





# LMD: A Learnable Mask Network to Detect Adversarial Examples for Speaker Verification

Xing Chen , Jie Wang, *Graduate Student Member, IEEE*, Xiao-Lei Zhang , *Senior Member, IEEE*, Wei-Qiang Zhang , *Senior Member, IEEE*, and Kunde Yang 

**Abstract**—Although the security of automatic speaker verification (ASV) is seriously threatened by recently emerged adversarial attacks, there have been some countermeasures to alleviate the threat. However, many defense approaches not only require the prior knowledge of the attackers but also possess weak interpretability. To address this issue, in this paper, we propose an *attacker-independent* and *interpretable* method, named *learnable mask detector* (LMD), to separate adversarial examples from the genuine ones. It utilizes score variation as an indicator to detect adversarial examples, where the score variation is the absolute discrepancy between the ASV scores of an original audio recording and its transformed audio synthesized from its masked complex spectrogram. A core component of the score variation detector is to generate the masked spectrogram by a neural network. The neural network needs only genuine examples for training, which makes it an attacker-independent approach. Its interpretability lies that the neural network is trained to minimize the score variation of the targeted ASV, and maximize the number of the masked spectrogram bins of the genuine training examples. Its foundation is based on the observation that, masking out the vast majority of the spectrogram bins with little speaker information will inevitably introduce a large score variation to the adversarial example, and a small score variation to the genuine example. Experimental results with 12 attackers and two representative ASV systems show that our proposed method outperforms five state-of-the-art baselines. The extensive experimental results can also be a benchmark for the detection-based ASV defenses.

**Index Terms**—Adversarial examples, detection, passive defense, automatic speaker verification.

Manuscript received 1 November 2022; revised 14 April 2023 and 17 May 2023; accepted 11 June 2023. Date of publication 22 June 2023; date of current version 30 June 2023. This work was supported in part by the National Science Foundation of China (NSFC) under Grants 62176211 and 62276153 and in part by the Project of the Science, Technology, and Innovation Commission of Shenzhen Municipality, China under Grants JCYJ20210324143006016 and JSGG20210802152546026. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Omid Sadjadi. (*Corresponding author: Xiao-Lei Zhang.*)

Xing Chen, Jie Wang, and Xiao-Lei Zhang are with the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China, and also with the Research & Development Institute of Northwestern Polytechnical University, Shenzhen 710072, China (e-mail: xing.chen@mail.nwpu.edu.cn; wangjie2017@mail.nwpu.edu.cn; xiaolei.zhang@nwpu.edu.cn).

Wei-Qiang Zhang is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: wqzhang@tsinghua.edu.cn).

Kunde Yang is with the Ocean Institute of Northwestern Polytechnical University, Xi'an 710072, China (e-mail: ykdzym@nwpu.edu.cn).

Digital Object Identifier 10.1109/TASLP.2023.3288417

## I. INTRODUCTION

**A**UTOMATIC speaker verification (ASV) is a task of verifying the identity of a speaker by his (or her) pre-recorded utterance clips [1]. Deep-learning-based speaker verification techniques can be categorized into two representative frameworks: *stage-wise* [2], [3], [4] and *end-to-end* [5], [6], [7]. A fundamental difference between the two frameworks is their loss functions, which are called *classification-based* loss and *verification-based* loss, respectively [1]. Both of the two frameworks have achieved excellent performance and have penetrated our daily lives with real-world applications such as authentication, bank transaction and forensics. However, adversarial attacks [8] were found to be able to defeat an ASV system even at a high signal-to-noise ratio (SNR) [9], which brought great challenges to the applications of the ASV systems.

Adversarial attack is a technique that aims to induce an ASV system to make wrong decisions by adding human-imperceptible perturbations into the clean speech during the inference phase of ASV. The perturbed speech, a.k.a *adversarial examples*, has been extensively studied in the ASV research [10], [11]. It can be generally classified into white-box attacks and black-box attacks. In the case of white-box attacks, i.e. the scenarios where the victim ASV model exposes all knowledge, including parameters, structure, and training data, to the attacker. Villalba et al. [9] found that the state-of-the-art (SOTA) x-vector ASV models are extremely vulnerable even at a high SNR level of 30 dB. Xie et al. [12] proposed to train a generator to efficiently craft adversarial examples. Since the white-box attacks have many obstacles in the reality, the black-box counterparts, which are knowledge-independent, have been paid more attention. Chen et al. [13] deployed a gray-box attack using only the output similarity scores of ASV. Further, ASV models were found to be vulnerable to transfer-based black-box attacks across training datasets [14] and model structures [15]. In addition, the works in [12], [16] explored robust adversarial examples in terms of the universality and imperceptibility, respectively. There are also some works focusing on applying adversarial attacks to realistic scenarios, such as the over-the-air [17], [18] or streaming input [17], [19] situations, and defeating the tandem system of ASV and its auxiliary subsystems [18], [20].

Since adversarial attacks have posed the serious threat, it becomes foremost important to develop an effective countermeasure to protect the ASV systems. The current countermeasures fall into two categories: *proactive defense* and *passive defense*.

Proactive defense mainly utilizes adversarial data augmentation techniques to retrain the ASV model, which is inconvenient to deploy [10]. For example, the works in [21], [22], [23] proposed to use adversarial examples generated by fast gradient sign method (FGSM), projected gradient descent (PGD) and feature scattering, respectively, to perform adversarial training defenses [24]. Passive defense does not modify the ASV model, instead, it defends against adversarial attacks by a mitigation or detection component. For example, the works in [25], [26], [27] proposed to remove the adversarial noise with the adversarial separation network, Parallel-Wave-GAN (PWG) module, and cascaded self-supervised learning based reformer (SSLR), respectively. Wu et al. [28] also employed a voting strategy with random sampling to mitigate the adversarial attacks.

This article focuses on the detection-based passive defense approaches. There have been many works in this direction. Li et al. [29] and Joshi et al. [30] discriminated adversarial and genuine examples by training a VGG-like binary classification network and an embedding feature extractor, respectively. However, their performance drops dramatically in unseen attacks, since the training relies on the prior knowledge of adversarial examples. Wu et al. [27] picked out adversarial examples by the statistics of the similarity scores between enrollment utterance and synthesized utterances from multiple cascaded SSLRs. However, their experiments were conducted on the MFCC-level, which means it works in the time-frequency domain and relies on specific acoustic feature extractors. Peng et al. [31] proposed to train a binary classifier by the consistency of the scores of twin ASV models, i.e. a premier and a mirror one. Because training the classifier needs genuine examples only, the method gets rid of the dependence on specific attackers. However, it needs to find a SOTA fragile ASV and a rare robust ASV. Wu et al. [32] proposed to detect adversarial examples by score variation, which was obtained by a vocoder composed of the Griffin-Lim (GL) algorithm or PWG model. However, it lacks strong interpretability, since there is no significant correlation between the training loss of PWG and the score variation in the detector. Chen et al. [33] separated adversarial examples from genuine ones by a masking operation at the feature-level. However, the masking operation is manually designed, and is dependent on the dimensionality of the input features.

To address the aforementioned issues of attacker-dependent, feature-dimensionality-dependent and manual selection, in this article, we propose to detect adversarial examples by a *learnable mask detector* (LMD). It takes score variation as an indicator, and calculates the score variation by a masking operation on complex spectrogram features. Specifically, it assumes that short-time fourier transform (STFT) disperses manually-added adversarial perturbation uniformly from the time domain to the time-frequency domain. Naturally, due to the robustness of the ASV model to noise, masking insignificant time-frequency bins of the complex spectrograms has a large impact on adversarial examples, and a small impact on genuine examples. Based on the above assumption and observation, we aim to learn an optimal mask matrix by a neural network, and then utilize the absolute discrepancy of ASV scores before and after the masking operation to detect the adversarial examples.

It is worth noting that (i) LMD only requires the genuine examples for training, so it is attacker-independent; (ii) LMD transforms the masked complex spectrograms to speech signals in the time domain by the inverse short-time fourier transform (iSTFT), thus it becomes feature dimensionality-independent; (iii) LMD obtains the mask matrix by a neural network automatically, instead of designed manually; (iv) further, LMD calculates the score variation of the detection as part of the training loss of the neural network, which makes the detection and training closely related. We conducted experiments on two SOTA ASV models with diverse adversarial examples, and obtained an excellent detection performance. For example, detection equal error rates (EER) of no more than 5.9% and 10.1% are achieved on the ECAPA\_TDNN ASV and the Fast-ResNet34 ASV, respectively, in a noisy and blended detection scenarios.

Our contributions are summarized as follows:

- We propose a mask-based and attacker-independent detector, named LMD, which effectively mitigated the threat posed by adversarial examples to ASV systems. To demonstrate the advantage of learning a mask matrix through a neural network as LMD, we also propose a manually designed masking complex spectrogram (MCS) method as a baseline.
- We conducted experiments on two SOTA ASVs with abundant attackers. The two ASVs, which behave as either victims or defenders, are derived from two representative frameworks, i.e. stage-wise ASV and end-to-end ASV. The attackers cover three kinds of generation algorithms, and act as either an impostor or an evader to the ASVs in both white-box and black-box attacks.
- Inspired by [9], we evaluated the performance of a number of detectors under a given SNR budget. The experiments were also conducted in a scenario where the adversarial examples of a single attacker with different parameter settings were mixed, and the corresponding genuine examples were added with white-noise at the same SNR. Experimental results show that our proposed method outperforms the SOTA baselines in terms of the detection EER at an SNR budget of 37 dB and the above.

The rest of the article is organized as follows: Section II describes some preliminaries, including a brief introduction of ASV and three adversarial attack algorithms. Section III introduces our proposed methods. Section IV shows the experimental settings and evaluation metrics, while the results are discussed in Section V. Finally, Section VI hands concluding remarks.

## II. PRELIMINARIES

### A. Automatic Speaker Verification

ASV aims to confirm whether an utterance is pronounced by a specified speaker. Deep-learning-based ASV consist of a speaker embedding extractor (including feature engineering, encoder network, and temporal pooling module), a training objective function, and a similarity scoring back-end [1]. An encoder network first extracts frame-level speaker embeddings from acoustic feature sequences, e.g. logarithmic filter-banks (LogFBank). Then, segment-level speaker features are obtained

by the cascading of a pooling module and a feed-forward network. Finally, either classification-based or verification-based objective functions are used to train the above frame-level and segment-level speaker embedding extractors jointly.

To demonstrate the generalizability of the proposed method to different ASV systems, we adopt two representative training objective functions, i.e. *additive angular margin softmax* (AAM-Softmax) [4] and *angular prototypical* [7], for the victim ASV systems. In the test phase of ASV, we determine whether a test utterance  $\mathbf{x}^t$  and an enrollment utterance  $\mathbf{x}^e$  belong to the same speaker by comparing the similarity of their speaker embeddings with a predefined threshold  $\eta$ . The test phase is formulated as:

$$s = \mathbf{S}(f(\mathbf{x}^t), f(\mathbf{x}^e); \theta) \underset{H_0}{\overset{H_1}{\geq}} \eta, \quad (1)$$

where  $\mathbf{S}(\cdot; \theta)$  denotes the well-trained ASV model  $\mathbf{S}$  with parameters  $\theta$ ,  $f(\cdot)$  is an acoustic feature extractor, and  $H_1$  represents the hypothesis of  $\mathbf{x}^t$  and  $\mathbf{x}^e$  belonging to the same speaker, and  $H_0$  is the opposite hypothesis of  $H_1$ ,  $s$  is the similarity score of the two embeddings. The higher the similarity score is, the more likely the hypothesis  $H_1$  is true.

### B. Audio Adversarial Attack

Audio adversarial attack refers to an emerging technique that artificially generates slight noise and blends it into genuine speech, so as to make a speech signal processing system behave wrongly according to the goal of the attacker [8].

In terms of how much knowledge of the system is exposed to the attacker, we consider two attack scenarios: *white-box* and *black-box* attacks respectively. In the white-box attack scenario, the attacker has access to the full knowledge of the victim model, and can optimize the adversarial noise with the help of gradient from the victim model. In the black-box attack scenario, we consider the transfer-based cross-model attacker, who uses the adversarial examples generated by a substitute ASV model to attack the victim ASV model.

