hat{A}_f), F_f = MFCC(A_f) \tag{2}$$

The residuals of F_f and $\hat{F}_f$ represent GAN fingerprints:

$$F_{GAN} = F_f - \hat{F}_f \tag{3}$$

3.1.2 Fingerprint Feature Enhancement. The GAN fingerprint contains sufficient information related to each GAN, which helps to distinguish different GAN types. To enlarge the differentiation between real and forged speech, we enhance the fingerprint feature F_{GAN} in this study. The enhancement module consists of two parts, i.e., (1) a fingerprint amplification module and (2) a **convolutional block attention module (CBAM)**, as shown in Figure 3.

Fingerprint amplification: The amplification module consists of a sigmoid function and a 1×1 convolutional block. The original GAN fingerprint F_{GAN} is extremely weak and must be enhanced. First, a sigmoid function σ is used to amplify the weak fingerprint, and then a 1×1 convolutional block is used to control the amplifying amplitude of the GAN fingerprint. The amplified GAN fingerprint is superimposed to obtain the amplified fingerprint information F_{GAN_a}. The above process can be expressed as follows:

$$F_{GAN\_a} = Conv(\sigma(F_{GAN})) \tag{4}$$

CBAM module: According to the introduction in [35], the CBAM represents the attention mechanism module of the convolutional module, which is a combination of spatial and channel modules. The input feature F_{GAN_a} passes through a channel attention module and weighs the result to F_{GAN_a} to get F_{GAN_ca}.

$$F_{GAN\_ca} = \sigma(MLP(AvgPool(F_{GAN\_a})) + MLP(MaxPool(F_{GAN\_a}))) + F_{GAN\_a} \tag{5}$$

MLP indicates the multilayer perceptron, σ is a sigmoid function. *AvgPool* indicates average pooling and *MaxPool* indicates maximum pooling. After obtaining the result of channel attention, it passes through a spatial attention module and finally obtains the result F_{GAN_out} through weighting.

$$F_{GAN\_out} = \sigma(f^{7\times7}(AvgPool(F_{GAN\_ca}) + MaxPool(F_{GAN\_ca}))) + F_{GAN\_ca} \tag{6}$$

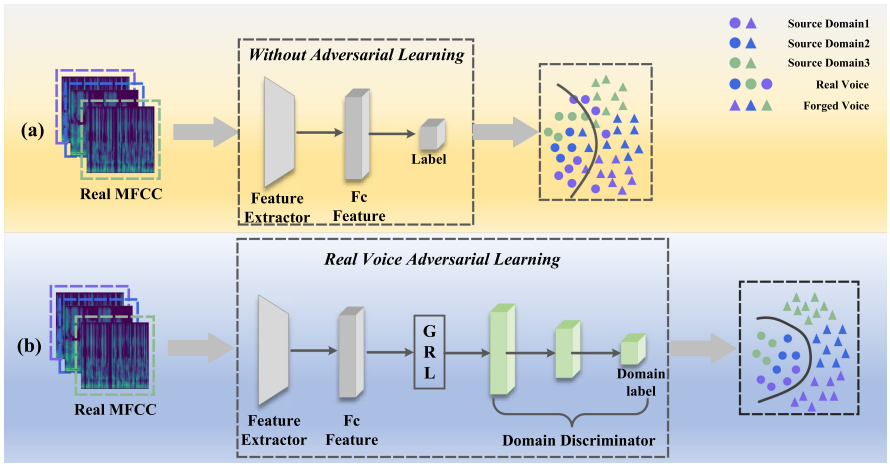


Fig. 4. Feature distribution without (a) and with (b) adversarial learning. The yellow block represents the general classification process without domain-adversarial learning. It can form the feature distribution, as shown in (a), which is roughly separable but not discriminating enough. The blue block represents the process of domain-adversarial learning on real speech, which forms the feature distribution as shown in (b). Unlike (a), (b) has a more concentrated feature distribution space of real speech.

σ is the sigmoid operation, and 7×7 is the size of the convolution kernel. Finally, the enhanced fingerprint information F_{GAN_out} is weighted to the reconstructed $\hat{F}_f$ to obtain the forged-speech MFCC feature F_{f_out}:

$$F_{f\_out} = F_{GAN\_out} + \hat{F}_f \tag{7}$$

3.2 Domain Generalization

We use the DG to further improve the generalization ability of the model for unseen forgery types. Because we add GAN fingerprints to the front-end feature, the distribution of the forged speech varies greatly in different domains. It is difficult to find a common and generalized feature space for forgery speech. Instead, there is little difference between the real speeches of different domains, as they are not processed by vocoders; therefore, it is easier for them to learn a compact space. Thus, the main idea of DG in our work is to learn a generalized feature space in which the feature distribution of real speech is compact, but the distribution of forgery speech in different domains is separated. To achieve DG, we utilize domain-adversarial learning and asymmetric triples loss. Domain-adversarial learning is used for real speech to form a compact feature space. Asymmetric triples loss is used to gather real speech and make forged speech feature dispersed among domains.

3.2.1 Domain-Adversarial Learning for Real Speech. Owing to the joint of the GAN fingerprint, it is difficult to find a generalized feature space suitable for all GAN-forged speech. However, it is relatively easy to gather common features for real speech. Thus, we conduct adversarial learning for real speech because real voices in different domains are close to one another in the feature space. Adversarial learning can help to learn domain-invariant features of real speech, thus learning a more compact feature space for them, as shown in Figure 4.

