FTDKD: Frequency-Time Domain Knowledge Distillation for Low-Quality Compressed Audio Deepfake Detection

Bo Wang, *Member, IEEE*, Yeling Tang, Fei Wei, *Member, IEEE*, Zhongjie Ba, *Member, IEEE*, and Kui Ren, *Fellow, IEEE*

***Abstract*—In recent years, the field of audio deepfake detection has witnessed significant advancements. Nonetheless, the majority of solutions have concentrated on high-quality audio, largely overlooking the challenge of low-quality compressed audio in real-world scenarios. Low-quality compressed audio typically suffers from a loss of high-frequency details and time-domain information, which significantly undermines the performance of advanced deepfake detection systems when confronted with such data. In this paper, we introduce a deepfake detection model that employs knowledge distillation across the frequency and time domains. Our approach aims to train a teacher model with high-quality data and a student model with low-quality compressed data. Subsequently, we implement frequency-domain and time-domain distillation to facilitate the student model's learning of high-frequency information and time-domain details from the teacher model. Experimental evaluations on the ASVspoof 2019 LA and ASVspoof 2021 DF datasets illustrate the effectiveness of our methodology. On the ASVspoof 2021 DF dataset, which consists of low-quality compressed audio, we achieved an Equal Error Rate (EER) of 2.82%. To our knowledge, this performance is the best among all deepfake voice detection systems tested on the ASVspoof 2021 DF dataset. Additionally, our method proves to be versatile, showing notable performance on high-quality data with an EER of 0.30% on the ASVspoof 2019 LA dataset, closely approaching state-of-the-art results.**

***Index Terms*—Audio deepfake detection, low-quality compressed audio, knowledge distillation.**

I. Introduction

In RECENT years, we have witnessed the rapid emergence of Artificial Intelligence Generative Content (AIGC) in the literature. As one of the most popular techniques, Deepfake, which can be applied for synthesizing high-quality visual and audio contents, has been extensively applied, e.g. VALL-E [1] released by Microsoft, a novel speech synthesis model. However, while these techniques greatly boost entertainment applications, they also raise significant concerns regarding malicious attacks and associated negative impacts.

Speech synthesis technology, such as text-to-speech (TTS) or voice conversion (VC), can be misused for AI fraud or the spread of misleading information. For example, fake speech generated by these technologies can be used to impersonate voices for scams, and the dissemination of fake videos or speeches of famous person can cause significant social impact. With all such concerns, the detection of forged content is in crucial need and there is a great amount of work focusing on the detection of forged audio content.

To address the aforementioned concerns, researchers focus on developing effective and generalizable deepfake detectors [2], [3], [4]. Meanwhile, addressing the security of Automatic Speaker Verification (ASV) systems and risks of spoofing attacks, the ASVspoof and Audio Deep Synthesis Detection challenges were successfully held in the past decade [5], [6], [7], [8], [9], [10]. In recent years, some researchers have also begun to pay attention to partially fake audio and are dedicated to developing detectors capable of identifying and distinguishing manipulated segments within audio [11], [12], [13]. Currently, state-of-the-art deepfake audio detection models exhibit excellent performance on high-quality datasets. However, in real-life scenarios, e.g. in the context of social media, audio data is commonly compressed in a low-quality format with a loss of information, rendering challenges on deepfake audio detection. Generally speaking, there are two primary issues with low-quality compressed data.

Loss of frequency domain information: Since humans have a more sensitive perception of low-frequency components, many lossy compression algorithms, e.g. MP3 and OGG, intentionally discard high-frequency data to reduce the file size. As visualized in Fig. 1, there is a significant difference in the high-frequency components between the original speech and the compressed speech. Nevertheless, [14] observes that while the synthesized voice well resembles the low-frequency information comparing with real speeches, the artifacts in high-frequency parts are more distinct - but the compression algorithms may cause a loss on

Received 20 February 2024; revised 17 July 2024 and 24 September 2024; accepted 23 October 2024. Date of publication 7 November 2024; date of current version 21 November 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62076052 and Grant 62106037, in part by the Science and Technology Innovation Foundation of Dalian under Grant 2021JJ12GX018, and in part by the Application Fundamental Research Project of Liaoning Province under Grant 2022JH2/101300262. The associate editor coordinating the review of this article and approving it for publication was Prof. Zhizheng Wu. *(Corresponding author: Fei Wei.)*

Bo Wang and Yeling Tang are with the School of Information and Communication Engineering, Dalian University of Technology, Dalian 116081, China (e-mail: bowang@dlut.edu.cn; tyl20001219@163.com).

Fei Wei is with the Alibaba Group, Hangzhou, Zhejiang 311121, China (e-mail: feiwei@alibaba-inc.com).

Zhongjie Ba and Kui Ren are with the School of Cyber Science and Technology, Zhejiang University, Hangzhou, Zhejiang 310027, China (e-mail: zhongjieba@zju.edu.cn; kuiren@zju.edu.cn).

Digital Object Identifier 10.1109/TASLP.2024.3492796

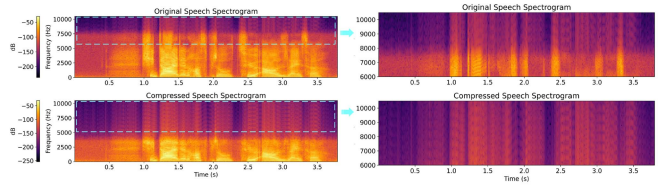


Fig. 1. The upper part of the figure displays the spectrogram of the original audio, while the lower part shows the spectrogram of the MP3 compressed audio. The left side of the figure contains the overall frequency spectrum ranging from 0 to 10,000 Hz. To observe the differences in high-frequency components in greater detail, the right side presents the high-frequency spectrum ranging from 6,000 to 10,000 Hz.