In terms of the goal of a attacker, we consider both *impersonation* and *evasion* types of attackers in this article. There are two kinds of trials in a realistic ASV system, i.e. target trials and non-target trials. A target (or non-target) trial regards the test utterance  $\mathbf{x}^t$  and the enrollment utterance  $\mathbf{x}^e$  come from the same (or different) speakers. Therefore, there are two types of misclassification, which delivers two kinds of attackers: (i) a non-target trial is misclassified as a target trial, and (ii) a target trial is misclassified as a non-target trial. We refer to these two attackers as adversarial impersonation and adversarial evasion, respectively [9]. The adversarial impersonation (or evasion) aims to generate an adversarial test utterance, which will be judged by the victim ASV model as a target (or non-target) trial of the enrollment utterance.

In this paper, we employ two gradient-based attackers, which are the basic iterative method (BIM) [34] and PGD [34], and an optimization-based attacker: Carlini Wanger (CW) [35], to craft adversarial example  $\tilde{\mathbf{x}}^t$  for the test utterance  $\mathbf{x}^t$ . We describe each attacker in detail as follows.

1) *BIM*: It is an attacker that generates adversarial examples in a multi-step. At each iteration, it obtains the gradient of the similarity score with respect to the input utterance  $\mathbf{x}_n$  and adds a perturbation of step  $\alpha$  along the gradient direction, followed by a cropping operation. The BIM attacker searches an adversarial example via the following formula:

$$\mathbf{x}_{n+1} = \text{Clip}_{\mathbf{x}^t, \epsilon}(\mathbf{x}_n + k\alpha \text{sign}(\nabla_{\mathbf{x}_n} \mathbf{S}(\mathbf{x}_n; \theta, f))), \quad (2)$$

where

$$k = \begin{cases} 1, & \text{if } \mathbf{x}^e \text{ and } \mathbf{x}^t \text{ contribute to a non-target trial} \\ -1, & \text{if } \mathbf{x}^e \text{ and } \mathbf{x}^t \text{ contribute to a target trial} \end{cases}$$

represents adversarial impersonation and adversarial evasion, respectively, and  $n = 0, 1, \dots, N$ , with  $N$  as the number of iterations,  $\epsilon = N\alpha$  constrains the magnitude of the perturbation,  $\mathbf{x}_n$  is initialized by the test utterance, i.e.  $\mathbf{x}_0 = \mathbf{x}^t$  (note that,  $\mathbf{x}^t$  is not normalized),  $\text{Clip}_{\mathbf{x}^t, \epsilon}(\cdot)$  denotes an element-wise clipping function which ensures the constraint  $\|\mathbf{x}_n - \mathbf{x}^t\|_\infty \leq \epsilon$ , and  $\mathbf{S}(\cdot; \theta, f)$  denotes a function to calculate the similarity score in (1) when the enrollment utterance  $\mathbf{x}^e$  is given. At the end of the  $N$  iterations of the BIM attacker, an adversarial example  $\tilde{\mathbf{x}}^t$  is found as  $\mathbf{x}_N$ .

2) *PGD*: It is essentially the same as BIM, but it initializes the perturbation to a random point in the  $L_p$  norm ball and replaces the cropping operation in (2) by the projection function. Instead of continuing to use the  $L_\infty$  norm in BIM, we adopt its counterpart of  $L_2$  norm in the PGD attacker to increase the diversity of the adversarial examples. The adversarial example  $\tilde{\mathbf{x}}^t$  is also found as  $\mathbf{x}_N$  via:

$$\mathbf{x}_{n+1} = \Pi_{\mathbf{x}^t + \mathcal{S}, \epsilon} \left( \mathbf{x}_n + k\alpha \frac{\nabla_{\mathbf{x}_n} \mathbf{S}(\mathbf{x}_n; \theta, f)}{\|\nabla_{\mathbf{x}_n} \mathbf{S}(\mathbf{x}_n; \theta, f)\|_2} \right), \quad (3)$$

where  $k, n, N, \alpha$  and  $\epsilon$  are defined in (2),  $\Pi_{\mathbf{x}^t + \mathcal{S}, \epsilon}(\cdot)$  represents a function of mapping the input into the sphere of  $L_2$  norm, which ensures the constraint  $\|\mathbf{x}_n - \mathbf{x}^t\|_2 \leq \epsilon$ .

3) *CW*: It is an optimization-based approach. It aims to get the minimum perturbation  $\delta^*$  for a successful attack and crafts an adversarial example by  $\tilde{\mathbf{x}}^t = \mathbf{x}^t + \delta^*$ ,

$$\delta^* = \min_{\delta} \frac{\|\delta\|_2}{\sqrt{L}} + c\mathcal{J}(\mathbf{x}^t + \delta), \quad (4)$$

where  $L$  is the length of the input test utterance  $\mathbf{x}^t$ , the normalized  $L_2$  distance, a.k.a the root mean square (RMS) distance, of the perturbation is adopted to eliminate the effect of signal duration [9], and  $c$  is a hyperparameter to balance the imperceptibility and aggressiveness of the adversarial perturbation, which is found by a binary search procedure. The optimization objective of the aggressiveness  $\mathcal{J}(\cdot)$  is defined as:

$$\mathcal{J}(\cdot) = \begin{cases} \max(0, -\mathbf{S}(\cdot; \theta, f) + (\eta + \kappa)), & \text{impersonation} \\ \max(0, \mathbf{S}(\cdot; \theta, f) - (\eta - \kappa)), & \text{evasion} \end{cases} \quad (5)$$

where  $\eta$  is a decision threshold and  $\kappa$  is a confidence value.

Finally, we summarize the adversarial attackers to the two ASV models that will be used in this article as in Table I, which covers most types of attacks in literature.



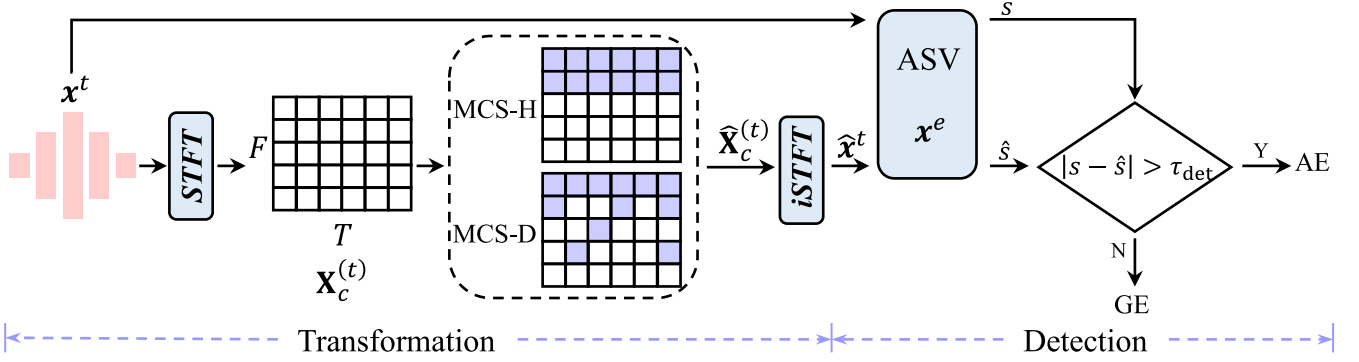


Fig. 1. Pipeline of the Masking Complex Spectrogram (MCS) detection method. The symbols  $x^t$  and  $\mathbf{X}_c^{(t)}$  denote the original test utterance and its complex spectrogram features respectively, and  $\hat{x}^t$ ,  $\hat{\mathbf{X}}_c^{(t)}$  are the corresponding transformed ones. The ASV score variation  $|s - \hat{s}|$  after the masking operation is used to identify whether the input utterance  $x^t$  is an adversarial example (AE) or a genuine example (GE).

TABLE I

TWELVE TYPES OF ATTACKERS ADOPTED IN THIS PAPER. EACH OF THE ATTACKER IS COMPOSED BY AN ALGORITHM, A TYPE OF PRIOR KNOWLEDGE, AND AN ATTACK GOAL FROM THE OPTIONS LISTED IN THE TABLE

Algorithm formulation	BIM ( $L_\infty$ ) 6 settings for $N$	PGD ( $L_2$ ) 6 settings for $N$	CW (RMS) 6 settings for $\kappa$
Prior knowledge	White-box attack		Black-box attack
Attack goals	Impersonation		Evasion

### III. METHODS

In this section, we first present the motivation of the proposed method in Section III-A, then present the framework of the proposed method in Section III-B, and finally present two implementations of the framework in Sections III-C and III-D respectively.

#### A. Motivations

Although adversarial examples seriously threaten the security of ASV, detection-based adversarial defense methods can effectively alleviate this threat. Based on the assumption that adversarial perturbations are uniformly distributed in acoustic features, Chen et al. [33] proposed Masking LogFBank features (MLFB) to detect adversarial examples. More specifically, masking as many insignificant speech features as possible will have a small impact on genuine examples and a large impact on adversarial examples, and thus utilize the variation of similarity scores after the masking operation to detect adversarial examples. However, MLFB has two problems: (i) *Non-universal*. Since MLFB performs masking operation directly on the input feature of an ASV system, its manually selected threshold is related to the dimensionality of the input feature. Moreover, when the dimensionality of the input feature decreases, which means the granularity of the features becomes coarser, MLFB may fail. (ii) *Hand-crafted mask*. MLFB masks the time-frequency bins of the input feature, either at high frequencies (MLFB-H) or using one-order difference (MLFB-D), both of which rely on human experience and lead to sub-optimal detection performance.

To address the above two shortcomings, we make improvements from two aspects respectively. For the non-universal problem, we perform ideal binary masking (IBM) operation on the complex spectrogram of the input, instead of performing it on the input speech features directly. Then, we detect adversarial examples by the recovered utterance, which is obtained by the iSTFT operation from the masked complex spectrogram. In this way, the hyperparameters are de-correlated with the dimensionality of the input features. For the hand-crafted mask problem, we attempt to obtain the mask matrix by a neural network instead of designing it manually, and replace the IBM matrix by either the ideal ratio masking (IRM) matrix or the approximate ideal binary masking (AIBM) matrix.

#### B. Framework

The pipeline of the proposed method contains two steps: transformation and detection, as shown in Fig. 1. The proposed two methods, i.e. MCS and LMD, differ in the transformation process, and share the same detection module.

1) *Transformation*: Given an input test utterance  $x^t$ , we first obtain its complex spectrogram  $\mathbf{X}_c^{(t)}$  by the STFT operation,

$$\mathbf{X}_c^{(t)} = g(x^t; \phi), \quad (6)$$

where  $\mathbf{X}_c^{(t)} \in \mathbb{C}^{F \times T}$  with  $F$  and  $T$  representing the number of frequency bins and frames respectively, and  $g(\cdot; \phi)$  represents the STFT operator with parameters  $\phi$ , such as frame length, frame shift, and number of points of the fast fourier transform. Then we use  $\mathbf{X}_c^{(t)}$  to calculate a mask matrix  $\mathbf{M}$  by either MCS or LMD, and perform the masking operation on the complex spectrogram  $\mathbf{X}_c^{(t)}$  via:

$$\hat{\mathbf{X}}_c^{(t)} = \mathbf{M} \odot \mathbf{X}_c^{(t)}, \quad (7)$$

where  $\hat{\mathbf{X}}_c^{(t)}$  is the masked complex spectrogram, and  $\odot$  denotes the element-wise product operator. Finally, the transformed utterance  $\hat{x}^t$  is obtained by:

$$\hat{x}^t = g^{-1}(\hat{\mathbf{X}}_c^{(t)}; \phi), \quad (8)$$

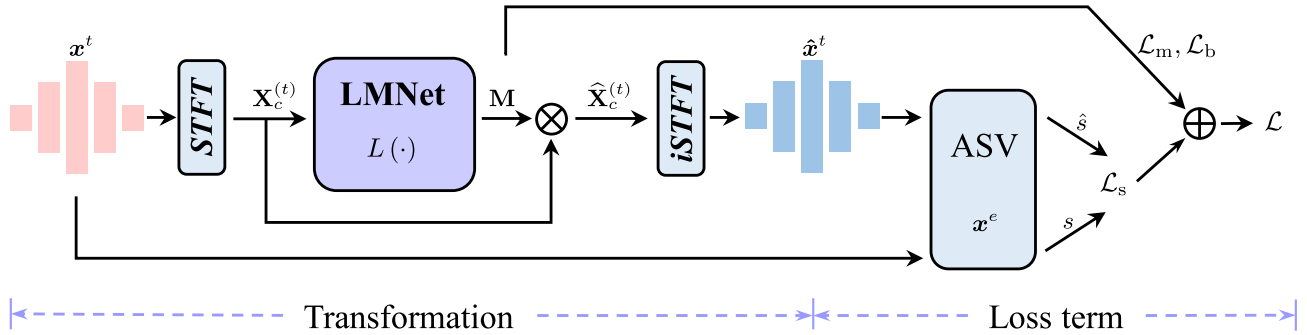


Fig. 2. Training process of the Learnable Mask Detector (LMD). Given a genuine utterance  $\mathbf{x}^t$ , the loss function  $\mathcal{L}$  in (16) takes the corresponding transformed utterance  $\hat{\mathbf{x}}^t$  and the mask matrix  $\mathbf{M}$  to train the learnable mask network (LMNet)  $L(\cdot)$ . The forward (black solid lines) and the gradients backward (red dashed lines) propagation process are shown. After the transformed utterance  $\hat{\mathbf{x}}^t$  is obtained by the well-trained LMNet  $L(\cdot)$ , we begin the detection process in Fig. 1.

where  $g^{-1}(\cdot; \phi)$  is the iSTFT operator with the same parameters  $\phi$  in (6).