First, speech from N different domains is represented as:

$$A = (A_f, A_r), A \in D_1, D_1, \ldots, D_N \tag{8}$$

Where A_f denotes fake speech and A_r denotes real speech. $D_1, D_1, \ldots, D_N$ represents different domains. The forged speech MFCC feature F_{f_out} is obtained as described in Section 3.1, and the

real voice MFCC feature F_r is obtained using the MFCC extractor as follows:

$$F_r = MFCC(A_r) \tag{9}$$

Subsequently, F_{f_out} and F_r are sent to the feature extractor to obtain the corresponding features.

$$X_r = E(F_r), X_f = E(F_{f\_out}) \tag{10}$$

where E is the shared feature extractor of F_r and F_{f_out} and X_r and X_f are the corresponding features extracted by E. To achieve domain-adversarial learning for real speech, we send X_r into domain discriminator D, which is used to recognize real speech from different domains. Features extractor E is to confront D by constantly training it to generate features that are invariant to the domain, thus confusing the discriminator and rendering it unable to recognize different domains. In this way, a generalized feature space for real speech is formed. The formula for the optimization objective function is as follows:

$$min_D max_E l_{adv}(E, D) = -\mathbb{E}_{x,y\sim X_r, Y_{D_r}} \sum_{n=1}^{N} \mathbb{1}_{[n=y]} log(D(E(x))) \tag{11}$$

where X_r represents the real-speech feature, Y_{D_r} represents the domain label of the real speech, and N represents the number of domains. This objective function optimizes the parameters of the feature generator by maximizing the adversarial loss and optimizes the parameters of the domain discriminator in the opposite direction. In addition, a **gradient reversal layer (GRL)** [3] is applied to optimize the feature extractor and domain discriminator inversely. In the backpropagation process, the gradient introduced to the GRL is multiplied by a negative coefficient, such that the training objectives of the network before and after the GRL are opposite.

3.2.2 Asymmetric Triplet Loss. Triplet losses typically converge the features of real and forged samples to separate them into feature spaces. However, owing to the joint of the GAN fingerprint, there is a large gap between different GAN-forged voices, and it is difficult to converge them. To form a better feature distribution space, they are separated among domains in the feature space. Therefore, an asymmetric triplet loss is used to separate real speech and forged speech and separate forged speech from different domains. Specifically, real speech is labeled as one class, and different fake speeches as different classes. Then, asymmetric triples are optimized to separate the forged speech into different domains and aggregate the real speech from all source domains. The optimization function for the asymmetric triplet loss is as follows:

$$min_E l_{triplet}(E) = \sum_{x_i^a, x_i^p, x_i^n} \left(\left\| E\left(x_i^a\right) - E\left(x_i^p\right) \right\|_2^2 - \left\| E\left(x_i^a\right) - E\left(x_i^n\right) \right\|_2^2 + \alpha \right) \tag{12}$$

Where x_i^a denotes the anchor, x_i^p denotes positive samples, which gave the same labels with x_i^a, x_i^n denotes the negative samples. α is the margin.

3.3 Dynamic Curriculum Learning

Curriculum learning is a technique used to improve model performance and generalization ability. Its idea is that simple samples should precede difficult samples during training. This is suitable for the forged speech detection task because the forged samples from different domains are easy and difficult. Therefore, we add a confidence score to our classification loss to achieve curriculum learning. Because the easy samples show better detection results (small loss), whereas the difficult samples show poor detection results (large loss), we propose computing the confidence score according to the loss of each sample to represent the degree of difficulty. In this way, the contribution of the sample with a large loss is reduced, and that of the sample with a small loss is increased.

Table 1. Architectures of the LCNN-9 Model

Type	A Filter Size/Stride, Pad	Output Size	# Params
Conv 1	$5 \times 5/1, 2$	$128 \times 128 \times 96$	2.4K
MFM 1	–	$128 \times 128 \times 48$	–
Pool 1	$2 \times 2/2$	$64 \times 64 \times 48$	–
Conv 2a	$1 \times 1/1$	$64 \times 64 \times 96$	4.6K
MFM 2a	–	$64 \times 64 \times 48$	–
Conv 2	$3 \times 3/1, 1$	$64 \times 64 \times 192$	165K
MFM 2	–	$64 \times 64 \times 96$	–
Pool 2	$2 \times 2/2$	32×32	–
Conv 3a	$1 \times 1/1$	$32 \times 32 \times 192$	18K
MFM 3a	–	$32 \times 32 \times 96$	–
Conv 3	$3 \times 3/1, 1$	$32 \times 32 \times 384$	331K
MFM 3	–	$32 \times 32 \times 192$	–
Pool 3	$2 \times 2/2$	16×16	–
Conv 4a	$1 \times 1/1$	$16 \times 16 \times 384$	73K
MFM 4a	–	$16 \times 16 \times 192$	–
Conv 4	$3 \times 3/1, 1$	$16 \times 16 \times 256$	442K
MFM 4	–	$16 \times 16 \times 128$	–
Conv 5a	$1 \times 1/1$	$16 \times 16 \times 256$	32K
MFM 5a	–	$16 \times 16 \times 128$	–
Conv 5	$3 \times 3/1, 1$	$16 \times 16 \times 256$	294K
MFM 5	–	$16 \times 16 \times 128$	–
Pool 4	$2 \times 2/2$	8×8	–
fc 1	–	512	4194K
MFM_fc 1	–	256	–
Total	–	–	5556K

This can help to dynamically adjust the learning process according to the importance of different samples and reduce the contribution of difficult samples in the training process.

Specifically, the confidence parameter ω is added to the classification loss l_{cls} for each sample in a small batch. If we learn an additional learnable parameter ω for each sample, it is not scalable for a detection task with an almost infinite number of samples, so ω is calculated directly in terms of l_{cls}:

$$\omega = e^{-W(\frac{1}{2}max(-\frac{2}{e},\beta))}, \beta = \frac{l_{cls} - \tau}{\lambda} \tag{13}$$

where W is the Lambert function. During backpropagation, ω is calculated in terms of the input loss l_{cls}, which is then treated as a constant. λ and τ are hyperparameters. Then, the obtained ω is added to the existing classification loss, and the final classification loss function is presented as follows:

$$l^*_{cls} = (l_{cls} - \tau)\omega + \lambda(log(\omega)^2 \tag{14}$$

The first term is the loss amplification term, and the second term is the regularization term. $\lambda > 0$, where τ is empirically estimated as the running average of the input loss during training.