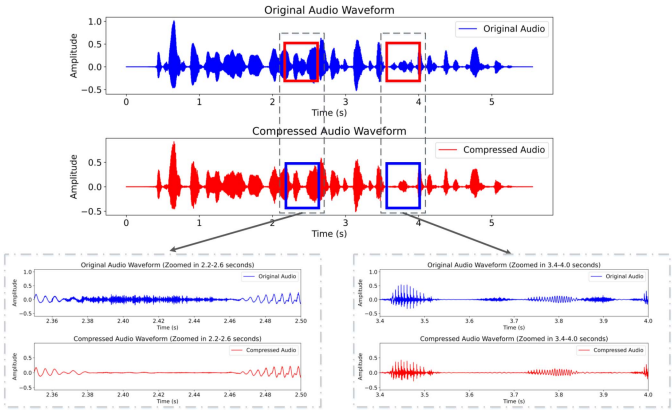


Fig. 2. The top half of the figure represents the overallwaveform plots of the original audio sample and its compressed copy, where blue indicates the waveform of the original audio and red indicates the waveform of the compressed audio. The boxed areas highlight the differences between them. The lower part of the figure displays an enlarged view of the differential areas.

this part of the information, resulting in the detection of forged speech after compression to be more challenging.

Loss of time domain information: Besides the aforementioned defects, lossy compression algorithms also involve quantization and clipping, resulting in the loss of time-domain details, rendering the speech signal to be smoother, particularly in regions with rapid changes. Fig. 2 visually illustrates the waveform graphs of the original audio sample and its compressed counterpart, and one may observe that the compressed audio exhibits a smoother waveform in certain segments, in other words, details are lost, such that detection model may suffer from performance drop.

In this paper, to address the issues mentioned above, we propose to conduct knowledge distillation in both frequency-domain and time-domain data to detect forged voice data in low quality. While knowledge distillation is a typical approach to convert deep learning models into lightweight versions, inspired by [15], [16], we adopt a data distillation approach, that is, we use high-quality data to train the teacher model and low-quality data to train the student model. Then, we apply frequency-domain and time-domain distillation to enable the student model to learn the frequency-domain and time-domain information regarding data compression loss from the teacher model, so as to improve the forgery detection performance of low-quality data. For time-domain distillation we use the sliced Wasserstein distance and contrast loss to evaluate time-domain feature differences, and for frequency-domain distillation we use the square of the Euclidean distance to evaluate frequency-domain feature differences, with details of the algorithms given in Section III of the paper. It's important to note that the high-quality data and low-quality data used to train the distillation model are paired, and we employ lossy compression algorithms to compress the high-quality dataset, yielding the corresponding low-quality compressed dataset.

Our contributions are summarized as follows:

- We propose a frequency domain knowledge distillation method. That is, we enable the student model to effectively learn the high-frequency information from the teacher model, which is lost by compression.
- We propose a time-domain knowledge distillation method. That is, we calculate the differences in inter-layer features between student and teacher models through Sliced Wasserstein Distance (SWD) and contrastive loss, enabling the student model to better learn the time-domain details lost in compression.
- We conducted experiments on the ASVspoof 2021 DF and the ASVspoof 2019 LA datasets, demonstrating the satisfactory results of our method over baseline models. Additionally, we perform compression over data from other datasets for extra tests, verifying the generalization capabilities of our proposed method.

II. Related Work

A. Audio Deepfake Detection

Currently, the main focus of research in fake speech detection is centered on the development of front-end features and back-end models. For front-end features, a substantial number of efforts [14], [17], [18] have underscored the significance of effective features for identifying fake attacks. The features employed in forged speech detection can be broadly classified into four categories: acoustic features [19], [20], raw audio [21], [22], paralinguistic features [23], [24], and self-supervised features [25], [26]. Commonly used handcrafted acoustic features include Linear Frequency Cepstral Coefficients (LFCC) [27], Mel Frequency Cepstral Coefficients (MFCC) [27], Constant Q Cepstral Coefficients (CQCC) [28], and others. Due to the use of a fixed window length in the Short-Time Fourier Transform (STFT) for traditional handcrafted features, there is often insufficient capture of rapidly changing temporal dynamics, particularly in transient and swiftly varying speech segments. In addition, there is a large amount of nonlinear information in speech signals that is difficult to capture with traditional linear processing methods. Therefore, given that hand-designed front-end features may miss a large amount of speech information, Tak et al. [29] advocated the direct use of raw audio as input to spoofing detection models. As speech synthesis technology advances, distinguishing between forged and genuine speech based solely on acoustic features becomes increasingly challenging. Some researchers advocate for the use of paralinguistic features to differentiate between genuine and forged speech. For instance, [30] suggested that audio deepfake techniques

fail to accurately synthesize natural emotional information, thus proposing the use of emotional information in speech for audio deepfake detection. Reference [23] focused on the duration of phonemes and pronunciation features for spoofing detection. Reference [31] discovered that speech synthesis systems are unable to synthesize human breathing sounds, leading to the proposal of employing breath detection techniques to support spoofing detection tasks. Another prominent feature extractor for fake audio detection is wav2vec [3], [26], which is grounded on a self-supervised method and is trained on a large corpus of unlabeled data. The features extracted by wav2vec demonstrate excellent generalization capabilities and can effectively differentiate between genuine and fake speech.

For back-end models, a wide collection of deep neural network-based classifiers have shown excellent performance in detecting fake speech. Several studies [32], [33] have introduced classifiers built on the ResNet architecture for this purpose. To improve the model's ability to generalize, Gao et al. [34] introduced the Res2Net structure, a variant designed to enhance performance further. This has led to numerous successful applications [35], [36] of the Res2Net architecture in the field of fake speech detection. Additionally, recent advancements have seen the development of end-to-end networks [37], [38] that integrate feature extraction and classification processes, optimizing them directly on raw audio waveforms. These models have achieved notable success, showcasing competitive performance in the task of fake speech detection.

While there has been considerable research on both front-end features and back-end classification networks for spoofing detection, the challenge of detecting spoofing in low-quality data encountered in real-life situations has received less attention. Models such as [37], [39] have shown relatively better performance on high-quality datasets. However, their effectiveness significantly decreases when applied to low-quality datasets, such as noisy or compressed data. For noisy data, Fan et al. [40] introduced a dual-branch knowledge distillation model for noise-robust synthetic speech detection. For compressed data, this paper introduces a knowledge distillation approach designed specifically for low-quality compressed data. This method enhances the student model's forgery detection capabilities on compressed speech by utilizing time and frequency domain distillation between the teacher and student models.