2) *Detection*: After the transformation process of MCS or LMD to  $\mathbf{x}^t$ , the transformed utterance  $\hat{\mathbf{x}}^t$  is obtained. Then, two similarity scores are calculated by:

$$s = \mathbf{S}(\mathbf{x}^t, \mathbf{x}^e; \theta, f), \quad (9)$$

$$\hat{s} = \mathbf{S}(\hat{\mathbf{x}}^t, \mathbf{x}^e; \theta, f). \quad (10)$$

Finally, the proposed method compares the score variation  $v = |s - \hat{s}|$  with a detection threshold  $\tau_{\text{det}}$ . When  $v > \tau_{\text{det}}$ , the test utterance  $\mathbf{x}^t$  is detected as an adversarial example, otherwise, it is considered as a genuine example.

### C. Masking Complex Spectrogram

MCS only uses the magnitude  $\mathbf{X}_m^{(t)}$  of the complex spectrogram to calculate a mask matrix  $\mathbf{M} \in \mathbb{R}^{F \times T}$ . It masks complex spectrograms either at high frequencies (MCS-H) or using one-order difference (MCS-D).

MCS-H obtains the mask matrix by:

$$\mathbf{M} = \begin{bmatrix} \mathbf{1}_{(F-l) \times T} \\ \mathbf{0}_{l \times T} \end{bmatrix}, \quad (11)$$

where  $l$  is the length of the masking, and the symbols  $\mathbf{1}_{a \times b}$  (or  $\mathbf{0}_{a \times b}$ ) denotes an all one (or zero) matrix with  $a$  rows and  $b$  columns.

MCS-D masks the time-frequency bins whose absolute values of the one-order difference along the frequency axis is smaller than a masking threshold  $\xi$ :

$$\mathbf{M}_{i,j} = \begin{cases} 1, & \text{if } \left| \mathbf{X}_m^{(t)}(i+1,j) - \mathbf{X}_m^{(t)}(i,j) \right| > \xi \\ 0, & \text{if } \left| \mathbf{X}_m^{(t)}(i+1,j) - \mathbf{X}_m^{(t)}(i,j) \right| \leq \xi \end{cases}, \quad (12)$$

$$\forall i = 1, 2, \dots, F-1, \quad \forall j = 1, 2, \dots, T$$

where the subscripts  $i$  and  $j$  are the coordinates of the frequency axis and time axis, respectively. To make the mask matrix the same size as  $\mathbf{X}_c^{(t)}$  in (6), we further concatenate an all-zero matrix  $\mathbf{0}_{1 \times T}$  at the highest frequency, i.e.,  $\mathbf{M}_{F,j} = \mathbf{0}$ ,  $\forall j = 1, \dots, T$ .

### Algorithm 1: Training Procedure of LMD.

**Input:** The training data  $\mathcal{D}^t$ , the validation data  $\mathcal{D}^v$ , and the defensive ASV model  $\mathbf{S}(\cdot; \theta, f)$ .

**Output:** The well-trained LMNet  $L(\cdot)$  with parameters  $\Psi^*$ .

- 1 Initialize the hyperparameters  $m$ ,  $\lambda_s$ , and  $\lambda_b$ ;
- 2 **while** the number of training iterations **do**
- 3      $\mathbf{X}^t \leftarrow$  minibatch of  $q$  samples from  $\mathcal{D}^t$ ;
- 4      $\mathbf{X}^e \leftarrow$  randomly sampling the utterances of the same speaker with  $\mathbf{X}^t$  from  $\mathcal{D}^t$ ;
- 5      $\mathbf{M}, \hat{\mathbf{X}}^t \leftarrow$  Propagate the minibatch data  $\mathbf{X}^t$  forward the LMNet  $L(\cdot)$  as shown in Fig. 2;
- 6     Compute the loss function (16), as Loss  $\leftarrow \frac{1}{q} \sum_i \mathcal{L}(\mathbf{X}^t, \mathbf{X}^e, \mathbf{M}, \hat{\mathbf{X}}^t; m, \lambda_s, \lambda_b, \mathbf{S})$ ;
- 7     Minimize the loss function to update  $L(\cdot)$ ;
- 8     **if** reach the validation iteration interval **then**
- 9         Compute the loss (16) from the validation data  $\mathcal{D}^v$ , denoted as validation loss, and update the best parameters  $\Psi^*$  based on the validation loss;
- 10    **end**
- 11 **end**

### D. Learnable Mask Detector

As mentioned in Section III-A, the LMD detection method improves MCS by learning  $\mathbf{M}$  automatically. It is worthy noting that (i) the learnable mask network (LMNet) of LMD only uses genuine examples for training, so it is insensitive to the parameters and types of adversarial examples, i.e. attacker-independent, and (ii) LMD obtains strong interpretability, since the training and detection phases of LMD are closely related. Fig. 2 illustrates the training process of LMD. We describe its transformation process and training loss for the masking generation as follows.

1) *Transformation Process*: As shown in the left part of Fig. 2, there are two important differences between the transformation of LMD and MCS. First, the complex spectrogram feature are explicitly divided into real and imaginary parts, as  $\mathbf{X}_c^{(t)} \in \mathbb{R}^{F \times T \times 2}$ . Second, the mask matrix  $\mathbf{M}$  with the same size of  $\mathbf{X}_c^{(t)}$  is obtained by the well-trained LMNet  $L(\cdot)$ .

2) *Training Loss*: The right part of Fig. 2 describes the loss function of LMD. The design of the training loss is based on the assumption that adversarial perturbations are uniformly distributed in the feature space (e.g. the complex spectrograms), which makes us believe that the more the time-frequency bins are masked, the more likely the adversarial examples are to fail. However, when more time-frequency bins are masked, the discriminability of the ASV to the genuine examples decreases as well.

To address the above contradictory effects simultaneously, we expect to mask out as much as possible the time-frequency bins that contain little speaker information. Three loss terms are designed for this purpose:

The first loss term  $\mathcal{L}_s$  is the score variation, which measure the amount of speaker information contained in the masked time-frequency bins:

$$\mathcal{L}_s = \max(0, |s - \hat{s}| - m), \quad (13)$$

where  $m$  is a margin<sup>1</sup> of the hinge-loss, which is used to quantify the magnitude of the score variation, and the score  $s$  is the cosine similarity of the speaker embeddings of the test utterance  $\mathbf{x}^t$  and the enrollment utterance  $\mathbf{x}^e$ , and  $\hat{s}$  is the cosine similarity of the speaker embeddings of the transformed utterance  $\hat{\mathbf{x}}^t$  and  $\mathbf{x}^e$ .

The second loss term  $\mathcal{L}_b$  is the binary penalty for an AIBM matrix:

$$\mathcal{L}_b = \|\mathbf{M} \odot (\mathbf{1} - \mathbf{M})\|_2^2, \quad (14)$$

where the symbol  $\mathbf{1}$  represents an all one matrix of the same shape as  $\mathbf{M}$ . The binary penalty loss term will force the elements of the mask matrix to either converge to 0 or converge to 1, i.e., an AIBM matrix will be achieved.

The third loss term  $\mathcal{L}_m$  is an  $L_1$  norm of the mask matrix, which represents the severity of the masking operation:

$$\mathcal{L}_m = \|\mathbf{M}\|_1. \quad (15)$$

Finally, we propose to train LMNet by minimize the following loss function:

$$\mathcal{L} = \mathcal{L}_m + \lambda_s \mathcal{L}_s + \lambda_b \mathcal{L}_b, \quad (16)$$

where  $\lambda_s$  and  $\lambda_b$  are the hyperparameters used to balance the three loss terms. See Algorithm 1 for the complete training process of LMD.

#### IV. EXPERIMENTAL SETUP

##### A. Datasets

All of our experiments were conducted on the VoxCeleb dataset [36], which contains over one million utterances from 7,363 speakers of different ethnicities, accents, professions, and ages. The VoxCeleb datasets are automatically collected from interview videos uploaded to YouTube, and the speech segments were contaminated with real-world noise. The two victim ASV models were trained on the development set of VoxCeleb2 [37]

and evaluated on the cleaned up version of the original verification test list, i.e. *VoxCeleb1-test*, which consists of 37,611 trials from 40 speakers.

Without loss of generality, we randomly selected 1,000 trials from the original test list, denoted as the attack list *VoxCeleb1-attack*, to generate the adversarial examples. The randomly selected attack list include 500 target trials and 500 non-target trials. We also constructed an evaluation list *VoxCeleb1-eval* based on the attack list to evaluate the performance of attackers and detectors. The enrollment utterances of the evaluation list were randomly replaced with utterances of the same speaker in the test set of *VoxCeleb1*, but all utterances in the attack list were excluded.

Note that our proposed methods, MCS and LMD, were trained on the *VoxCeleb1-dev* dataset, which do not have overlapped speakers with the *VoxCeleb1-test* list. Moreover, *VoxCeleb1-dev* was divided into a training subset  $\mathcal{D}^t$  and a validation subset  $\mathcal{D}^v$  with a ratio of 19:1.

##### B. Experimental Settings

1) *Victim ASV Systems*: Different ASV models are characterized by different network structures, pooling strategies and training objectives. In this study, we used two ASV models as the victim. The first one is the ECAPA\_TDNN<sup>2</sup> [38] with a classification-based loss (AAM-Softmax [4]) and the attentive statistical pooling. The second one is the Fast-ResNet34<sup>3</sup> with a verification-based loss (Angular Prototypical [7]) and attentive average pooling. They adopted the same acoustic feature extractor: a hamming window of width 25 ms with a step size of 10 ms was used to partition speech signals into frames, and a 80-dimensional LogFBank followed by cepstral mean and variance normalization (CMVN) were extracted as the acoustic features. Online data augmentation, such as perturbing speed, superimposed disturbance, and simulating reverberation were adopted in the training process. In addition, they all used cosine similarity as the back-end scoring. The system decision threshold  $\eta$  is picked to be the threshold corresponding to the EER on the *VoxCeleb1-test*.

2) *Attackers*: We generated adversarial examples for three attackers based on the attack list *VoxCeleb1-attack*. For the BIM and PGD attackers, with the step size  $\alpha = 1$  and  $\alpha = 300$  fixed respectively, we generated adversarial examples for each value of the maximum iterations  $N$ , and constructed the *adversarial trial set*  $\mathcal{A}_i$ , where  $i = 1, 2, \dots, 6$ , and  $N = 5, 10, 20, 50, 100, 200$ . For the CW attacker, with the maximum number of binary search and iterations,  $N_{bs} = 9$  and  $N = 100$ , respectively, we also constructed adversarial trial set  $\mathcal{A}_i$  for each value of the confidence  $\kappa$ , where  $\kappa = 0.0, 0.1, 0.2, 0.3, 0.4, 0.5$ . We denotes the mixture set of the adversarial trials, i.e. *adversarial mixture set*, crafted by the BIM attacker as  $\mathcal{A}_{BIM} = \{\mathcal{A}_i \mid i = 1, 2, \dots, 6\}$ . For the PGD and CW attackers,  $\mathcal{A}_{PGD}$  and  $\mathcal{A}_{CW}$  were defined similarly with  $\mathcal{A}_{BIM}$ . In addition, the corresponding *genuine trial set*  $\mathcal{G}_i$  was constructed by adding the Gaussian

<sup>1</sup>Unless specified otherwise, the margin  $m$  is set to 0.05 in our LMD.