3.4 Model Architecture

Our model architecture includes a front-end feature extraction module and a back-end network. It contains an MFCC feature extractor, a reconstructor module, and a fingerprints enhancement

ALGORITHM 1: GAN fingerprint extraction and model optimization of spoofing speech detection based on GAN vocoders

Input: different domain datasets $(A_f, A_r) \in \{D_1, D_2, \ldots, D_N\}$($A_f$ denotes fake audio, A_r denotes real audio), LCNN model E,domain discriminator D

Output: Final LCNN model parameter $\Phi(\cdot)$

reconstruct synthetic speech A_f to $\hat{A_f}$;

extract MFCC feature $F_f \leftarrow A_f$,$\hat{F_f} \leftarrow \hat{A_f}$,$F_r \leftarrow A_r$;

extract GAN fingerprint from residual of F_f and $\hat{F_f}$: $F_{GAN} \leftarrow (F_f - \hat{F_f})$;

Input F_{GAN} into CBAM module to get enhanced GAN fingerprint F_{GAN_out} based on Equations (3)–(5);

Input F_{GAN_out} into $\hat{F_f}$ to get distinguish feature F_{f_out} based on Equation (6);

while *not end of iteration* **do**

  Extract feature from LCNN model $X_r = E(F_r)$, $X_f = E(F_{f\_out})$;

  Input X_r to the discriminator D and compute the adversarial loss L_{adv} based on Equation (10);

  Utilize X_r and X_f to compute the asymmetric triplet loss $L_{triplet}$ based on Equation (11);

  Input X_r and X_f to the classifier and compute the classification loss L_{cls};

  Apply a confidence parameter ω to L_{cls} to achieve dynamic curriculum learning: $L^*_{cls} \leftarrow L_{cls}$;

  Compute $L_{total} = L^*_{cls} + \lambda_1 * L_{adv} + \lambda_2 * L_{triplet}$;

  Make gradient back propagation and update the model parameters $\Phi(\cdot)$;

end

module in the front-end. An MFCC feature extractor is used to extract the acoustic features as input for the back-end network. The reconstructor is an autoencoder based on Resnet18, which is used to extract GAN fingerprints. The fingerprint enhancement module consists of a sigmoid, a 1×1 conv, and a CBAM module, as detailed in Section 3.3.2. In the back-end network, we use LCNN-9 as the backbone, and the core operation of the LCNN is the **max-feature map (MFM)**, which is a new feature map fusion method. The LCNN-9 contains nine layers of MFM operations, and its specific architecture is listed in Table 1. The classifier contained an LCNN and an FC layer with two nodes, and the domain discriminator in adversarial learning contained two FC layers with 512 nodes and three nodes.

3.5 Loss Function

The domain-adversarial loss is used to gather the shared feature space for real samples, whereas the asymmetric triplet loss is used to widen the distance between real and fake samples and increase the distinction between fake samples in different domains. Dynamic curriculum learning is used by adding a confidence score to the classified loss to improve the generalization ability and detection performance of the model. All the above works are integrated to obtain the total loss of work as follows. We outline the whole active learning process in Algorithm 1.

$$\mathcal{L}_{total} = l^*_{cls} + \lambda_1 * l_{adv} + \lambda_2 * l_{triplet} \tag{15}$$

where λ_1 and λ_2 are the constraint parameters.

4 EXPERIMENT

4.1 Dataset

In this study, four sets of real speech and the corresponding sets of four GAN vocoder-generated speeches are used to evaluate the effectiveness of the proposed method.

Real speech: Four sets of real speech are used: LJSpeech, CSMSC, JUST, and KSS. LJSpeech is a public-domain speech dataset consisting of 13,100 short audio clips from passages read by speakers in seven nonfiction books. Transcripts are provided for each clip. The clips range in length from 1 to 10 s, with a total length of approximately 24 h and a sampling rate of 22.05 kHz. JUST consists of a Japanese text (transcribed) and reading-style audio. Audio data is sampled at 48 kHz. The corpus contains 10 h of speech and various data. From the basic 5000 dataset, 5,000 corpora are used. The CSMSC collect professional standard Putonghua female vocals between 20 and 30 years old using professional recording equipment and software, consisting of 10,000 recordings, with a total length of approximately 12 h and a sampling rate of 48 kHz. The KSS dataset is designed for Korean TTS tasks. It consists of audio files recorded by professional female voice actors and aligned text extracted from books. It contains 12,853 corpora with a total length of more than 12 h and a sampling rate of 44.10 kHz.

Forged speech: There are four different GAN vocoders for forged-speech synthesis: Parallel WaveGAN [37], HifiGAN [7], Style-MelGAN [20], and MelGAN [9]. The pre-training model of each GAN is used to generate the corresponding GAN-forged speech for each group of real speeches. Four datasets are formed: LJSpeech and its MelGAN-synthesized speech (labeled M), CSMSC and its Style-MelGAN-synthesized speech (labeled S), JUST and its HifiGAN-synthesized speech (labeled H), KSS and its Parallel WaveGAN-synthesized speech (labeled P). One of them is randomly selected as the target domain for the test, and the other three are selected as the source domain for training. Thus, we have four test tasks: M& S & H to P; M & P & S to H; M & P & H to S; P & S & H to M.

4.2 Experimental Setting

The sampling rate of all data is unified at 24khz, and the duration is unified at 2 seconds. We use MFCC as the speech front-end feature extractor to extract 60-dimensional MFCC features. A Hamming analysis window is applied, with a size of 25ms and an offset of 10ms. Nfft is set to 512.

We conduct four groups of experiments in total, and the back-end feature extraction model uses LCNN architecture. The model parameters are randomly initialized. We use the ADAM optimizer to update the weight parameters of the model. The momentum is 0.9, the weight decay is 0.0005, and the learning rate is 0.0001. In dynamic course learning, we set τ as a fixed threshold $\tau = log(C)$, where C is the number of categories. We set $\lambda = 1$, $\lambda_1 = 2$, $\lambda_2 = 0.5$ (as details for selecting parameters can be seen in Section 4.4.4). There are 5565k parameters in total in our LCNN model and domain discriminator. The network is trained on a single NVIDIA GTX 2080 Ti GPU. The batch size is 24 and the number of epochs is 50 without early stopping.