B. Knowledge Distillation

Knowledge distillation, a concept first introduced by Hinton et al. [41] in 2015, is aimed at reducing the size of a model âspecifically its depth and widthâwhile either maintaining or enhancing its performance. The fundamental principle behind knowledge distillation is the transfer of knowledge from a complex, often larger, teacher model (usually a deep neural network) to a simpler, typically smaller, student model (often a shallow neural network). This method involves retraining the student model using the predicted distributions from the teacher model as labels, aiming to closely mimic the teacher model's behavior. Moreover, knowledge distillation can improve the performance of models by leveraging the insights gained from larger models to enable smaller models to generalize more effectively to new data, thereby boosting their performance.

With the widespread application and development of knowledge distillation techniques, numerous variations have been introduced to enhance model performance across different tasks. These include:

Attention Transfer [42]: This technique leverages attention mechanisms to guide the student model in learning the teacher model's attention distribution, aiming to improve performance.

Multi-Teacher Distillation [43]: This method uses multiple teacher models to impart knowledge, thereby enhancing the student model's performance. Reference [44] introduced the Teach-DETR model, which combines predictions from multiple teacher detectors to provide parallel supervision to the student detector, improving voice deepfake detection performance.

Self-Distillation [45]: For this method, the teacher and student models are different iterations of the same architecture, engaging in knowledge transfer through self-supervised training. Reference [46] applied this method to improve detection performance in speech deepfake detection tasks.

FitNets [47]: This approach facilitates knowledge transfer at intermediate layers, enabling the student model to learn from both the final output and the intermediate layers of the teacher model. Reference [15] employed this strategy in noise-conditioned speech deepfake detection, achieving notable success.

Integral Knowledge Amalgamation [48]: Based on integral knowledge amalgamation, this method generalizes voice spoofing detection and aims to detect synthetic and replay attacks simultaneously. This method involves using two teacher models, each focusing on a type of attack, to transfer knowledge to the student model through feature fusion. An adversarial learning-based fusion module ensures the global structural consistency between both teacher models and the student model.

These developments show the dynamic nature of knowledge distillation techniques and their capacity to significantly boost the capabilities of models in various domains, including the challenging arena of spoofing detection, as demonstrated in [49], [50], [51]. Notably, the knowledge distillation framework has become a popular method across various task scenarios, showcasing remarkable performance improvements. In this work, we propose a knowledge distillation model tailored for forgery detection tasks, specifically designed for low-quality compressed scenes. Our method draws inspiration from the FitNets approach, employing intermediate layer outputs from both teacher and student models for distillation in the frequency and time domains. By doing so, the student model is enabled to more accurately replicate the teacher model's behavior in these domains. This approach significantly enhances the forgery detection capabilities of our model when dealing with low-quality compressed speech, addressing a critical need in the field of audio security.

III. Proposed Method

The prime novelty of the paper is a knowledge distillation-based solution for deepfake speech detection, which consists of two major modules:

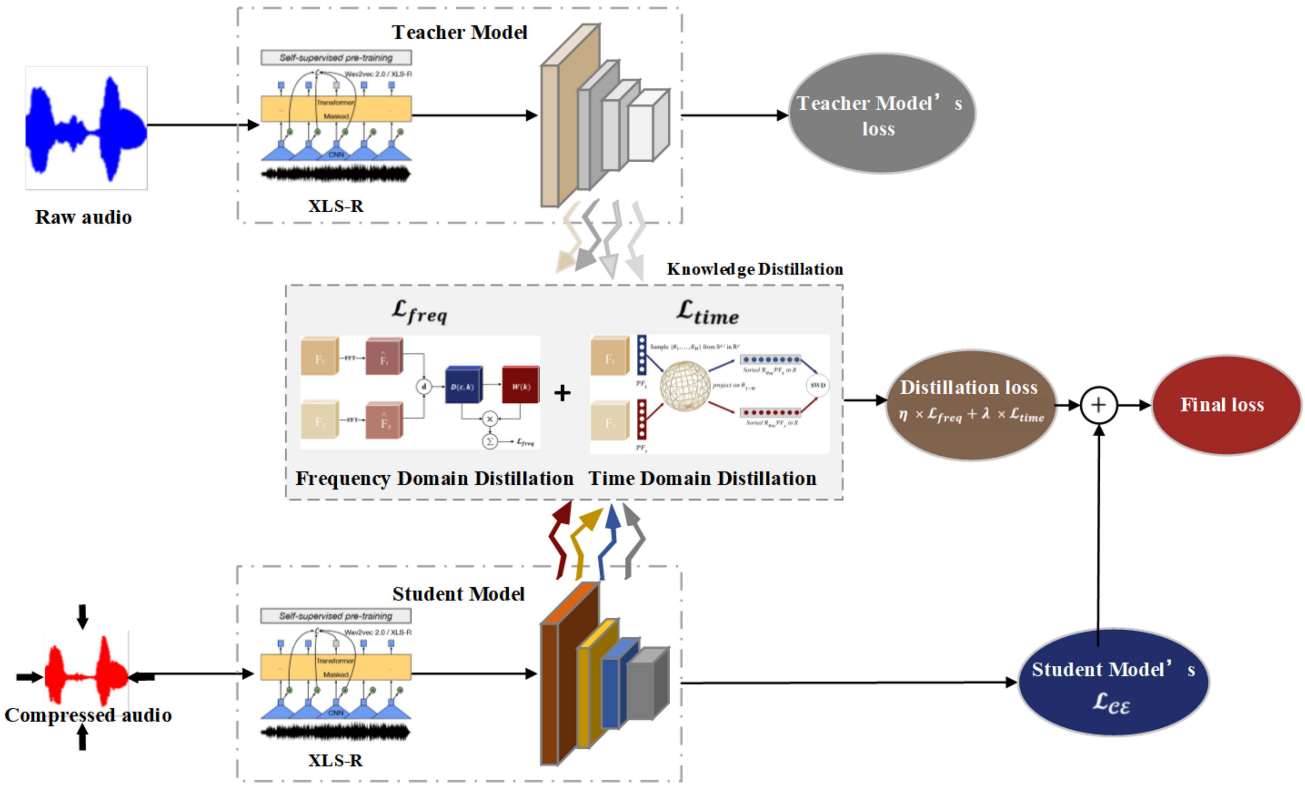


Fig. 3. The schematic diagram of the proposed Knowledge Distillation framework. The student model and teacher model both consist of XLS-R and ResNet-18. This framework incorporates both frequency and time domain distillation, aiming to enable the student model to learn high-frequency information and temporal information lost in the compressed data.