<sup>2</sup>[Online]. Available: <https://github.com/wenet-e2e/wespeaker>

<sup>3</sup>[Online]. Available: [https://github.com/clovaai/voxceleb\\_trainer](https://github.com/clovaai/voxceleb_trainer)

---

**Algorithm 2:** Method for Searching the Hyperparameters of MCS.
 

---

**Input:** The training data  $\mathcal{D}^t$ .

**Output:** The optimal hyperparameter  $p$ .

```

1 Initialize  $\lambda_s = 10, \lambda_b = 0, m = 0.1$  in (16),  $p_{\text{lower}} = 0,$ 
   $p_{\text{upper}} = 257$  for MCS-H and  $p_{\text{upper}} = 10^5$  for MCS-D;
2 while the maximum number of search has not been reached or
   $|p_{\text{upper}} - p_{\text{lower}}| \geq 1$  do
3   Divide the interval  $[p_{\text{lower}}, p_{\text{upper}}]$  into four equal parts, and
  obtains three parameters  $p_1, p_2, p_3$  with an ascending order;
4   Load a minibatch data from  $\mathcal{D}^t$ , and randomly select
  utterances of the same speaker as the enrollment;
5   Calculate the loss value corresponding to the three obtained
  parameters by (16), denoted as  $L_1, L_2, L_3$ . Note that the
  mask matrix  $\mathbf{M}$  and transformed utterance  $\hat{\mathbf{x}}^t$  were crafted
  by MCS-H or MCS-D;
6   if  $L_1$  is the smallest loss of the three then
7      $p_{\text{upper}} \leftarrow p_2$ 
8   else if  $L_3$  is the smallest loss of the three then
9      $p_{\text{lower}} \leftarrow p_2$ 
10  else
11     $p_{\text{lower}} \leftarrow p_1, p_{\text{upper}} \leftarrow p_3$ 
12  end
13 end

// Note the results are rounded.
Result:  $(p_{\text{lower}} + p_{\text{upper}}) / 2$ 

```

---

TABLE II

STATISTICAL RESULTS OF THE SEARCHED HYPERPARAMETERS OF MCS-H AND MCS-D OVER TEN INDEPENDENT RUNS OF ALGORITHM 2 ON THE VoxCeleb1-Dev DATASET. THE MEANS OF THE HYPERPARAMETERS WERE ADOPTED IN OTHER EXPERIMENTS

mean $\pm$ std	ECAPA_TDNN + AAM-Softmax	Fast-ResNet34 + Angular Prototypical
MCS-H $\rightarrow l$	79 $\pm$ 2	120 $\pm$ 10
MCS-D $\rightarrow \xi$	643 $\pm$ 71	1164 $\pm$ 285

white-noise to the original clean utterances in the attack list, which aims to obtain the same SNR as the adversarial utterances in  $\mathcal{A}_i$ . The black-box attacker employed in this article is the transfer-based cross-model attacker, i.e., the adversarial example generated by one substitute ASV is used to attack another victim ASV.

3) *Defenders:* The baseline detectors are the Vocoder, GL-mel, and GL-lin respectively, all of which followed the settings in [32]. They also utilize the score variation for detection, and the difference is that the phase reconstruction transformation are performed on the input utterances by vocoders, such as the PWG model. The settings of the masking length  $l$  and masking threshold  $\xi$  for the proposed MCS-H and MCS-D are shown in Table II, which were determined by Algorithm 2. The LMNet of the proposed LMD, which uses the network structure of DCCRN [39], aims to obtain a mask matrix with high generalization by the complex convolution. The complex spectrogram was extracted as the input feature by a hanning window of width 25 ms plus a step size of 10 ms and the convolutional STFT. The batch size was set to 32 and the length

of each audio clip was set to 500 frames. The Adam optimizer with an initial learning rate of 0.002 was used to train the LMNet  $L(\cdot)$  guided by the loss in (16), where the hyperparameter  $\lambda_s$  was set to 1. The hyperparameter  $\lambda_b$  in (16) controls the type of the mask matrix<sup>4</sup>. The learning rate was decayed by 0.9 times for every 1,000 steps. A total of 30 K iterations were trained, and the optimal model was selected based on the validation data  $\mathcal{D}^v$  with a validation interval of 1,000 steps.

### C. Evaluation Metrics

To evaluate the harmfulness of the attackers, we employ the attack success rate (ASR), normalized minimum detection cost function (minDCF) of the victim ASV with  $p = 0.01$  and  $C_{\text{miss}} = C_{\text{fa}} = 1$  [40], and SNR, as the evaluation metrics.

To evaluate the performance of the detectors, we adopt EER and the detection success rate (DSR) with different given false acceptance rate (FAR), as the evaluation metrics.

Before introducing the evaluation metrics, we first define the score variation set for the genuine trial set and adversarial trial set, respectively. For the genuine trial set  $\mathcal{G} = \{(\mathbf{x}_i^t, \mathbf{x}_i^e) \mid i = 1, 2, \dots, I\}$  defined in Section IV-B2, a score variation set  $\mathcal{V}_{\text{gen}}$  after the masking operation can be obtained by:

$$v_i = \left| \mathbf{S}(\mathbf{x}_i^t, \mathbf{x}_i^e; \boldsymbol{\theta}, f) - \mathbf{S}(\hat{\mathbf{x}}_i^t, \mathbf{x}_i^e; \boldsymbol{\theta}, f) \right| \quad (17)$$

where  $v_i \in \mathcal{V}_{\text{gen}}$  with  $i = 1, 2, \dots, I$ , and  $\hat{\mathbf{x}}_i^t$  represents that the test utterance  $\mathbf{x}_i^t$  is transformed by our mask-based detection methods. For the adversarial trial set  $\mathcal{A} = \{(\tilde{\mathbf{x}}_i^t, \mathbf{x}_i^e) \mid i = 1, 2, \dots, I\}$ , its score variation set  $\mathcal{V}_{\text{adv}}$  is also calculated by (17), except that  $\mathbf{x}_i^t$  and  $\hat{\mathbf{x}}_i^t$  are replaced by the corresponding adversarial example  $\tilde{\mathbf{x}}_i^t$  and the transformed adversarial example, respectively.

Then the evaluation metric EER is defined by:

$$\text{EER}_{\text{det}} = \text{FAR}_{\text{det}}(\tau_{\text{eer}}) = \text{FRR}_{\text{det}}(\tau_{\text{eer}}), \quad (18)$$

where

$$\text{FAR}_{\text{det}}(\tau) = \frac{|\{v_i > \tau \mid v_i \in \mathcal{V}_{\text{gen}}\}|}{|\mathcal{V}_{\text{gen}}|}, \quad (19)$$

$$\text{FRR}_{\text{det}}(\tau) = \frac{|\{v_i \leq \tau \mid v_i \in \mathcal{V}_{\text{adv}}\}|}{|\mathcal{V}_{\text{adv}}|}, \quad (20)$$

are the FAR and the false rejection rate (FRR), respectively, of the detector given a threshold  $\tau$ ,  $|\mathcal{S}|$  represents the number of the elements in the set  $\mathcal{S}$ . After manually given a tolerable FAR of detection, denoted as  $\text{FAR}_{\text{given}}$ , we define the evaluation metric DSR as:

$$\text{DSR} = \frac{|\{v_i > \tau_{\text{det}} \mid v_i \in \mathcal{V}_{\text{adv}}\}|}{|\mathcal{V}_{\text{adv}}|}, \quad (21)$$

where

$$\tau_{\text{det}} = \underset{\tau}{\text{argmin}} |\text{FAR}_{\text{det}}(\tau) - \text{FAR}_{\text{given}}|, \quad (22)$$

is the detection threshold for  $\text{FAR}_{\text{given}}$ . We also evaluated the DSR of detectors under the adversarial mixture sets, i.e.  $\mathcal{A}_{\text{BIM}}$ ,  $\mathcal{A}_{\text{PGD}}$  and  $\mathcal{A}_{\text{CW}}$ .

<sup>4</sup> $\lambda_b = 15$  indicates the LMD-AIBM, and  $\lambda_b = 0$  indicates the LMD-IRM.



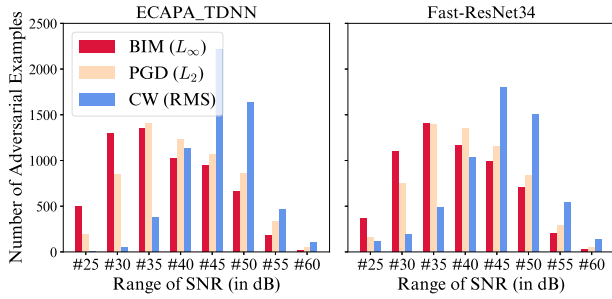


Fig. 3. Statistical results of the number of adversarial examples in a SNR range. The ECAPA\_TDNN and Fast-ResNet34 act as the victim ASV. The symbol “# $n$ ” means the range of “[ $n, n + 5$ )”.

Finally, the detection EER is defined over a given SNR budget, as in [9]. Specifically, we assume an evaluator function  $\mathbf{E}(\mathcal{A}, \mathcal{G})$  that computes the detector EER given the adversarial trial set  $\mathcal{A}$  and genuine trial set  $\mathcal{G}$  with  $I$  trials. We assume that  $\mathbf{p}_{\text{adv}} = [p_{\text{adv},1}, \dots, p_{\text{adv},I}]^T$  and  $\mathbf{p}_{\text{gen}} = [p_{\text{gen},1}, \dots, p_{\text{gen},I}]^T$  are vectors describing the SNRs of the corresponding trial sets respectively. Then, for each value of the SNR budget  $b$  that we want to evaluate, we obtain an adversarial trial set  $\mathcal{A}(b)$  and a genuine trials set  $\mathcal{G}(b)$  by:

$$t_i = \begin{cases} (\tilde{\mathbf{x}}_i^t, \mathbf{x}_i^e), & \text{if } p_{\text{adv},i} \geq b \text{ or } p_{\text{gen},i} \geq b \\ \emptyset, & \text{otherwise} \end{cases}, \quad (23)$$

where  $t_i \in \mathcal{A}(b)$  with  $i = 1, 2, \dots, I$ , and  $\mathcal{G}(b)$  is composed of the corresponding trials in  $\mathcal{G}$ . The detector EER for budget  $b$  is obtained by evaluating  $\mathbf{E}(\mathcal{A}(b), \mathcal{G}(b))$ .

## V. RESULTS AND DISCUSSIONS

In this section, we first present the performance of the attackers in Section V-A so as to show their great threat to the ASV systems, then present the performance of the detectors against different attackers in Section V-B so as to show how much the threat is mitigated. Finally, we present several additional experiments in Section V-C.

### A. Performance of the Attackers

Fig. 3 shows the number of adversarial examples at different ranges of SNR. The SNR of adversarial examples generated by the CW attacker is higher than that of the BIM and PGD attackers. Note that the SNRs are calculated on the adversarial mixture sets, i.e.  $\mathcal{A}_{\text{BIM}}$ ,  $\mathcal{A}_{\text{PGD}}$  and  $\mathcal{A}_{\text{CW}}$ .

Fig. 4 illustrates the performance of the three attackers. ECAPA\_TDNN achieves an EER and minDCF of 1.25% and 0.1372 on the test list VoxCeleb1-test. Similarly, Fast-ResNet34 achieves 1.97% and 0.2553 respectively. The above results indicate that the two ASV models are SOTA. In the case of the white-box attacks, the BIM attacker and CW attacker achieves an ASR of 97% at a SNR of 35 dB and 42 dB, respectively. The PGD attacker achieves similar performance with BIM. All of the three attackers leads to an minDCF of 0.99+

even at a SNR of 45 dB. In the case of the transfer-based black-box attack, the attackers generally deliver better performance on the Fast-ResNet34 ASV than on the ECAPA\_TDNN ASV. The BIM, PGD and CW attacker achieve their maximum ASR of 23%, 20% and 7% on Fast-ResNet34, respectively. These results show that the attackers highly threaten the SOTA ASV models.