4.3 Evaluation Metric

Two evaluation metrics are used in this study: the **equal error rate** (**EER**, %) and the **area under the curve** (**AUC**, %). The EER is an important metric used to evaluate the performance of a spoofing detection system and measure the security and accuracy of the system. It is used to determine the error acceptance rate and the threshold for the error rejection rate in advance. When the rates are equal, the common value is called the EER. This value indicates that the proportion of false acceptances is equal to that of false rejections. The lower the error rate, the higher the accuracy of the biometric identification system. The error rejection rate E_{FR} and error acceptance rate E_{FA} are defined as:

$$E_{FR} = \frac{N_{fr}}{N_{target}} \tag{16}$$

$$E_{FA} = \frac{N_{fa}}{N_{non\text{-}target}} \tag{17}$$

N_{fr} and N_{fa} refer to the number of false rejections and false acceptances in the test, and N_{target} and $N_{non\text{-}target}$ refer to the total number of true tests and the number of false tests in the test, respectively. When the threshold in the system is fixed, E_{FR} and E_{FA} are fixed. As the threshold decreases, more tests will be accepted, and E_{FA} increases and E_{FR} decreases. On the contrary, when the threshold increases, the test will not pass easily. E_{FR} increases, and E_{FA} decreases. The error rate when E_{ER} is $FR = FA$: $E_{ER}R = E_{FR} = E_{FA}$.

The AUC is one of the most important evaluation metrics for checking the performance of the classification model. The **receiver operating characteristic (ROC)** curve is drawn with the TPR relative to the FPR, where the TPR is on the y-axis and the FPR is on the x-axis:

$$TPR = \frac{TP}{TP + FN} \tag{18}$$

$$FPR = \frac{TN}{TN + FP} \tag{19}$$

TPR denotes the true positive rate, and FPR denotes the false positive rate. AUC = area under the ROC curve. The closer the value of the AUC is to 1, the better the classifier performance. In contrast, the closer the AUC value is to 0, the worse the classifier performance.

4.4 Experimental Analysis

4.4.1 Comparison with Existing Detection Methods. The generalization performance of the model is tested using three of the four datasets as the source domain and one as the target domain. As can be seen from Table 2, our method exhibits a significant improvement over the baseline method, which proves the effectiveness of our method. In addition, our method outperforms existing generalization methods [6, 11, 15, 19, 25, 38, 40]. [11, 40] improved detection generalization from the perspective of training strategies, [40] focused on the distribution of real speech to obtain a generalized feature space. [11] proposed an angular-margin-based softmax activation for training a generalized classifier. However, they did not consider the intrinsic connection of real speech in different domains and did not find domain-invariant features between them. [6, 15] achieved generalization ability through back-end network optimization, ignoring the discriminative cues existing in the front-end feature. Recently, [25, 38] attempted to develop generalization ability by digging vocoder artifacts during training. [19] proposed enhancing the robustness of networks using multitask learning. The limitation of these methods is that they need the help of auxiliary tasks to mine artifacts. Instead, our method outperforms [19, 25, 38] because we extract GAN fingerprints directly from vocoders, without the help of other tasks. The GAN fingerprints can amplify the distribution difference between real speech and forged speech. Besides, we achieve DG to force the model to pay attention to the distribution relationships among different domains, thus forming a more generalized feature space.

4.4.2 Performance of Each Component Module. In this section, detailed ablation studies are presented for each component of the proposed framework to evaluate the performance of different combinations using each component. The framework in this study is divided into three main components: **fingerprint extraction (fprint)**, DG, and dynamic **curriculum learning (CL)**. The influence of different combinations of various components on the detection performance is studied through four experimental groups, i.e., M& S & H to P; M & P & S to H; M & P & H to S; P & S & H to M., and the results are shown in Table 3.

Each component is added to the baseline model. As shown in Table 3, when the GAN fingerprints and DG strategy are used separately based on the baseline, the detection performance of the four groups of experiments is improved. In particular, when we use DG strategy,the P & S & H to M testing task shows high performance improvement, with an EER increase of more than 10%.

Table 2. Comparison of Our Method and the State-of-the-Art Methods for Generalization of Forgery Speech Detection

Method	M& S & H to P		M& S & P to H		M& P & H to S		P& S & H to M	
	EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)
Baseline(MFCC+LCNN)	7.32	97.88	7.78	96.55	2.87	99.62	32.91	73.16
AM-Softmax [11]	6.52	97.99	0.96	99.63	0.42	99.98	31.81	73.80
OC-Softmax [40]	5.87	98.02	2.18	99.61	0.68	99.96	28.35	75.77
Res2Net [15]	3.38	99.46	0.79	99.88	0.60	99.92	25.89	82.81
Vocoder fingerprints [38]	3.19	99.39	0.52	99.97	0.73	99.96	23.80	85.78
Vocoder Artifacts [25]	2.98	99.45	0.43	99.97	0.88	99.93	20.95	90.43
Res-TSSDNet [6]	1.75	99.80	0.54	99.78	0.40	99.98	15.72	92.40
Multi-Task [19]	1.80	99.82	0.54	99.56	0.42	99.99	10.17	96.38
ours	**0.25**	**99.99**	**0.36**	**99.89**	**0.29**	**99.99**	**3.09**	**99.54**

Table 3. Evaluations of Different Components of the Proposed Model in the Task on Four Test Sets

fprint	DG	CL	M& S & H to P		M& S & P to H		M& P & H to S		P& S & H to M	
			EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)
			7.32	97.88	7.78	96.55	2.87	99.62	32.91	73.16
✓			4.98	98.91	5.20	98.14	1.05	99.91	26.25	81.55
	✓		4.24	99.15	5.04	98.12	1.06	99.95	21.71	86.15
✓	✓		0.58	99.98	2.84	99.64	0.56	99.97	13.13	9.29
✓	✓	✓	**0.25**	**99.99**	**0.36**	**99.89**	**0.29**	**99.99**	**3.09**	**99.54**

The synthesized speech of the MelGAN vocoder contains noise, so the above result indicates that DG has a strong generalization ability without interference from noise. When GAN fingerprints and DG are used together, the performance has been further improved, which proves that the combination of GAN fingerprints and DG can improve the detection generalization performance of the model. Finally, the effect is further improved when all the modules are applied. This shows that all three modules are effective in improving the detection generalization of the model.