1) *frequency domain distillation*, which enables the student model to capture the high-frequency information contained in high-quality data (which is lost during compression);
2) *time domain distillation*, which allows the student model to acquire time-domain detailed information that may be lost in compressed data. Fig. 3 shows the schematic diagram of the proposed Frequency-Time Domain Knowledge Distillation Model. In this paper, we employ XLS-R [52] to extract speech features and utilize ResNet-18 [53] as the backbone network. Our teacher and student models use the same network architecture and the only difference is the input data used for training. We begin by training the teacher model with high-quality data and subsequently load the trained teacher model to perform knowledge distillation training on the student model. In what follows, we first introduce the feature extraction and backbone network, then provide details regarding the two distillation modules we propose, as well as the aggregation of the overall losses of the model.

A. Feature Extraction and Backbone Network

Self-supervised learning has been extensively applied in the literature. In the field of speech (i.e. voice data), [54] presented wav2vec 2.0, a framework designed for self-supervised speech representation learning that comprises both convolutional neural network (CNN) and Transformer. The CNN extracts a sequence of primary feature vectors from the input (voice) waveform, while the transformer maps these primary feature vectors to high-level feature representations containing global contextual information to capture the information contained in the entire sequence. Reference [52] presents XLS-R, a large-scale cross-lingually pre-trained wav2vec 2.0 model. It leverages $436K$ hours of recorded speech audio collected from 128 different languages for training, and the extensive training data admits XLS-R with great generality for various downstream tasks.

In this paper, we utilize the pre-trained large-scale model XLS-R to extract speech features from the raw speech waveform signal. It is worth noting that, we employed a fine-tuning step to empower XLS-R to better distinguish fake speeches from genuine ones. XLS-R is composed of both CNN and Transformer, both of which are network models designed for processing sequential data. Since speech is a time-series data and the speech data input to the XLS-R model doesn't undergo frequency domain transformation, we consider the speech feature output by the XLS-R model as time domain feature. The feature obtained after the Fast Fourier Transform (FFT) of the time domain feature is regarded as frequency domain feature. Subsequently, we perform knowledge distillation separately on the time and frequency domain features. This allows the student model to learn the information lost in the compressed data from the teacher model through distillation in both the time and frequency domains.

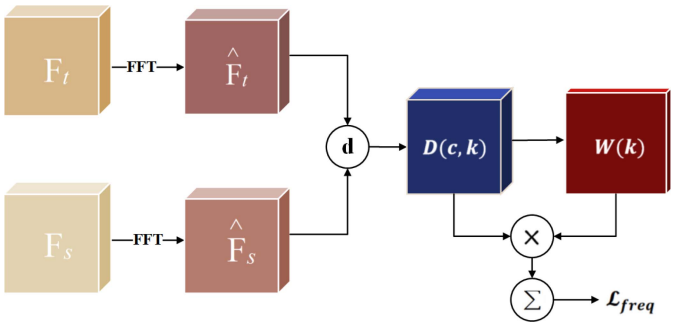


Fig. 4. Illustration of the frequency domain distiller. FFT represents the fast Fourier transform, converting features into frequency domain representations of $\hat{\mathbf{F}}_t$ and $\hat{\mathbf{F}}_s$. The distance metric function d is used to calculate the difference between the two features in the frequency domain. W represents weight, which is composed of the exponential difference across channels between the features in the frequency domain. Finally, the product of the distance and the weight is accumulated to obtain the distillation loss in the frequency domain.

In this work, we utilize the ResNet as the backbone network to implement audio forgery detection. ResNet has been widely applied in various contexts, which introduces residual connections, that is, adding input data to subsequent layers through shortcut connections, to enable learning on the residual information added from the identity mapping and alleviating the vanishing gradient problem. More specifically, we apply ResNet-18. Compared to other more complicated variants, ResNet-18 is better suited for the classification task in this paper as i) The features extracted by the XLS-R model already possess a high level of informativeness and discriminative power. Therefore, utilizing ResNet-18 allows for the full exploitation of these rich features and efficient execution of the classification task. In contrast, deeper networks may introduce overfitting issues, leading to a decrease in generalization performance. ii) ResNet-18 entails fewer parameters and computational requirements during both training and testing processes, thus offering greater computational efficiency.

B. Frequency Domain Knowledge Distillation

As mentioned previously, compression may result in high-frequency information loss in the data. As illustrated in Fig. 4, we propose to utilize frequency domain distillation to enable the student model to acquire the missing frequency domain information from the teacher model.

In what follows, we denote the number of channels as C, the total number of frames as T, and the dimension of the features as F. Firstly, the pre-trained XLS-R model extracts feature vector $\mathbf{F} \in \mathbb{R}^{T\times F}$ from the voice data. With $\mathbf{F}$ being taken as input to the classification network, from where we extract the intermediate layer features. The intermediate layer features of the teacher and student models are denoted as $\mathbf{F}_t, \mathbf{F}_s \in \mathbb{R}^{C\times T\times F}$. For capturing the frequency-domain difference between raw and compressed data, we use the canonical Fast-Fourier-Transform (FFT) to derive the frequency domain feature vectors $\hat{\mathbf{F}}_s$ and $\hat{\mathbf{F}}_t$, respectively. Here, for the teacher model, we compute the corresponding feature vector in the frequency domain

$$\hat{\mathbf{F}}_t(c,k,f) = \sum_{t=0}^{T-1} \mathbf{F}_t(c,t,f) \cdot e^{\frac{-\mathbf{i}\cdot 2\pi t k}{T}}. \tag{1}$$

Similarly, we can derive the frequency feature for the student models. Here, we use k to represent the frequency index and $\mathbf{i}$ as the imaginary unit.

To enable the student model to acquire the missing frequency domain information, we measure the disparity between the teacher and student model's frequency domain features. We first compute the square of the l_2-norm per channel c and frequency k as follows,

$$D(c,k) = \sum_{f=1}^{F} \left(\hat{\mathbf{F}}_t(c,k,f) - \hat{\mathbf{F}}_s(c,k,f)\right)^2. \tag{2}$$

where F represents the number of feature dimensions.