### B. Performance of the Detectors

The performance of our proposed detectors is shown below, where  $\lambda_b = 15$  indicates the LMD-AIBM method, and  $\lambda_b = 0$  indicates the LMD-IRM method. The difference between the two methods lies in the type of their masking matrices.

Tables III and IV comprehensively show the EER performance of the detectors in the white-box and black-box scenarios, respectively. Note that the EER is calculated in a noisy situation by evaluating  $\mathbf{E}(\mathcal{A}_i, \mathcal{G}_i)$  for the three attackers, where  $i = 1, 2, \dots, 6$  represent the six different parameter settings. The victim ASV and the defended ASV are always consistent. Several conclusions can be drawn: (i) From the perspective of the white-box attack scenario, our proposed LMD method outperforms the baseline methods in the most detection conditions. For example, LMD-AIBM achieves a detection EER of 0.8% and 1.5% on ECAPA\_TDNN and Fast-ResNet34, respectively, when encountering the BIM attacker with  $N = 50$ , which is 38% and 11% higher than Vocoder. (ii) MCS-H possesses the worst detection performance due to its coarse mask matrix, while MCS-D achieves comparable performance to GL-mel and GL-lin by finely designing the mask matrix, which shows the effectiveness of our mask-based idea, despite the mask matrices of MCS-H and MCS-D are both manually crafted. (iii) Further, we desire to leverage the neural network to learn an AIBM matrix or an IRM matrix for detection. LMD-AIBM performs better than LMD-IRM when the perturbation intensity is high, while LMD-IRM performs better than LMD-AIBM when the perturbation intensity is low. (iv) From the perspective of the black-box attack scenario, our proposed LMD method achieves the optimal performance in almost all detection conditions. The results on ECAPA\_TDNN and Fast-ResNet34 are basically the same, obtaining an EER of 37% when encountering the BIM attacker and the PGD attacker, and an EER of 44% when encountering the CW attacker. There is still great development potential to separate adversarial examples in the black-box scenario for the detection-based passive defense approaches. In addition, we believe that the main reason for the low detection performance of black-box attacks is that the large number of failed adversarial examples pulls down the adversarial score variation, thus leading to higher detection EER. Therefore, we conducted an ablation experiment in Section V-C7.

Fig. 5 shows the impact of the SNR budget on the detector performance. From the figure, we draw the following conclusions: (i) the performance of all detectors gradually drops as the SNR budget decreases. In the range of SNR budget of 50 dB to 40 dB, our LMD-IRM maintains an EER of 3% to 9% and outperforms the comparison methods. (ii) In the range of SNR budget of 35 dB to 25 dB, our LMD methods achieve comparable



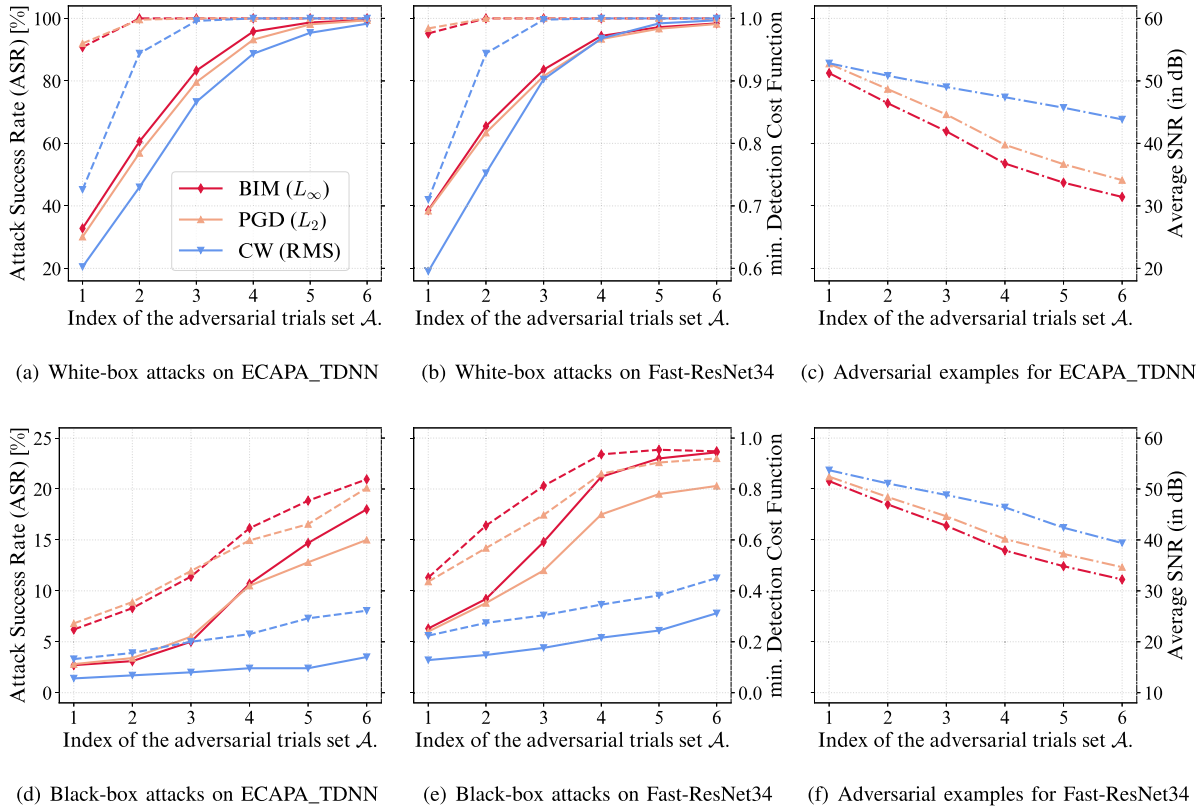


Fig. 4. Attack performance of the three attackers: BIM, PGD and CW in terms of ASR, minDCF, and mean SNR, where **ASR is described in solid line, the minDCF with  $p = 0.01$  is described in dashed line, and the mean SNR is described in dotted line.** The captions of the subfigures “(a), (b), (d), (e)” are concise. For example, “Black-box attacks on ECAPA\_TDNN” means that the victim and substitute ASV models are ECAPA\_TDNN and Fast-ResNet34, respectively. The subfigures “(c)” and “(f)” depict the average SNR of the adversarial examples. Note that, the EER of the ECAPA\_TDNN ASV model with the AAM-Softmax loss on the test list VoxCeleb1-test is 1.25%; the EER of the Fast-ResNet34 ASV model with the Angular Prototypical is 1.97%.

TABLE III  
DETECTION EER OF THE DETECTORS AGAINST THREE ATTACKERS IN THE WHITE-BOX ATTACK SCENARIO ON THE TWO ASVS

EER (%) $\searrow$	Attacker $\rightarrow$	BIM ( $L_\infty, \alpha = 1$ )						PGD ( $L_2, \alpha = 300$ )						CW (RMS, $N = 100$ )							
		$N/\kappa \rightarrow$	5	10	20	50	100	200	5	10	20	50	100	200	0	0.1	0.2	0.3	0.4	0.5	
ECAPA_TDNN + AAM-Softmax	Vocoder		11.50	4.30	<b>1.70</b>	1.30	2.00	2.00	12.50	4.70	<b>2.10</b>	<b>1.10</b>	1.10	1.10	1.50	11.60	5.20	2.70	1.80	<b>0.50</b>	<b>0.30</b>
	GL-mel		28.00	14.00	5.60	2.50	2.20	2.80	28.60	14.60	6.80	2.90	2.10	2.00	26.40	16.00	9.00	4.50	2.70	1.40	
	GL-lin		20.80	9.80	3.90	2.20	2.10	2.90	21.60	10.30	5.10	2.30	2.10	2.50	19.20	10.50	5.00	3.00	1.80	1.50	
	MCS-H		24.70	14.00	8.10	5.80	5.90	6.70	26.30	15.30	9.00	5.80	5.60	6.00	23.40	15.20	9.90	6.70	4.10	3.00	
	MCS-D		15.50	8.40	5.20	2.90	3.20	2.70	15.30	8.50	4.90	3.10	2.80	2.40	15.50	9.10	5.40	3.50	2.60	1.90	
	LMD ( $\lambda_b = 15$ )		16.10	6.70	2.30	<b>0.80</b>	<b>0.90</b>	<b>1.20</b>	15.90	6.70	2.50	<b>1.10</b>	<b>0.80</b>	<b>1.10</b>	14.50	7.50	3.80	1.50	0.90	<b>0.30</b>	
	LMD ( $\lambda_b = 0$ )		<b>7.70</b>	<b>3.60</b>	3.00	4.10	4.60	6.70	<b>7.90</b>	<b>4.40</b>	3.80	4.20	6.10	7.00	<b>9.10</b>	<b>4.20</b>	<b>2.00</b>	<b>1.20</b>	0.90	1.10	
Fast-ResNet34 + Angular Prototypical	Vocoder		12.70	<b>5.00</b>	<b>2.20</b>	1.70	2.00	<b>2.00</b>	12.00	<b>5.10</b>	<b>1.90</b>	1.80	1.70	<b>1.80</b>	17.70	8.60	<b>2.90</b>	<b>1.20</b>	<b>0.80</b>	<b>1.60</b>	
	GL-mel		23.50	11.50	5.20	2.40	3.10	3.40	24.10	10.60	5.20	2.90	3.20	3.90	28.50	17.30	9.20	4.00	2.30	3.20	
	GL-lin		16.50	7.60	3.50	2.60	2.90	4.10	15.60	6.80	3.60	2.90	3.80	4.10	22.30	11.90	5.30	2.70	2.30	5.10	
	MCS-H		30.70	18.70	11.50	9.40	10.60	12.00	30.50	18.50	12.10	9.80	10.60	11.90	35.80	24.50	16.20	10.40	8.50	10.20	
	MCS-D		18.80	9.50	5.60	3.50	3.30	3.60	18.40	8.80	5.00	3.40	3.30	3.50	24.60	14.80	9.00	5.60	4.70	6.60	
	LMD ( $\lambda_b = 15$ )		17.30	6.70	2.90	<b>1.50</b>	<b>1.90</b>	<b>2.00</b>	15.80	6.40	2.30	<b>1.50</b>	<b>1.60</b>	2.30	20.30	11.20	4.60	2.10	0.90	<b>1.30</b>	
	LMD ( $\lambda_b = 0$ )		<b>8.80</b>	<b>5.00</b>	5.40	8.80	11.50	13.60	<b>9.90</b>	6.40	6.30	9.40	12.50	15.50	<b>12.90</b>	<b>6.00</b>	3.10	3.30	6.50	12.20	

performance with Vocoder with an EER fluctuating around 5%. (iii) Our proposed LMD-IRM outperforms all baseline detectors at a SNR budget higher than 37 dB.

Table V shows the variation of the detector accuracy with the  $FAR_{\text{given}}$ . From the figure, it can be concluded that, as the  $FAR_{\text{given}}$  decreases from 5% to 0.1%, the detection threshold will increase meanwhile, and more adversarial examples will be missed, so the accuracy of all detectors drops. Moreover, Vocoder reaches the top accuracy while its DSR drops

from 96% to 86%, on the contrary, our proposed LMD-AIBM achieves the runner-up accuracy while its DSR drops from 93% to 84%.