We further visualize the feature distribution using t-SNE to validate the performance of the fprint and DG modules under the M & S & H to P testing tasks. As can be seen in Figure 5, the classification boundary in baseline (a) is relatively vague compared to (b) and (c). Besides, the distribution distance of real and forged speech in (c) is further enlarged compared to (b), which indicates that the GAN fingerprints we extract can help to better discriminate real and synthetic speech than [38]. We further draw the distribution while training with the DG, which is achieved by domain-adversarial learning and asymmetric triplet loss, as shown in Figure 5(d). It can be seen that (d) shows a better generalized feature space than (c). Although (c) can enlarge the boundary between real and forged speech, there is a gap between the fake speech of different domains because the GAN fingerprint is added. If we use DG for fake speech, it will be difficult to form a compact feature space for them. Thus, we utilize domain-adversarial learning for real speech and conducted asymmetric triplet loss between real and fake speech to form a better generalized feature space, where real speech is compact in the feature space and fake speech is separated among the domains.

4.4.3 Influence of Domain-Adversarial Loss and Asymmetric Triplet Loss in DG. In the DG module, the domain-adversarial loss of real speech is used to determine the shared feature space of

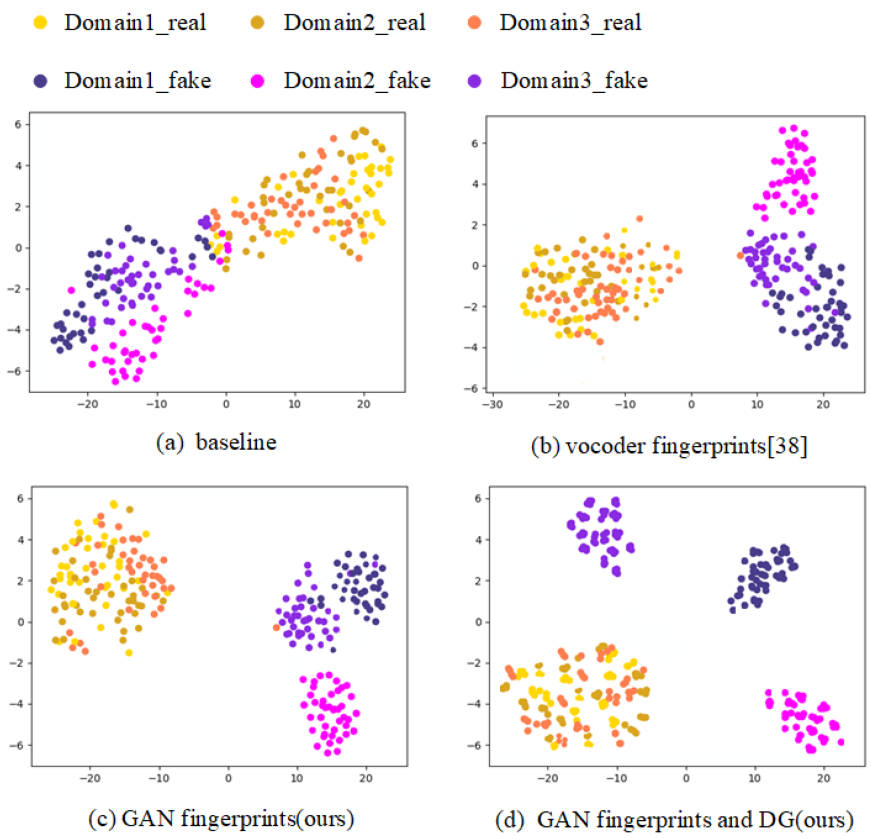


Fig. 5. The t-SNE features visualizations. The distribution is best viewed in color. Features in (a) are extracted by the baseline method, features in (b) are extracted by the method [38] which extract vocoder artifacts with the help of vocoder recognition task, features in (c) are extracted by our proposed method with GAN fingerprints, features in (d) are extracted by our proposed method with GAN fingerprints and DG. These experiments are conducted under M & S & H to P testing tasks.

real speech. Asymmetric triplet loss is used to reduce the intra-class distance of real speech and widen the inter-class distance between real speech and forged speech in each domain. To prove the validity of the two losses, the two modules are ablated among four groups of experiments. To observe the influence of the two losses on the model detection performance, three conditions are considered: with adv (to use domain-adversarial loss), with triplet (to use asymmetric triplet loss), and with both (to use domain-adversarial loss and asymmetric triplet loss). The results are summarized in Table 4.

As shown in Table 4, when real voice adversarial learning and asymmetric triplet loss are used simultaneously, the model exhibited the best detection effect. If any of the components are removed, the model's detection performance in the unknown domain is reduced to varying degrees, which proves that domain- adversarial learning and triplet loss in DG help improve the model effect. In particular, the effect is greatest when they were combined.

4.4.4 Parameter Selection. **Regularization parameters λ in CL regularization**: To examine the influence of different regularization parameters λ on the classification performance in curriculum learning, four parameter values are set: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1$, and $\lambda = 10$. The test results

Table 4. Effects of the Domain-Adversarial Loss and Asymmetric Triplet Loss in DG

Method	M& S & H to P		M& S & P to H		M& P & H to S		P& S & H to M	
	EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)
with adv	6.13	98.45	6.27	97.05	2.43	99.75	24.83	83.22
with triplet	4.84	98.99	6.26	97.44	1.19	99.92	22.04	84.93
with both	**4.24**	**99.15**	**5.04**	**98.12**	**1.06**	**99.95**	**21.71**	**86.15**

Table 5. Influence of Different λ Values on Detection Performance in CL

Params	M& S & H to P		M& S & P to H		M& P & H to S		P& S & H to M	
	EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)
$\lambda = 0.01$	3.49	97.04	0.70	99.39	3.27	99.55	35.16	67.41
$\lambda = 0.1$	2.49	98.86	0.62	99.62	0.27	99.73	33.75	75.58
$\lambda = 1$	**1.98**	**99.81**	**0.40**	**99.79**	**0.37**	**99.99**	**17.54**	**90.61**
$\lambda = 10$	3.99	97.52	0.45	99.57	39.02	75.04	0.44	99.56