Since the output of FFT is a complex number, $\hat{\mathbf{F}}_t$ and $\hat{\mathbf{F}}_s$ are also complex-valued data. In (2), $\hat{\mathbf{F}}_t - \hat{\mathbf{F}}_s$ is computed by separately taking the real and imaginary parts of $\hat{\mathbf{F}}_t$ and $\hat{\mathbf{F}}_s$ and calculating their differences. The detailed process is as follows:

$$\hat{\mathbf{F}}_t - \hat{\mathbf{F}}_s = \sqrt{(\mathrm{Re}(\hat{\mathbf{F}}_t) - \mathrm{Re}(\hat{\mathbf{F}}_s))^2 + (\mathrm{Im}(\hat{\mathbf{F}}_t) - \mathrm{Im}(\hat{\mathbf{F}}_s))^2} \tag{3}$$

were Re and Im denote the real and imaginary parts of the complex-valued data, respectively.

In order to emphasize the distinctions in frequency domain features more clearly, we calculate the exponential difference across channels between the teacher and student frequency domain features. This calculation serves as the weight. This approach is specifically designed to highlight the disparities in the features, thereby reducing the impact of similarities. The weight is mathematically defined in the following manner:

$$W(k) = \exp\left(\lambda_{\text{freq}} \cdot \frac{1}{C}\sum_{c=1}^{C} D(c,k)\right) \tag{4}$$

Here, to further refine our approach, we introduce λ_{freq}, a hyperparameter, for scaling the weight. This weight is crucial. Specifically, when the distance $D(c,K)$ between the frequency domain features of the teacher and student models is large, the weight $W(K)$ is exponentially amplified. Conversely, when the distance is small, the weight $W(K)$ decreases. Therefore, this weighting mechanism can amplify the differences between the frequency-domain features, ensuring that the model training process focuses more on these differing aspects. Integrating this concept, we define the frequency domain distillation loss for our teacher and student models in the following manner:

$$\mathcal{L}_{\text{freq}} = \sum_{c=1}^{C}\sum_{k=1}^{K}\sum_{f=1}^{F}\left(W(k)\cdot\left(\hat{\mathbf{F}}_t(c,k,f) - \hat{\mathbf{F}}_s(c,k,f)\right)^2\right) \tag{5}$$

C. Time Domain Knowledge Distillation

To facilitate the effective imitation of the teacher model by the student model, particularly in learning the time domain information that is lost in compressed data, we utilize the Sliced

Wasserstein Distance (SWD) as described in [55] and contrastive loss as detailed in [56]. These methods are employed for time domain distillation on the intermediate layer features of both models.

The Sliced Wasserstein Distance is a metric for quantifying the similarity between two probability measures. It is a modification of the Wasserstein distance, tailored to lessen computational demands by projecting the data onto lower dimensions. This technique involves projecting a high-dimensional distribution onto various one-dimensional marginal distributions and then computing the optimal transportation cost for each projection.

Here we chose to use the sliced Wasserstein distance to measure time-domain feature differences for two reasons. First, it preserves structural information about the data. The sliced Wasserstein distance takes into account the distribution and shape of the data, rather than simply focusing on statistical features such as the mean and variance of the data, which is particularly important for time-series speech data. Second, the sliced Wasserstein distance is able to be robust to outliers and noise in the data when calculating the distance between distributions. This enables more accurate comparisons of similarities between different speech signals without excessive interference from noise or outliers when dealing with real speech data.

Let us denote Ω as a probability space, and μ and υ as two probability measures in $\mathcal{P}(\Omega)$. Given a norm parameter q, the q-Wasserstein distance between μ and υ in $\mathcal{P}(\Omega)$ is defined as follows:

$$W_q(\mu,\upsilon)=\left(\inf_{\gamma\in\Pi(\mu,\upsilon)}\int_{\Omega\times\Omega}c(Z_1,Z_2)^q\,d\gamma(Z_1,Z_2)\right)^{1/q} \tag{6}$$

In (6), $\gamma\in\Pi(\mu,\upsilon)$ is defined as $\Pi(\mu,\nu)=\{\gamma\in\mathcal{P}(\Omega\times\Omega)|\pi_{1\#}\gamma=\mu,\pi_{2\#}\gamma=\nu\}$, where π_1 and π_2 are two marginal projections of $\Omega\times\Omega$ to Ω. The expression $\pi_{1\#}\gamma=\mu$ signifies that the marginal projection of γ onto the first component equals μ, and similarly, $\pi_{2\#}\gamma=\upsilon$ signifies that the marginal projection of γ onto the second component equals υ. Additionally, we introduce $c:\Omega\times\Omega\to\mathbb{R}^+$ being a transportation cost function. In our experiments, we set $c(Z_1,Z_2)=|Z_1-Z_2|^2$. Here, Z_1 and Z_2 represent samples from the distributions μ and υ, respectively.

Calculating the Sliced Wasserstein distance entails selecting multiple random directional vectors on the unit sphere S^{d-1}. Subsequently, the two probability measures μ and υ are projected onto the chosen projection directions. For each distribution's projection, calculate the 1-Wasserstein distance (where q is set to 1in the q-Wasserstein distance). Finally, obtain the Sliced Wasserstein Distance (SWD) by taking the weighted average of the 1-Wasserstein distances for all projection directions. The Sliced 1-Wasserstein distance (SWD) is defined as follows:

$$\mathrm{SWD}(\mu,\upsilon)=\int_{S^{d-1}}W_1(\mathcal{R}_{\theta\mu},\mathcal{R}_{\theta\upsilon})\,d\theta \tag{7}$$

where $\mathcal{R}$ represents the Radon transform, which is a method used to project high-dimensional data onto a lower-dimensional (typically one-dimensional) space, $\mathcal{R}_{\theta\mu}$ and $\mathcal{R}_{\theta\upsilon}$ represent one-dimensional linear projection operations on probability measures μ and υ, and θ is the uniform measure on the unit sphere S^{d-1} in R^d such that $\int_{S^{d-1}}d\theta=1$. Consequently, calculating the Sliced Wasserstein distance is equivalent to solving several one-dimensional optimal transport problems with closed-form solutions.