In Fig. 6, the detection error tradeoff (DET) curve is used to evaluate the detector performance more delicately than Table V. Experimental results on the ECAPA\_TDNN ASV system indicate that our LMD-IRM detector is always ahead of Vocoder, and both of them obtain an EER of less than 5%. Experimental results on the Fast-ResNet34 ASV system show

TABLE IV  
DETECTION EER OF THE DETECTORS AGAINST THREE ATTACKERS IN THE BLACK-BOX ATTACK SCENARIO ON THE TWO ASVs

EER (%) $\searrow$	Attacker $\rightarrow$	BIM ( $L_\infty$ , $\alpha = 1$ )						PGD ( $L_2$ , $\alpha = 300$ )						CW (RMS, $N = 100$ )						
		Victim Model $\downarrow$		$N/\kappa \rightarrow$	5	10	20	50	100	200	5	10	20	50	100	200	0	0.1	0.2	0.3
ECAPA_TDNN + AAM-Softmax	Vocoder		49.60	50.60	47.30	43.80	41.20	41.10	50.60	49.20	49.00	44.40	41.90	42.40	49.80	50.50	49.70	49.60	50.90	48.20
	GL-mel		54.70	55.10	56.70	54.10	52.50	52.20	54.50	56.30	57.00	55.20	54.70	53.20	53.30	54.30	54.60	54.40	53.40	53.00
	GL-lin		53.30	53.50	51.00	45.50	45.00	43.10	54.30	54.30	51.80	48.90	46.90	43.80	53.40	54.30	55.70	55.30	53.50	51.60
	MCS-H		53.80	52.50	51.80	52.80	52.70	51.10	53.00	53.10	53.00	52.90	52.60	51.20	52.90	53.50	53.20	54.10	54.20	52.30
	MCS-D		52.10	50.40	45.80	39.60	37.80	36.10	52.80	51.20	46.80	40.80	38.50	<b>37.20</b>	52.30	52.50	52.30	52.40	51.00	50.10
	LMD ( $\lambda_b = 15$ )		53.50	51.40	47.60	42.00	39.00	39.80	53.30	51.90	47.80	42.80	40.50	39.70	54.10	54.00	54.30	53.50	51.70	51.00
	LMD ( $\lambda_b = 0$ )		<b>46.60</b>	<b>43.70</b>	<b>40.00</b>	<b>36.30</b>	<b>36.50</b>	<b>36.00</b>	<b>45.80</b>	<b>44.00</b>	<b>41.20</b>	<b>40.00</b>	<b>39.70</b>	38.30	<b>47.60</b>	<b>46.10</b>	<b>44.40</b>	<b>43.40</b>	<b>43.50</b>	<b>44.60</b>
Fast-ResNet34 + Angular Prototypical	Vocoder		51.60	48.60	46.40	44.80	44.00	44.30	50.80	49.60	47.50	45.70	45.60	44.20	50.40	50.20	51.00	49.20	49.90	50.00
	GL-mel		52.50	51.70	50.90	46.70	47.60	47.80	51.90	51.40	50.50	48.30	48.10	48.50	52.20	52.60	53.10	52.50	52.30	51.50
	GL-lin		49.00	47.10	42.60	38.20	39.50	40.00	50.30	48.70	45.40	41.90	40.50	40.80	50.00	50.30	48.90	48.10	48.50	47.30
	MCS-H		50.60	50.80	50.50	49.90	51.20	51.90	51.40	50.80	51.20	51.40	51.90	52.60	51.80	52.00	51.80	51.50	52.00	51.50
	MCS-D		50.50	48.70	42.90	38.00	<b>36.50</b>	<b>37.90</b>	50.20	49.10	44.10	39.80	<b>37.20</b>	<b>38.70</b>	50.30	49.50	48.80	47.80	46.60	45.10
	LMD ( $\lambda_b = 15$ )		50.80	48.90	43.70	39.60	39.60	41.10	51.50	49.40	46.70	40.00	40.40	40.90	53.50	54.30	52.30	50.60	48.60	49.00
	LMD ( $\lambda_b = 0$ )		<b>43.80</b>	<b>40.60</b>	<b>38.30</b>	<b>36.90</b>	38.30	38.60	<b>46.10</b>	<b>43.30</b>	<b>40.70</b>	<b>39.70</b>	38.90	40.60	<b>47.20</b>	<b>47.10</b>	<b>45.30</b>	<b>44.10</b>	<b>43.70</b>	<b>42.80</b>

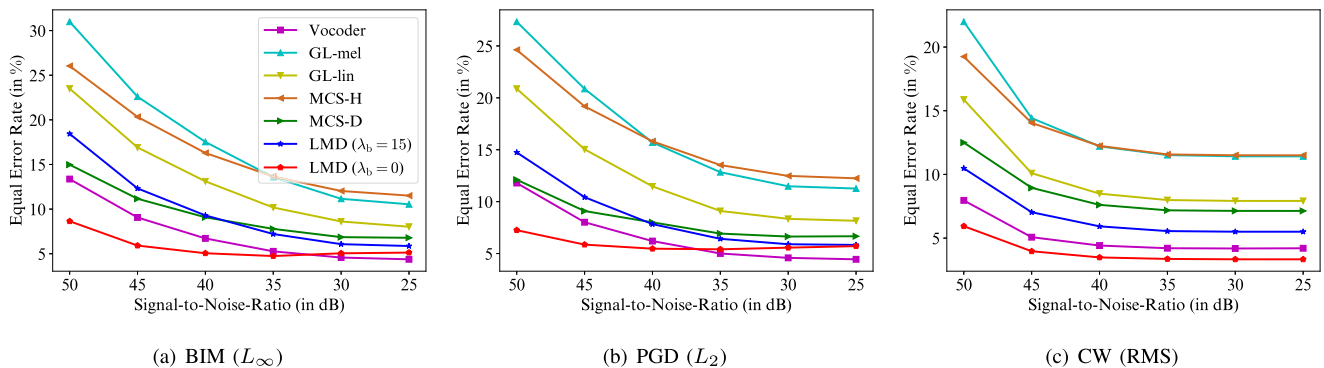


Fig. 5. Detection EER of the detectors along with the SNR budget. The performance of detectors was evaluated on ECAPA\_TDNN and the three adversarial mixture sets ( $\mathcal{A}_{BIM}$ ,  $\mathcal{A}_{PGD}$  and  $\mathcal{A}_{CW}$ ) in the white-box attack scenario.

TABLE V  
DSR OF THE DETECTORS ALONG WITH  $FAR_{GIVEN}$

DSR (%)	$FAR_{given}(\%)$	5.0	1.0	0.5	0.1
ECAPA_TDNN + AAM-Softmax	Vocoder	<b>95.98</b>	<b>93.15</b>	<b>92.02</b>	<b>88.80</b>
	GL-mel	86.98	80.82	78.55	60.72
	GL-lin	90.47	83.35	80.08	69.22
	MCS-H	82.37	73.30	70.58	52.28
	MCS-D	91.73	78.97	71.50	66.15
	LMD ( $\lambda_b = 15$ )	93.92	91.07	<b>89.97</b>	<b>88.37</b>
	LMD ( $\lambda_b = 0$ )	<u>94.85</u>	90.72	89.20	82.35
Fast-ResNet34 + Angular Prototypical	Vocoder	<b>95.25</b>	<b>91.05</b>	<b>89.05</b>	<b>85.67</b>
	GL-mel	88.72	82.73	80.10	69.63
	GL-lin	92.58	86.37	83.37	78.20
	MCS-H	74.98	66.82	62.90	51.53
	MCS-D	88.30	74.67	71.45	57.02
	LMD ( $\lambda_b = 15$ )	<u>93.38</u>	<u>89.50</u>	<u>88.67</u>	<u>84.07</u>
	LMD ( $\lambda_b = 0$ )	89.28	82.03	79.68	72.17

The performance of detectors was evaluated on the adversarial mixture set of  $\mathcal{A}_{BIM}$ . The best results are in bold, while the runner-up results are underlined.

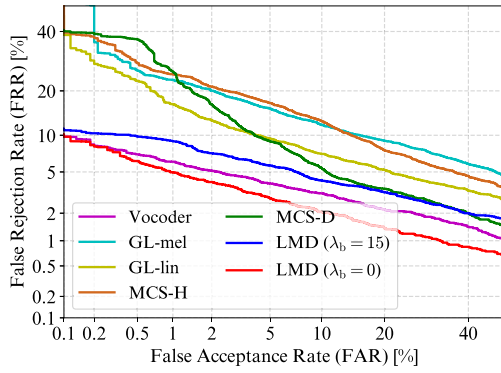
that Vocoder always leads our LMD detectors with an EER of less than 10%.

### C. Ablation Studies

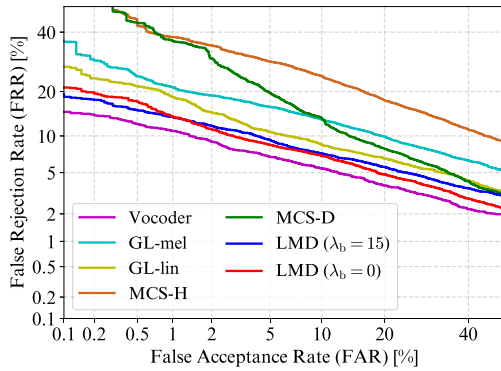
1) *Effects of the Hyperparameter  $m$  on Performance:* The hyperparameter  $m$  in (13), i.e. the score margin, controls the amount of speaker information to be masked out. To study its effect on the performance of LMD, we trained the

LMD-AIBM and LMD-IRM detectors with  $m$  set to 0.05, 0.1 and 0.15, respectively, and evaluated them with  $\mathcal{A}_{BIM}$  on the two ASVs. We draw several conclusions from the results in Fig. 7 as follows. (i) In the initial naive state where a random mask matrix is generated, LMD obtains an EER of about 16%, which proves the effectiveness of our mask-based idea again. (ii) When the training of LMD proceeds, the detection EER gradually decreases and becomes smooth after 20 K steps with an EER of 5% to 8%. (iii) For LMD-AIBM (blue lines) and LMD-IRM (red lines), the hyperparameter  $m$  performs optimally on 0.05 and 0.1, respectively. However, we set  $m$  to 0.05 in all experiments for the sake of controlling variables. (iv) LMD-AIBM is difficult to be trained successfully when  $m$  is set large, especially when Fast-ResNet34 acts as the victim model, which could be mitigated by increasing  $\lambda_b$ .

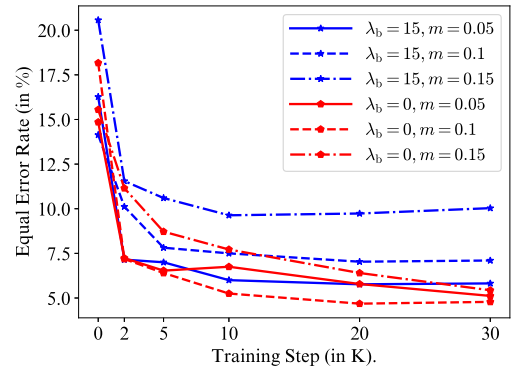
2) *Interpretation of the Principles of LMD:* To explain why our proposed LMD method is effective, we present the boxplot of the score variations in Fig. 8 for the analysis. Specifically, the boxplot depicts the distribution of the score variations of the detectors when confronted with the adversarial examples and genuine examples. From the figure, it can be seen that, our LMD ensures that the score variations for the genuine examples do not exceed  $m$ , and moreover, it makes the score variations for the adversarial examples as large as possible, which consequently gets the detection easier.