Table 6. Influence of Different λ_1 and λ_2 on Detection Performance

Params	M& S & H to P		M& S & P to H		M& P & H to S		P& S & H to M	
	EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)
$\lambda_1 = 1, \lambda_2 = 1$	3.00	9.62	0.60	99.98	0.67	99.97	17.34	90.42
$\lambda_1 = 0.5, \lambda_2 = 1$	2.94	99.62	2.55	99.70	0.63	99.98	17.46	90.53
$\lambda_1 = 0.5, \lambda_2 = 2$	2.71	99.66	2.41	99.73	0.56	99.98	13.42	93.53
$\lambda_1 = 1, \lambda_2 = 0.5$	1.89	99.84	1.65	99.87	0.44	99.98	11.95	94.95
$\lambda_1 = 2, \lambda_2 = 0.5$	**0.30**	**99.99**	**0.40**	**99.79**	**0.37**	**99.99**	**9.60**	**96.68**

of the four groups of experiments, i.e., M& S & H to P; M & P & S to H; M & P & H to S; P & S & H to M, are shown in Table 5.

Table 5 compares the influence of different λ on the model detection performance. When $\lambda = 1$, all the four groups of experiments show the best performance, which means the confidence score added to the classification loss can best adjust the contribution of the training sample to the model when $\lambda = 1$. This result also proves that curriculum learning at this time has the best effect on the optimization of the classification task.

Super parameter λ_1 and λ_2 in $\mathcal{L}_{total}$: λ_1 and λ_2 are the balanced parameters for the total loss $\mathcal{L}_{total}$. We conduct a set of experiments to select suitable λ_1 and λ_2 values for the proposed method. We set five groups of values for λ_1 and λ_2 based on prior experimental experience. From Table 6 we can see that all the experiments show the best performance when $\lambda_1 = 2$, $\lambda_2 = 0.5$, especially in the "P & S & H to M" group. Thus, we select $\lambda_1 = 2$, $\lambda_2 = 0.5$ as our best parameters.

τ in Dynamic Curriculum Learning: τ is a threshold that ideally separates easy samples from hard samples based on their respective loss. According to [1], τ can be estimated as the running average of the input loss during training or set as a constant based on prior knowledge of the task. To determine which is better for our work, we select τ as a fixed threshold $\tau = log(C)$ according to [1], where C is the number of classes. We respectively use $\tau = log(C)$ and τ as a running average in our experiment, then compared the detection performance. As shown in Figure 6, it is observed that performance is similar whether using $\tau = log(C)$ or using running averaging. Thus, we randomly select $\tau = log(C)$ for the proposed method.

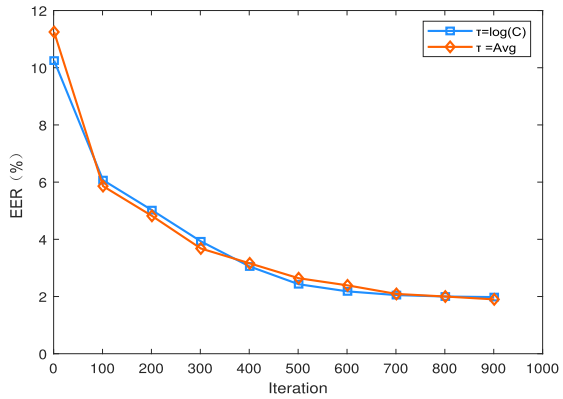


Fig. 6. The performance of the model, with $\tau = log(C)$ or $\tau = trainingAvg$ under M & S & H to P testing tasks.

Table 7. Comparison of Results in the Limited Source Domains

Method	M& H to P		M& P to H		P& H to M	
	EER(%)	AUC(%)	EER(%)	AUC(%)	EER(%)	AUC(%)
Baseline(MFCC+LCNN)	12.86	94.50	12.79	93.09	41.68	62.08
AM-Softmax [11]	9.57	95.48	4.93	97.36	34.87	62.28
OC-Softmax [40]	9.11	95.90	13.69	93.47	30.25	64.59
Res2Net [15]	6.48	98.49	5.14	98.88	29.23	75.77
Vocoder fingerprints [38]	6.31	98.08	4.60	97.70	25.72	77.16
Vocoder Artifacts [25]	5.27	97.80	4.43	97.97	23.92	83.00
Res-TSSDNet [6]	4.19	98.13	4.82	98.08	21.47	86.47
Multi-Task [19]	4.36	97.09	4.56	96.09	14.19	90.26
ours	**3.38**	**99.52**	**3.38**	**99.52**	**9.08**	**97.05**

4.4.5 Impact of Reduced Source Domains. In the previous experiment, three source domains for DG are used. The effectiveness of the proposed approach is evaluated when the available source domains are limited (i.e., only two source domains). Specifically, MelGAN, HifiGAN, and Parallel WaveGAN are selected for the three groups of experiments.

As shown in Table 7, with only two source domains, the proposed method also achieves the best performance among the other methods, with the largest improvement in the generalization performance of forged speech detection compared to the other methods. In other words, in the case of limited source domains, this method can gather the feature space of real speech and widen the feature distance between real and forged speech. This has high application value in real-world scenarios with more invisible GAN vocoders.

4.4.6 Fingerprint Feature Visualization. To prove the uniqueness of the fingerprints of the different GAN vocoders, the same real speech is used to synthesize four fake voices from four GAN vocoders. According to the process described in Section 3.1, the four voices are sent to the Griffin-Lim vocoder for reconstruction, and the fingerprints of the four GANs are then obtained. Subsequently, the MFCC features of the four GAN fingerprints are extracted, and the MFCC feature map is visualized. Figure 5 illustrates the following. As can be seen from the dotted line in Figure 7(a), the forged speech synthesized by different GAN vocoders is quite different in the