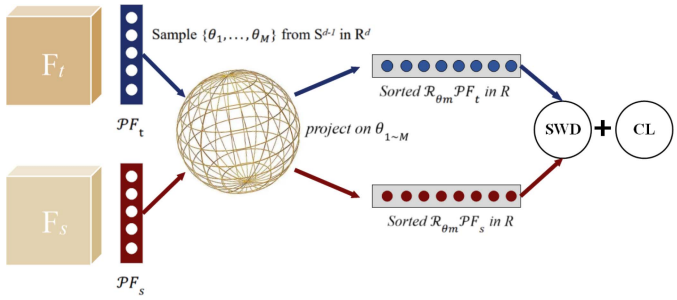


Fig. 5. Illustration of the time domain distiller. $\mathcal{P}\mathbf{F}_t$ and $\mathcal{P}\mathbf{F}_s$ represent the probability distributions obtained by normalizing the features of the teacher and student models. Then $\mathcal{P}\mathbf{F}_t$ and $\mathcal{P}\mathbf{F}_s$ are projected onto the random direction vector θ_i, and the projection results are sorted in ascending order. CL represents contrastive loss.

In particular, when sorting the $\mathcal{R}_{\theta\mu}$ and $\mathcal{R}_{\theta\upsilon}$ samples in ascending order, SWD can be approximated as:

$$\mathrm{SWD}(\mu,\upsilon)\approx\sum_{m=1}^{M}\sum_{i=1}^{N}c\left(\mathcal{R}_{\theta_m\mu_{\alpha(i)}},\mathcal{R}_{\theta_m\upsilon_{\beta(i)}}\right) \tag{8}$$

where M is the number of uniformly random samples of θ, N represents the number of samples in $\mathcal{R}_{\theta\mu}$ and $\mathcal{R}_{\theta\upsilon}$, and α and β denote permutations of the samples from $\mathcal{R}_{\theta\mu}$ and $\mathcal{R}_{\theta\upsilon}$ after they have been sorted in ascending order. In this work, to calculate the SWD for the intermediate layer features of the student and teacher models, we perform the square of the Frobenius norm normalization on $\mathbf{F}_s$ and $\mathbf{F}_t$ to obtain their probability measure representations $\mathcal{P}\mathbf{F}_s$ and $\mathcal{P}\mathbf{F}_t$. The specific definition is as follows:

$$\mathbf{F}_t\_\mathrm{norm}=\sqrt{\sum_{f=1}^{F}(\mathbf{F}_t(c,t,f)^2)}\ \text{where}\ \mathbf{F}_t\_\mathrm{norm}\in\mathbb{R}^{c\times t\times 1} \tag{9}$$

$$\mathcal{P}\mathbf{F}_t(c,t,f)=\left(\frac{\mathbf{F}_t(c,t,f)}{\mathbf{F}_t\_\mathrm{norm}(c,t,1)}\right)^2 \tag{10}$$

Similarly, we can derive the probability measure representation of the feature of the student model.

Fig. 5 pictorially illustrates our overall time domain distillation. The loss for time domain distillation is specifically defined as follows:

$$\mathcal{L}_{\text{time}}=\alpha_{\text{time}}\cdot\mathrm{SWD}\left(\mathcal{P}\mathbf{F}_t,\mathcal{P}\mathbf{F}_s\right)+\beta_{\text{time}}\cdot CL\left(\mathcal{P}\mathbf{F}_t,\mathcal{P}\mathbf{F}_s\right) \tag{11}$$

where $CL(\mathcal{P}\mathbf{F}_t,\mathcal{P}\mathbf{F}_s)$ represents the contrastive loss for $\mathcal{P}\mathbf{F}_t$ and $\mathcal{P}\mathbf{F}_s$, and α_{time} and β_{time} are hyperparameters that balance the SWD and contrastive loss.

It is important to note that, in order to improve the student model's ability to imitate the teacher model, we introduce the contrastive loss in time domain distillation to narrow the distance between samples of the same category and increase the distance between samples of different categories. $CL(\mathcal{P}\mathbf{F}_t, \mathcal{P}\mathbf{F}_s)$ is defined as follows:

$$CL\left(\mathcal{P}\mathbf{F}_t, \mathcal{P}\mathbf{F}_s\right) = \frac{1}{2N}\sum_{n=1}^{N}\Big[y_n \cdot d_n(\mathcal{P}\mathbf{F}_t, \mathcal{P}\mathbf{F}_s)^2 + (1-y_n)\cdot \max\left(0, \Delta - d_n\left(\mathcal{P}\mathbf{F}_t, \mathcal{P}\mathbf{F}_s\right)\right)^2\Big] \quad (12)$$

where N represents the number of sample pairs, each sample pair consists of a feature from student model and a feature from teacher model. When these features belong to the same category, the label y_n is set to 1, otherwise, the label is 0. The d_n denotes the Euclidean distance between $\mathcal{P}\mathbf{F}_t$ and $\mathcal{P}\mathbf{F}_s$ for the n-th sample pair, and Δ is a margin threshold used to specify the minimum distance between negative sample pairs.

D. Overall Loss Function

The total loss of our proposed knowledge distillation framework consists of three components, defined as follows:

$$\mathcal{L}_{\text{overall}} = \gamma \times \mathcal{L}_{CE} + \eta \times \mathcal{L}_{\text{freq}} + \lambda \times \mathcal{L}_{\text{time}} \quad (13)$$

where $\mathcal{L}_{\mathcal{CE}}$ represents the classification loss of the student model. The parameters γ, η, and λ are hyperparameters used to balance these three losses, respectively. Because the frequency domain distillation loss being accumulated from differences across three dimensions - channels, frame numbers, and feature dimension numbers, it tends to be much larger compared to the time domain distillation loss. Therefore, in subsequent experiments, to balance time and frequency domain distillation, we typically set η to 1 and choose a relatively larger value for λ.

IV. Experimental Setups

A. Compression Algorithms

In our experiments, the original data are compressed into the following formats.