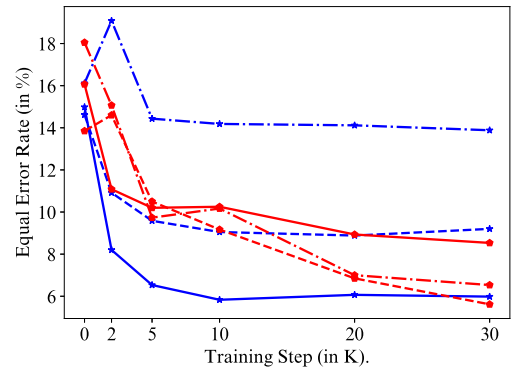
(a) ECAPA\_TDNN + AAM-Softmax



(b) Fast-ResNet34 + Angular Prototypical



(a) ECAPA\_TDNN + AAM-Softmax



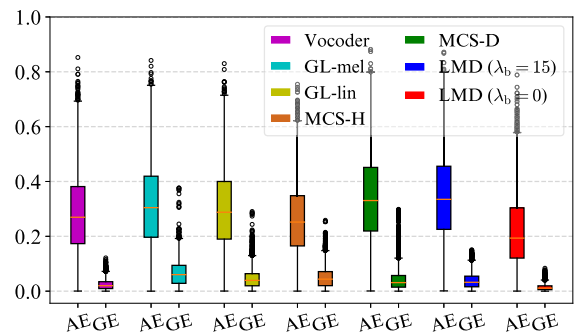
(b) Fast-ResNet34 + Angular Prototypical

Fig. 6. DET curves of the detectors on the adversarial mixture set of  $\mathcal{A}_{CW}$ .TABLE VI  
ADDITIONAL PURIFICATION EFFECTS OF OUR LMD

EER (%) $\searrow$	Attacker $\rightarrow$	Clean	BIM ( $L_\infty$ )	PGD ( $L_2$ )	CW (RMS)
Victim Model $\downarrow$	$N/\kappa \rightarrow$	-	200	200	0.5
ECAPA_TDNN + AAM-Softmax	No-Defense	1.25	100.00	99.80	98.40
	Vocoder	<b>1.20</b>	70.80	64.40	11.80
	LMD ( $\lambda_b = 15$ )	1.80	<b>30.00</b>	<b>25.00</b>	<b>3.60</b>
	LMD ( $\lambda_b = 0$ )	1.40	97.60	97.20	42.00
Fast-ResNet34 + Angular Prototypical	No-Defense	1.97	100.00	100.00	10.00
	Vocoder	<b>1.80</b>	58.40	52.80	21.20
	LMD ( $\lambda_b = 15$ )	2.80	<b>20.00</b>	<b>18.80</b>	<b>8.20</b>
	LMD ( $\lambda_b = 0$ )	2.00	99.40	98.80	85.00

Fig. 9 further visualizes the spectrograms of the original audio and transformed audio. From the figure, it can be seen that LMD-AIBM masks the low-energy regions and samples sparsely, while LMD-IRM masks most of the low-energy regions, which are consistent with our goal of masking the most time-frequency bins that contain little speaker information. Therefore, they reach large score variations for the adversarial examples, and small score variations for the genuine examples.

3) *Purification Effects of LMD*: The PWG-based Vocoder has also been utilized for the mitigation-based defense in [26]. Here we further explored the effectiveness of our LMD to purify the adversarial noise in Table VI, where we used the pre-trained model provided by Wu et al. [32] as a baseline, and employed EER of the victim ASV as the evaluation metric.

Fig. 7. Connection between the detection EER and training steps of our proposed LMD with different score margins, on the adversarial mixture set of  $\mathcal{A}_{BIM}$ .Fig. 8. Boxplot of the score variations of the adversarial mixture set  $\mathcal{A}_{CW}$  and genuine mixture set  $\mathcal{G}_{CW}$  for the seven detectors with the ECAPA\_TDNN ASV as the victim. AE and GE represent adversarial examples and genuine examples, respectively.

From the table, we see that LMD-AIBM achieves much better purification effect than LMD-IRM, because the  $L_1$  norm of the mask matrix measures the masking degree of LMD-AIBM more appropriately than LMD-IRM. Our LMD is designed to mask as many spectrogram bins as possible at the cost of little speaker information. Therefore, EER increases slightly on clean examples but decreases the most on adversarial examples. However, Vocoder behaves more like a speech enhancement module,

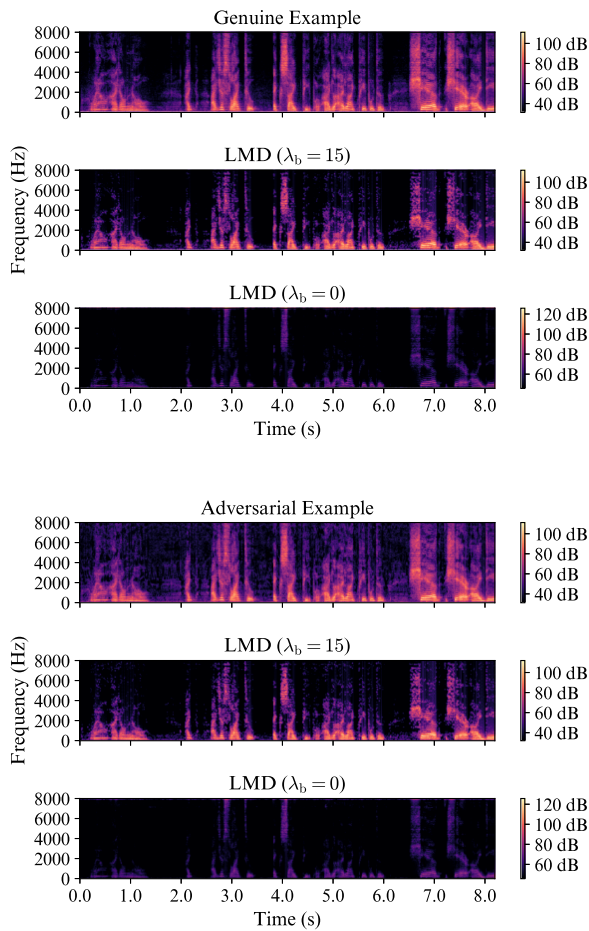


Fig. 9. Spectrograms of the original audio examples and their corresponding transformed audio examples obtained by our LMD. The genuine example is from `id10284/7yx9A0yZLYk/00010.wav` of VoxCeleb1. The hypothesis enrollment utterance of the adversarial example is from `id10305/gbTZ7k9e/Z0_00001.wav` of VoxCeleb1. The adversarial example was generated by the BIM attacker with  $N = 50$ .

TABLE VII  
DETECTOR PERFORMANCE AGAINST ADAPTIVE ATTACKERS

Attacker ↓	Detector ↓	ASR (%)		EER (%)
		trans. (w/o)	trans. (w/)	
BIM ( $L_\infty, N = 50$ )	Vocoder	69.20	83.70	12.50
	LMD-AIBM	3.50	82.80	<b>1.70</b>
PGD ( $L_2, N = 50$ )	Vocoder	64.70	80.80	12.10
	LMD-AIBM	2.80	73.90	<b>2.10</b>
CW (RMS, $\kappa = 0.3$ )	Vocoder	41.60	84.80	9.80
	LMD-AIBM	1.70	42.30	<b>5.60</b>

The term “trans (w/o)” means that the input is not transformed by the detector, while the term “trans (w/)” is the opposite. ECAPA\_TDNN is employed as the victim ASV.

where the input goes through a front-end noise reduction. Therefore, EER decreases on clean examples but the decrease in EER on adversarial examples is less apparent than LMD. Compared with Vocoder, the threat brought by adversarial examples are significantly mitigated by our LMD-AIBM.

4) *Encounter With Adaptive Attackers*: Table VII explores the performance of the detectors under the adaptive attack. The so-called adaptive attack means that the attacker can further

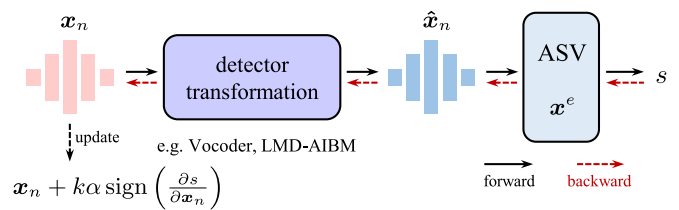


Fig. 10. Pipeline of the adaptive attacker for generating adversarial examples. The symbols  $x_n$  and  $\hat{x}_n$  denote the test utterance and the transformed utterance, respectively. The BIM attacker in (2) is used here as an example.

TABLE VIII  
DETECTOR PERFORMANCE UNDER MORE REALISTIC NOISES

Victim Model ↓	Noise Type →	noise			
		gauss	speech	music	
ECAPA_TDNN + AAM-Softmax	Vocoder	2.00	1.90	1.80	1.80
	LMD-AIBM (w/o)	<b>1.20</b>	1.10	<b>1.10</b>	1.40
	LMD-AIBM (w/)	1.60	<b>1.00</b>	<b>1.10</b>	<b>1.00</b>
Fast-ResNet34 + Angular Prototypical	Vocoder	<b>2.00</b>	2.00	2.30	2.40
	LMD-AIBM (w/o)	<b>2.00</b>	2.10	2.00	2.00
	LMD-AIBM (w/)	2.10	<b>1.60</b>	<b>1.90</b>	<b>1.90</b>

The term “LMD-AIBM (w/)” indicates the training was augmented with the noises from MUSAN, and “LMD-AIBM (w/o)” indicates no data-augmentation. BIM( $L_\infty, N = 200$ ) attacker is employed.

access the detector parameters to generate an adversarial example. Specifically, as shown in Fig. 10, adversarial examples are updated by utilizing the gradient of the score w.r.t. the test utterance. From the Table VII, two conclusions can be drawn: (i) the adaptive attacker cannot breach the system without a LMD-AIBM transformation, for example, the ASR drops from 82.8% to 3.5% under the BIM attacker. (ii) LMD-AIBM can also achieve a detection EER no higher than 5.6% under the adaptive attack. Our analysis reveals that the attackers can only breach the original victim system, or the hybrid system with the LMD-AIBM transformation, i.e., they cannot breach both systems simultaneously. Eventually, we utilize the score variation of the two systems to detect the adversarial examples, and these adaptive adversarial examples will still yield a large score variation, so the detection performance remains robust.

5) *Data Augmentation for LMD*: Table VIII explores the detection performance of the LMD-AIBM with and without data-augmentation against a variety of noises. Specifically, we employ the MUSAN corpus [41] for data-augmentation with a probability of 60%. First, the noise sample is cropped or padded (in wrap mode) to the target length, and then it is scaled to a random SNR between [25, 40] before being superimposed to the target speech. The detection EER is employed as the evaluation metric, i.e.,  $\mathbb{E}(\mathcal{A}, \mathcal{G})$ , where  $\mathcal{G}$  is constructed by adding noises to the original clean utterances, such as the Gaussian white-noise, or three types of noises from the MUSAN corpus. From the table, two conclusions can be drawn: (i) data-augmentation can further improve the detection performance of LMD-AIBM. (ii) The detection EER increases slightly for the type of noise that is unseen during the training, i.e., the Gaussian white-noise.

6) *Effect of Calibration on Performance*: In the previous sections, we only studied the situation where the ASV victim systems are un-calibrated, i.e. they simply produce the



TABLE IX

PERFORMANCE OF THE DETECTORS WHEN APPLYING THE WHITE-BOX ADVERSARIAL ATTACKERS ON THE ASV VICTIM SYSTEMS (EITHER CALIBRATED, I.E. “COSINE”, OR UN-CALIBRATED, I.E. “BCE”), WHERE THE ATTACKERS GENERATE ADVERSARIAL EXAMPLES EITHER FROM THE CALIBRATED ASV VICTIM SYSTEM OR FROM THE UN-CALIBRATED ASV VICTIM SYSTEM. THE ATTACKER BIM ( $L_\infty$ ,  $N = 50$ ) AND THE VICTIM ASV ECAPA\_TDNN ARE EMPLOYED. THE CALIBRATED SYSTEM OBTAINS AN act.DCF<sub>0.01</sub> OF 0.19 ON THE GENUINE EXAMPLES

ASV victim models ↓	White-box Attackers ↓	Attacker Perf.		Detector Perf. by EER (%)	
		ASR (%)	act.DCF	Vocoder	LMD-AIBM
Cosine	Cosine	95.80	-	1.30	0.90
Cosine	BCE	95.50	-	1.50	1.10
BCE	Cosine	79.50	0.77	1.30	0.90
BCE	BCE	78.90	0.78	1.50	1.10

similarity of two embeddings in terms of some measurement, like cosine similarity. However, the ASV systems are typically calibrated [42], [43] in their real-world applications, i.e. they transform the un-calibrated similarity score of two embeddings to a target posterior probability, denoted as a calibrated score. A common calibration function is the binary-cross-entropy (BCE) loss [9]. In this section, we will further study the situation where the ASV victim systems are calibrated.

For each of the above two ASV victim systems, we can also have two kinds of white-box attackers: one kind generates adversarial examples from a victim system with the un-calibrated loss, such as the “Cosine” similarity  $S(\cdot)$  in (2), and the other kind generates adversarial examples from a victim system with the calibrated loss, such as BCE. Finally, we have four “ASV-attacker” pairs.

In this section, we present the performance of the detectors on the evaluation environments of the above four “ASV-attacker” pairs in Table IX. From the table, three conclusions can be drawn: (i) the calibration does not affect the detection EER, due to the fact that only positive scaling and offset are performed on the scores, whereas we obtain the variation of log-likelihood-ratio for detection. (ii) The ASR decreases after the calibration, because the threshold corresponding to EER and the threshold of the Bayesian decision operate on different points. (iii) The generation losses of “Cosine” and “BCE” produce the equivalent adversarial examples in terms of both principle and experimental results. They show little difference in terms of ASR, act.DCF and detection EER.