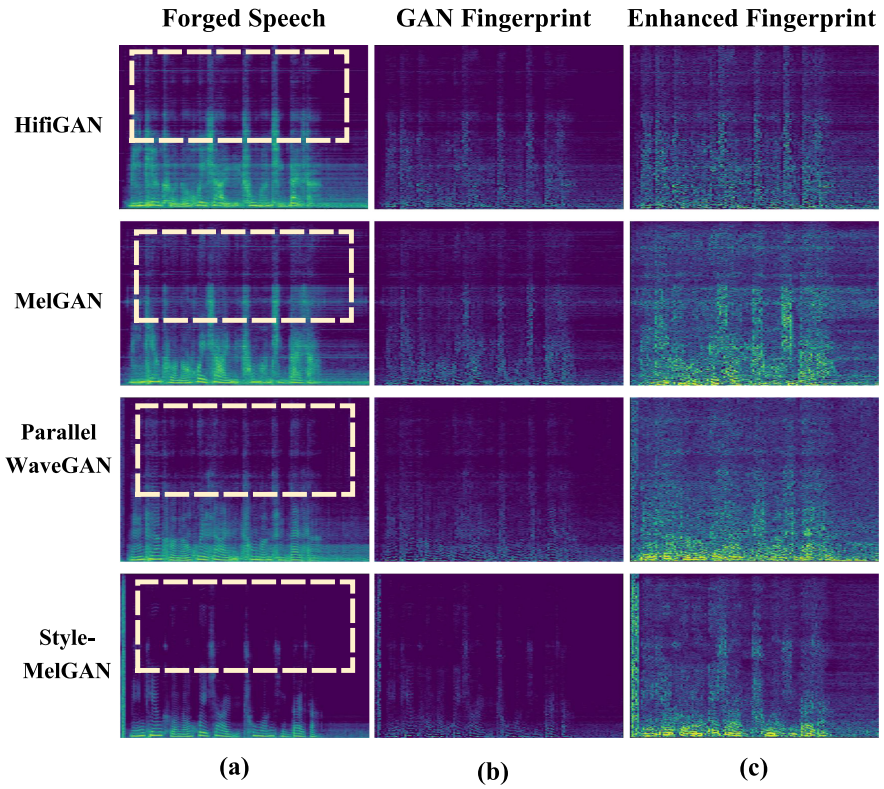


Fig. 7. Four GAN vocoders are used to synthesize forged speech from the same speech. The abscissa represents time, the ordinate is frequency. Each row represents a GAN vocoder. Column (a) represents the spectrogram of different GAN forged speech, column (b) represents the GAN fingerprint extracted from the forged speech, column (c) represents the enhanced GAN fingerprint spectrogram. The brightness of the colors in the figure represents the energy of the speech data.

high-frequency part; this discrepancy proves the existence of GAN fingerprints. Figure 7(b) is the result of GAN fingerprints visualization. Because GAN fingerprints are relatively weak, we amplified and enhanced them, and the enhanced GAN fingerprints have a higher energy, as shown in Figure 7(c).

5 CONCLUSION

To improve the generalization ability of forged speech detection on unknown datasets, an end-to-end detection framework based on GAN fingerprint extraction and DG is proposed, and dynamic curriculum learning is used to schedule sample learning strategies. First, the existence of GAN fingerprints is demonstrated by reconstructing fake speech using an autoencoder and obtaining GAN fingerprints from the residual of forged speech and reconstructed-forged speech. The GAN fingerprint represents the unique information of different GAN vocoders and can increase the inter-class distance between real speech and forged speech. We then achieve DG learning by domain-adversarial learning for real speech and asymmetric triplet optimization. Through DG learning, a compact feature distribution space can be learned for real voices, and forged speech in different domains can be dispersed in space simultaneously. Finally, dynamic curriculum learning

is introduced to add a confidence score based on the classification loss. This confidence score is used to adjust the importance of the different difficulty samples. Multiple experiments have shown that the proposed method is effective in improving the generalization performance of detection. In the future, we will apply our methods to explore better detection performance for other forgery scenes, such as environmental sound synthesis and not only human speech. Therefore, the proposed method can be applied more generally.

REFERENCES

[1] Thibault Castells, Philippe Weinzaepfel, and Jerome Revaud. 2020. SuperLoss: A generic loss for robust curriculum learning. *Advances in Neural Information Processing Systems* 33 (2020), 4308–4319.

[2] Zhuxin Chen, Zhifeng Xie, Weibin Zhang, and Xiangmin Xu. 2017. ResNet and model fusion for automatic spoofing detection. In *Interspeech*. 102–106.

[3] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*. PMLR, 1180–1189.

[4] Alex Graves, Marc G. Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *International Conference on Machine Learning*. PMLR, 1311–1320.

[5] Wei Han, Cheong-Fat Chan, Chiu-Sing Choy, and Kong-Pang Pun. 2006. An efficient MFCC extraction method in speech recognition. In *2006 IEEE International Symposium on Circuits and Systems (ISCAS'06)*. IEEE, 4–pp.

[6] Guang Hua, Andrew Beng Jin Teoh, and Haijian Zhang. 2021. Towards end-to-end synthetic speech detection. *IEEE Signal Processing Letters* 28 (2021), 1265–1269.

[7] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems* 33 (2020), 17022–17033.

[8] A. Kishore Kumar, Dipjyoti Paul, Monisankha Pal, Md. Sahidullah, and Goutam Saha. 2021. Speech frame selection for spoofing detection with an application to partially spoofed audio-data. *International Journal of Speech Technology* 24 (2021), 193–203.

[9] Kundan Kumar, Rithesh Kumar, Thibault De Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre De Brebisson, Yoshua Bengio, and Aaron C. Courville. 2019. MelGAN: Generative adversarial networks for conditional waveform synthesis. *Advances in Neural Information Processing Systems* 32 (2019).

[10] Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Vadim Shchemelinin. 2017. Audio replay attack detection with deep learning frameworks. In *Interspeech*. 82–86.

[11] Galina Lavrentyeva, Sergey Novoselov, Andzhukaev Tseren, Marina Volkova, Artem Gorlanov, and Alexandrg Kozlov. 2019. STC antispoofing systems for the ASVspoof2019 challenge. *arXiv preprint arXiv:1904.05576* (2019). https://doi.org/10.48550/arXiv.1904.05576

[12] Zhenchun Lei, Yingen Yang, Changhong Liu, and Jihua Ye. 2020. Siamese convolutional neural network using Gaussian probability feature for spoofing speech detection. In *Interspeech*. 1116–1120.