- *MP3:* This format employs lossy compression by exploiting auditory maskingâwhere certain signals become inaudible in high (or low) frequency domains. It reduces data volume by removing less perceptually significant high and low-frequency sounds and by quantizing sound more coarsely.
- *MP2:* As a precursor to MP3, MP2 uses less advanced psychoacoustic models for lossy compression. It decreases file size by removing less perceptible high-frequency signals and utilizing masking effects, in addition to lowering the sampling rate to reduce the data stream.
- *AAC (in M4A):* The M4A file format typically contains AAC-encoded audio, which uses more efficient encoding algorithms like subband analysis and vector quantization. These algorithms partition audio signals in the frequency domain to remove less noticeable components, thus reducing data volume.
- *OGG:* This format uses the Modified Discrete Cosine Transform (MDCT) and psychoacoustic models to achieve compression. It reduces file size by discarding high-frequency signals and other elements less perceptible to the human ear.
- *GSM:* Focused on reducing bandwidth for voice transmission, the GSM codec employs Linear Predictive Coding (LPC). This technique compresses voice data by predicting future samples based on past samples and transmitting only the residual differences, effectively removing redundant high-frequency information.
- *OPUS:* OPUS codec utilizes advanced psychoacoustic models to identify and remove audio information less sensitive to human hearing. It employs lossy compression to eliminate or minimize lower-level noise, certain frequency signals, and transient signals, making it efficient for a wide range of audio types.
- *AC3:* uses subband filtering techniques to remove signals within certain frequency ranges and employs quantization and coding to reduce data volume. This often results in the loss of some high-frequency and low-frequency signals, as well as a decrease in precision.
- *DTS:* DTS utilizes psychoacoustic models and quantization techniques to remove or reduce less perceptible audio information in the data. It provides high audio quality through multi-channel encoding and high bitrates.
- *WMA:* WMA compresses audio signal frequency components, dynamic range, and sound details. It employs various encoding techniques, including audio coding, subband coding, and psychoacoustic models, to reduce the size of audio files.
- *RA:* RA is commonly used for online audio broadcasting to save bandwidth. It primarily compresses the frequency components and sound dynamic range of audio.

B. Datasets

1) ASVspoof Datasets: The Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof challenges) is designed to advance research in detecting fake audio. It primarily focuses on deceptive techniques in automated speaker verification and corresponding defensive strategies. In our work, we mainly used ASVspoof 2019 LA dataset [57] and ASVspoof 2021 DF dataset [58].

- *ASVspoof 2019 LA dataset:* The ASVspoof 2019 LA Database comprises both authentic and counterfeit speech. The latter includes converted and synthesized speech, predominantly created using 19 different spoofing algorithms (A01-A19). This database is divided into three segments: a training set for model development, a validation set for selecting the most effective model during the training process, and a testing set for assessing the model's performance. The training and validation sets predominantly speech samples synthesized by four speech synthesis algorithms and converted by two speech conversion algorithms, labeled A01 to A06. To evaluate the model's generalization capability, the test set integrates spoofing attacks created by

TABLE I
THE DETAILED INFORMATION OF ASVSPOOF 2019 LA DATASET

Subset	Number of speakers		Number of speech	
	Males	Females	Genuine speech	forged speech
training	8	12	2580	22800
development	4	6	2548	22296
evaluation	21	27	7355	63882

TABLE II
DETAILS OF THE COMPRESSION IN THE ASVSPOOF 2021 DF DATASET

Cond.	Compression	VBR(kbps)
DF-C1	-	-
DF-C2	mp3	∼80-120
DF-C3	mp3	∼220-260
DF-C4	m4a	∼20-32
DF-C5	m4a	∼96-112
DF-C6	ogg	∼80-96
DF-C7	ogg	∼256-320
DF-C8	mp3→m4a	∼80-120, ∼96-112
DF-C9	ogg→m4a	∼80-96, ∼96-112

'-' represents no compression, "TBA" stands for unknown compression technology, and "VBR" indicates the compressed bit rate.

TABLE III
SUMMARY OF ASVSPOOF 2021 LA DATA CONDITIONS

Cond.	Codec	Audio bandwidth	Transmission
LA-C1	-	16KHZ	-
LA-C2	a-law	8 kHz	VoIP
LA-C3	unk.+ μ-law	8 kHz	PSTN + VoIP
LA-C4	G.722	16 kHz	VoIP
LA-C5	μ-law	8 kHz	VoIP
LA-C6	GSM	8 kHz	VoIP
LA-C7	OPUS	16 kHz	VoIP

13 different algorithms, identified as A07 to A19. Among these, there are 11 unknown spoofing algorithms and 2 known algorithms (A16 and A19, using the same methods as A04 and A06). Comprehensive details of the ASVspoof 2019 LA dataset can be found in Table I.

- *ASVspoof 2021 LA dataset:* The ASVspoof 2021 LA evaluation data includes a collection of bona fide and spoofed utterances transmitted over a variety of telephony systems including voice-over-IP (VoIP) and a public switched telephone network (PSTN). This database, like the 21 DF dataset, contains only a test set. It includes a total of 181,566 utterances, comprising both genuine and spoofed ones. Detailed information can be found in Table III.
- *ASVspoof 2021 DF dataset:* The ASVspoof 2021 DF Database provides only the test set, designed to mirror real-life situations where the characteristics of forged speech are unpredictable. Most of the speech in the database has undergone lossy compression. Consequently, this characteristic substantially elevates the challenge of detection in the ASVspoof 2021 DF Database. Encompassing a total of 611,829 utterances, both genuine and forged, the database utilizes a variety of lossy codecs commonly employed in media storage. Detailed in Table II, these compression methods include three single lossy compression algorithms and two sequential compression algorithms. The source data for the ASVspoof 2021 DF Database is drawn from the test set of ASVspoof 2019 LA and additional datasets. Accordingly, it results in a comprehensive collection that features forged speech generated by over 100 different spoof-attack algorithms.

2) ASVspoof 2019 LA-Train-Com Dataset: In our work, we train the teacher model on the training set in the ASVspoof 2019 LA Database. Simultaneously, we apply 6 lossy compression algorithms, MP3, MP2, M4A, OGG, GSM, and Opus, to compress this Database to obtain the compressed versions for training the student model.