7) *Exclusion of Failed Adversarial Examples:* In Tables III and IV, the adversarial examples that are failed to attack the ASV systems are taken into the account when reporting the performance of the detectors. However, they show in fact slight difference from the genuine examples from the perspective of not only the ASV systems but also human listeners, so as to the detectors. In this section, we study how the detectors perform when we exclude the failed adversarial examples.

Table X analyzes the performance of the detectors against the adversarial examples that can successfully attack the ASV system. The successful adversarial examples in the white-box scenario are able to move greatly away from the decision threshold, which results in an easy discrimination between adversarial and genuine examples. In contrast, the successful adversarial

TABLE X

DETECTION EER OF THE DETECTORS IN THE ABSENCE OF THOSE FAILED ADVERSARIAL EXAMPLES

EER (%) ↓	$N, \epsilon \rightarrow$	5	10	20	50	100	200	
		White-box		Vocoder	<b>2.74</b>	2.48	<b>1.20</b>	1.15
		LMD ( $\lambda_b = 15$ )	5.18	<b>2.31</b>	1.44	<b>0.84</b>	<b>0.91</b>	<b>1.11</b>
		LMD ( $\lambda_b = 0$ )	3.66	2.48	2.88	3.55	4.05	4.32
Black-box		Vocoder	<b>33.33</b>	38.71	42.00	30.84	29.25	30.00
		LMD ( $\lambda_b = 15$ )	44.44	35.48	32.00	28.97	29.25	28.33
		LMD ( $\lambda_b = 0$ )	40.74	<b>32.26</b>	<b>22.00</b>	<b>23.36</b>	<b>23.13</b>	<b>20.00</b>

The attacker BIM ( $L_\infty$ ) and the victim ASV ECAPA\_TDNN are employed.

examples in the black-box only slightly cross the decision threshold, and thus only achieve a detection EER of 20% at best. However, compared to Table IV, the performance of the detectors in the black-box scenario improve substantially after excluding those failed adversarial examples.

## VI. CONCLUSION

In this article, we have proposed a detection-based passive defense approach called LMD to detect adversarial example for ASV systems. It is attacker-independent and possesses high interpretability. First, it masks out the regions of complex spectrograms with little speaker information to introduce a large impact on adversarial examples, and small impact on genuine examples, respectively. Then, it identifies the adversarial examples by calculating the ASV score variations before and after the masking operation. Experimental results show that our proposed LMD achieves comparable performance with the SOTA baselines. Specifically, it achieves detection EERs of no more than 5.9% and 10.1% on the ECAPA\_TDNN ASV and Fast-ResNet34 ASV, respectively. LMD achieves a DSR of nearly 90% in the stringent setting of a given FAR of 1% when encountering the BIM attacker. In addition, we evaluated the detector performance against a given SNR budget. Experimental results on the ECAPA\_TDNN ASV show that LMD outperforms the baseline approaches at a SNR budget of higher than 37 dB. In an additional experiment, we find that the LMD-AIBM detector has the effect of purifying adversarial noise, which further alleviates the threat brought by the adversarial attacks.

## REFERENCES

- [1] Z. Bai and X.-L. Zhang, “Speaker recognition based on deep learning: An overview,” *Neural Netw.*, vol. 140, pp. 65–99, 2021.
- [2] 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.*, 2014, pp. 4052–4056.
- [3] 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.*, 2018, pp. 5329–5333.
- [4] Y. Liu, L. He, and J. Liu, “Large margin softmax loss for speaker verification,” in *Proc. Interspeech*, 2019, pp. 2873–2877.
- [5] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 4879–4883.
- [6] Z. Bai, X.-L. Zhang, and J. Chen, “Partial AUC optimization based deep speaker embeddings with class-center learning for text-independent speaker verification,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 6819–6823.

- [7] J. S. Chung et al., "In defence of metric learning for speaker recognition," in *Proc. Interspeech*, 2020, pp. 2977–2981.
- [8] C. Szegedy et al., "Intriguing properties of neural networks, 2013," *arXiv:1312.6199*.
- [9] J. Villalba, Y. Zhang, and N. Dehak, "X-vectors meet adversarial attacks: Benchmarking adversarial robustness in speaker verification," in *Proc. Interspeech*, 2020, pp. 4233–4237.
- [10] J. Lan, R. Zhang, Z. Yan, J. Wang, Y. Chen, and R. Hou, "Adversarial attacks and defenses in speaker recognition systems: A survey," *J. Syst. Architecture*, vol. 127, 2022, Art. no. 102526.
- [11] H. Tan, L. Wang, H. Zhang, J. Zhang, M. Shafiq, and Z. Gu, "Adversarial attack and defense strategies of speaker recognition systems: A survey," *Electronics*, vol. 11, no. 14, 2022, Art. no. 2183.
- [12] Y. Xie, Z. Li, C. Shi, J. Liu, Y. Chen, and B. Yuan, "Enabling fast and universal audio adversarial attack using generative model," in *Proc. Assoc. Advance. Artif. Intell.*, vol. 35, no. 16, 2021, pp. 14129–14137.
- [13] G. Chen et al., "Who is real bob? adversarial attacks on speaker recognition systems," in *Proc. IEEE Symp. Secur. Privacy*, 2021, pp. 694–711.
- [14] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 1962–1966.
- [15] X. Li, J. Zhong, X. Wu, J. Yu, X. Liu, and H. Meng, "Adversarial attacks on GMM I-vector based speaker verification systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 6579–6583.
- [16] Q. Wang, P. Guo, and L. Xie, "Inaudible adversarial perturbations for targeted attack in speaker recognition," in *Proc. Interspeech*, 2020, pp. 4228–4232.
- [17] Z. Li, Y. Wu, J. Liu, Y. Chen, and B. Yuan, "ADVPulse: Universal, synchronization-free, and targeted audio adversarial attacks via subsecond perturbations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2020, pp. 1121–1134.
- [18] W. Zhang et al., "Attack on practical speaker verification system using universal adversarial perturbations," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 2575–2579.
- [19] Y. Xie, Z. Li, C. Shi, J. Liu, Y. Chen, and B. Yuan, "Real-time, robust and adaptive universal adversarial attacks against speaker recognition systems," *J. Signal Process. Syst.*, vol. 93, no. 10, pp. 1187–1200, 2021.
- [20] X. Zhang, X. Zhang, X. Zou, H. Liu, and M. Sun, "Towards generating adversarial examples on combined systems of automatic speaker verification and spoofing countermeasure," *Secur. Commun. Netw.*, 2022.
- [21] Q. Wang, P. Guo, S. Sun, L. Xie, and J. H. Hansen, "Adversarial regularization for end-to-end robust speaker verification," in *Proc. Interspeech*, 2019, pp. 4010–4014.
- [22] H. Wu, S. Liu, H. Meng, and H.-Y. Lee, "Defense against adversarial attacks on spoofing countermeasures of ASV," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 6564–6568.
- [23] M. Pal, A. Jati, R. Peri, C.-C. Hsu, W. AbdAlmageed, and S. Narayanan, "Adversarial defense for deep speaker recognition using hybrid adversarial training," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 6164–6168.
- [24] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.
- [25] H. Zhang, L. Wang, Y. Zhang, M. Liu, K. A. Lee, and J. Wei, "Adversarial separation network for speaker recognition," in *Proc. Interspeech*, 2020, pp. 951–955.
- [26] S. Joshi, J. Villalba, P. Żelasko, L. Moro-Velázquez, and N. Dehak, "Study of pre-processing defenses against adversarial attacks on state-of-the-art speaker recognition systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 4811–4826, 2021.
- [27] H. Wu, X. Li, A. T. Liu, Z. Wu, H. Meng, and H.-Y. Lee, "Improving the adversarial robustness for speaker verification by self-supervised learning," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 202–217, 2021.
- [28] H. Wu, Y. Zhang, Z. Wu, D. Wang, and H.-Y. Lee, "Voting for the right answer: Adversarial defense for speaker verification," in *Proc. Interspeech*, 2021, pp. 4294–4298.
- [29] X. Li et al., "Investigating robustness of adversarial samples detection for automatic speaker verification," in *Proc. Interspeech*, 2020, pp. 1540–1544.
- [30] S. Joshi, S. Kataria, J. Villalba, and N. Dehak, "ADVEst: Adversarial perturbation estimation to classify and detect adversarial attacks against speaker identification," 2022, *arXiv:2204.03848*.
- [31] Z. Peng, X. Li, and T. Lee, "Pairing weak with strong: Twin models for defending against adversarial attack on speaker verification," in *Proc. Interspeech*, 2021, pp. 4284–4288.
- [32] H. Wu et al., "Adversarial sample detection for speaker verification by neural vocoders," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 236–240.
- [33] X. Chen, J. Yao, and X.-L. Zhang, "Masking speech feature to detect adversarial examples for speaker verification," in *Proc. IEEE Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2022, pp. 191–195.
- [34] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2016, *arXiv:1611.01236*.
- [35] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57.
- [36] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.
- [37] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.
- [38] B. Desplanques, J. Thienpondt, and K. Demuyne, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Interspeech*, 2020, pp. 3830–3834.
- [39] Y. Hu et al., "DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement," in *Proc. Interspeech*, 2020, pp. 2472–2476.
- [40] A. Nagrani et al., "VoxSRC 2020: The second VoxCeleb speaker recognition challenge," 2020, *arXiv:2012.06867*.
- [41] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," 2015, *arXiv:1510.08484*.
- [42] N. Brummer et al., "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 7, pp. 2072–2084, 2007.
- [43] N. Brümmer and E. De Villiers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," 2013, *arXiv:1304.2865*.



**Xing Chen** received the B.S. degree in electronic and communication engineering from Northwestern Polytechnical University, Xi'an, China, where he is currently working toward the M.S. degree. His research interests include speaker identification and adversarial defense.



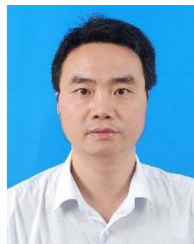
**Jie Wang** (Graduate Student Member, IEEE) received the B.S. degree in electronic and communication engineering from Northwestern Polytechnical University, Xi'an, China, where he is currently working toward the M.S. degree in electronic and communication engineering. His research interests include machine learning and data mining.



**Xiao-Lei Zhang** (Senior Member, IEEE) received the Ph.D. degree in information and communication engineering from Tsinghua University, Beijing, China, in 2012. He is currently a Full Professor with the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China. He was a Postdoctoral Researcher with the Perception and Neurodynamics Laboratory, The Ohio State University, Columbus, OH, USA. His research interests include speech processing, underwater acoustic signal processing, machine learning, statistical signal processing, and artificial intelligence. He is a Member of IEEE SPS and ISCA.



**Wei-Qiang Zhang** (Senior Member, IEEE) received the B.S. degree in applied physics from the University of Petroleum, Dongying, China in 2002, the M.S. degree in communication and information systems from the Beijing Institute of Technology, Beijing, China, in 2005, and the Ph.D degree in information and communication engineering from Tsinghua University, Beijing, in 2009. From 2016 to 2017, he was a Visiting Scholar with the Center for Computer Research in Music and Acoustics, Stanford University, Stanford, CA, USA. He is currently an Associate Professor with the Department of Electronic Engineering, Tsinghua University and the Head of the Speech and Audio Technology Laboratory (SATLab). His research interests include the area of speech and audio recognition and analysis, signal and information processing, and machine learning.



**Kunde Yang** received the B.S., M.S., and Ph.D. degrees from the Northwestern Polytechnical University (NPU), Xi'an, China, in 1996, 1999, and 2003, respectively. From 2006 to 2007, he was a Visiting Scholar with the School of Earth and Ocean Sciences, University of Victoria, Victoria, BC, Canada. Since 2003, he has been with the School of Marine Science and Technology, NPU, where he is currently a Full Professor and the Vice President of the School from 2018 to 2022. Since 2022, he has been with the Ocean Institute of NPU, where he is currently the President of the Ocean Institute. He has authored or coauthored about 180 papers indexed by SCI and has authorized about 70 patents. His research interests include underwater acoustical physics, signal processing, and ocean communication.