[13] Hao Li, Yongguo Kang, and Zhenyu Wang. 2018. EMPHASIS: An emotional phoneme-based acoustic model for speech synthesis system. *arXiv preprint arXiv:1806.09276* (2018). https://doi.org/10.48550/arXiv.1806.09276

[14] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. 2018. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5400–5409.

[15] Xu Li, Xixin Wu, Hui Lu, Xunying Liu, and Helen Meng. 2021. Channel-wise gated Res2Net: Towards robust detection of synthetic speech attacks. *arXiv preprint arXiv:2107.08803* (2021). https://doi.org/10.48550/arXiv:2107.08803

[16] Jiachen Ma, Yong Liu, Meng Liu, and Meng Han. 2022. Curriculum contrastive learning for fake news detection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4309–4313.

[17] Kaijie Ma, Yifan Feng, Beijing Chen, and Guoying Zhao. 2023. End-to-end dual-branch network towards synthetic speech detection. *IEEE Signal Processing Letters* 30 (2023), 359–363.

[18] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. 2019. Do GANs leave artificial fingerprints?. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR'19)*. IEEE, 506–511.

[19] Yichuan Mo and Shilin Wang. 2022. Multi-task learning improves synthetic speech detection. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'22)*. IEEE, 6392–6396.

[20] Ahmed Mustafa, Nicola Pia, and Guillaume Fuchs. 2021. StyleMelGAN: An efficient high-fidelity adversarial vocoder with temporal adaptive normalization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'21)*. IEEE, 6034–6038.

[21] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. *ISCA Speech Synthesis Workshop* (2016), 125–125. https://doi.org/10.48550/arXiv.1609.03499

[22] Tanvina B. Patel and Hemant A. Patil. 2017. Significance of source–filter interaction for classification of natural vs. spoofed speech. *IEEE Journal of Selected Topics in Signal Processing* 11, 4 (2017), 644–659. https://doi.org/10.1109/JSTSP.2017.2682788

[23] Dipjyoti Paul, Monisankha Pal, and Goutam Saha. 2017. Spectral features for synthetic speech detection. *IEEE Journal of Selected Topics in Signal Processing* 11, 4 (2017), 605–617. https://doi.org/10.1109/JSTSP.2017.2684705

[24] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C. Yuen. 2019. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10023–10031.

[25] Chengzhe Sun, Shan Jia, Shuwei Hou, and Siwei Lyu. 2023. AI-synthesized voice detection using neural vocoder artifacts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 904–912.

[26] Hemlata Tak, Jee-weon Jung, Jose Patino, Madhu Kamble, Massimiliano Todisco, and Nicholas Evans. 2021. End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection. *arXiv preprint arXiv:2107.12710* (2021). https://doi.org/10.48550/arXiv:2107.12710

[27] Hemlata Tak, Jose Patino, Andreas Nautsch, Nicholas Evans, and Massimiliano Todisco. 2020. Spoofing attack detection using the non-linear fusion of sub-band classifiers. *arXiv preprint arXiv:2005.10393* (2020). https://doi.org/10.48550/arXiv.2005.10393

[28] Massimiliano Todisco, Héctor Delgado, and Nicholas W. D. Evans. 2016. A new feature for automatic speaker verification anti-spoofing: Constant Q Cepstral coefficients. In *Odyssey*, Vol. 2016. 283–290.

[29] Massimiliano Todisco, Héctor Delgado, Kong Aik Lee, Md. Sahidullah, Nicholas Evans, Tomi Kinnunen, and Junichi Yamagishi. 2018. Integrated presentation attack detection and automatic speaker verification: Common features and Gaussian back-end fusion. In *Interspeech 2018-19th Annual Conference of the International Speech Communication Association*. ISCA.

[30] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7167–7176.

[31] Huaming Wang, Jianwei Fei, Yunshu Dai, Lingyun Leng, and Zhihua Xia. 2022. General GAN-generated image detection by data augmentation in fingerprint domain. *arXiv preprint arXiv:2212.13466* (2022). https://doi.org/10.48550/arXiv:2212.13466

[32] Wenfu Wang, Shuang Xu, and Bo Xu. 2016. First step towards end-to-end parametric TTS synthesis: Generating spectral parameters with neural attention. In *Interspeech*. 2243–2247.

[33] Xin Wang and Junich Yamagishi. 2021. A comparative study on recent neural spoofing countermeasures for synthetic speech detection. *arXiv preprint arXiv:2103.11326* (2021). https://doi.org/10.48550/arXiv:2103.11326

[34] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. 2017. Tacotron: Towards end-to-end speech synthesis. *Proc. Interspeech* (2017), 4006–4010. https://doi.org/10.48550/arXiv.1703.10135

[35] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. 3–19.

[36] Pengxiang Xu, Xue Mei, Yi Wei, and Tiancheng Qian. 2021. Robust facial manipulation detection via domain generalization. In *2021 7th International Conference on Computing and Artificial Intelligence*. 196–201.

[37] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2020. Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'20)*. IEEE, 6199–6203.

[38] Xinrui Yan, Jiangyan Yi, Jianhua Tao, Chenglong Wang, Haoxin Ma, Tao Wang, Shiming Wang, and Ruibo Fu. 2022. An initial investigation for detecting vocoder fingerprints of fake audio. In *Proceedings of the 1st International Workshop on Deepfake Detection for Audio Multimedia*. 61–68.

[39] Ning Yu, Larry S. Davis, and Mario Fritz. 2019. Attributing fake images to GANs: Learning and analyzing GAN fingerprints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7556–7566.

[40] You Zhang, Fei Jiang, and Zhiyao Duan. 2021. One-class learning towards synthetic voice spoofing detection. *IEEE Signal Processing Letters* 28 (2021), 937–941. https://doi.org/10.1109/LSP.2021.3076358

[41] Yuxiang Zhang, Wenchao Wang, and Pengyuan Zhang. 2021. The effect of silence and dual-band fusion in anti-spoofing system. In *Proc. Interspeech*.

Received 24 May 2023; revised 25 August 2023; accepted 21 October 2023