3) Other Datasets: To evaluate the generalization performance of our proposed model, we conducted cross-database testing on three datasets: **in_the_wild [4]**, **FOR [59]**, and **wavefake [60]**. All the data sets have been compressed using six known compression algorithms from the training set (MP3, MP2, M4A, OGG, GSM, and Opus) and four unknown compression algorithms (DTS, AC3, WMA, and RA).

- *in_the_wild:* The in_the_wild dataset comprises only the test set, consisting of 37.9 hours of found audio recordings of celebrities and politicians, of which 17.2 hours are deepfakes. Genuine speech was collected from materials in real environments such as podcasts and speeches. Forged speech was created by segmenting 219 publicly available video and audio files that were explicitly promoted as audio deepfakes.
- *FOR:* The FOR dataset contains more than 198,000 utterances from the latest deep-learning speech synthesizers as well as real speech. The dataset includes four different subsets, and we use the "for-2seconds" subset to test our proposed model. In this subset, all the speech segments are 2 seconds in duration. This dataset contains more than 198,000 utterances from the latest deep-learning speech synthesizers as well as real speech.
- *wavefake:* The wavefake dataset consists of 117,985 synthetic audio files, totaling approximately 196 hours of synthetic audio. These audio files are generated by six different speech synthesis models trained in two languages. The dataset exclusively contains generated audio and does not include any real audio. Additionally, the dataset is not divided into training, validation, and test sets. We have selected 10,000 speech samples from this dataset for testing.

C. Experimental Settings and Metrics

In our experiments, we use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ and weight decay 10^{-4}. We set the learning rate to 10^{-6} and the batch size to 10, with each model trained for 30 epochs. Additionally, each model was trained and tested in three rounds, with the final experimental results being the average of these three tests. For the feature extraction model, we selected the pre-trained XLS-R with parameter 0.3B. Our hyperparameter settings are $\lambda_{\text{freq}} = 0.1$, $\alpha_{\text{time}} =$

TABLE IV
EXPERIMENTAL RESULTS OF OUR PROPOSED METHOD AND OTHER FIVE DIFFERENT BASELINE APPROACHES ON ASVSPOOF 2021 DF DATASET

Model	DA	EER
FKD [51]	×	17.16%
Pindrop Labs' Submission [62]	✓	16.05%
STC Antispoofing Systems [63]	✓	15.64%
LLGF+XLS-R [64]	×	6.10%
ATT_ResNet18+XLS-R+Logmet [65]	×	5.34%
ATT_ResNet18+XLS-R+Logmet [65]	✓	3.72%
AASIST+XLS-R [66]	×	4.57%
AASIST+XLS-R [66]	✓	3.21%
OUR METHOD	✓	**2.82%**

The best results are highlighted in bold. DA refers to data augmentation.

$100, \beta_{\text{time}} = 50, \gamma = 1, \eta = 1, \Delta = 0.012$. Regarding λ, we determine an optimal value through experiments.

We use two metrics: the default minimum tandem detection cost function (min t-DCF) [61] and the equal error rate (EER). The min t-DCF demonstrates the impact of spoofing and spoofing detection systems on the performance of an automatic speaker verification system, while the EER reflects the purely independent spoofing detection performance. The equal error rate (EER) is the rate at which the false rejection rate and false acceptance rate are equal; the smaller the EER, the better the model's detection performance. The EER is defined as follows:

$$P_{\text{fa}}(\theta) = \frac{\#\{fake\ trials\ with\ score > \theta\}}{\#\{total\ fake\ trials\}} \tag{14}$$

$$P_{\text{miss}}(\theta) = \frac{\#\{genuine\ trials\ with\ score < \theta\}}{\#\{total\ genuine\ trials\}} \tag{15}$$

$$EER = P_{\text{fa}}(\theta) = P_{\text{miss}}(\theta) \tag{16}$$

where $P_{\text{fa}}(\theta)$ and $P_{\text{miss}}(\theta)$ denote the false alarm and miss rates at threshold θ .

V. RESULTS AND DISCUSSION

A. Comparison With Other Deepfake Detection Systems

Firstly, we evaluated our proposed frequency-time domain distillation model on the ASVspoof 2021 DF Database, and the results are shown in the last row of Table IV. Our model achieved an EER of 2.82% on the ASVspoof 2021 DF Database, which is the most competitive result we have known so far on the ASVspoof 2021 DF Database.

The FKD also employed a knowledge distillation model, but its performance on the ASVspoof 2021 DF dataset was unsatisfactory. The third and fourth rows in Table IV, representing the baseline STC Antispoofing Systems and Pindrop Labs' Submission, are models submitted by the first and second-place contestants in the ASVspoof 2021 DF track. Since the ASVspoof 2021 DF Database was proposed by the ASVspoof 2021 competition, we chose these two models for comparison. The STC Antispoofing Systems and Pindrop Labs' Submission systems used five compression algorithms (SBC, MP3, M4A, Opus, Vorbis) and two compression algorithms (MP3, M4A) respectively, to augment the training data. From the results, it can be observed that our model reduced the EER by 13.23% and 12.82% compared to these two award-winning models which also used data augmentation, demonstrating the effectiveness of our proposed method. Furthermore, we compared our model to LLGF+XLS-R, AASIST-XLS-R, and ATT_resnet18+XLS-R+Logmet, which also utilize XLS-R as the pre-trained model for speech feature extraction. The results, as shown in Table IV, indicate that these three models achieved EERs of 6.10%, 5.34%, and 4.57%, respectively. Compared to the baseline models in rows two and three of Table IV, it is evident that using XLS-R pre-trained model for speech feature extraction indeed improves the performance of spoofed speech detection. However, in comparison to the results of our model, their performance is slightly inferior. This indicates that our proposed frequency-time domain distillation model is competitive.

TABLE V
EXPERIMENTAL RESULTS OF OUR PROPOSED METHOD AND OTHER THREE DIFFERENT BASELINE APPROACHES ON ASVSPOOF 2019 LA DATASET, ASVSPOOF 2021 LA AND ASVSPOOF 2021 DF DATASET

Model \ Dataset	19 LA eval		21 LA eval		21 DF eval
	EER	min t-DCF	EER	min t-DCF	EER
TSSD [67]	2.12%	0.0623	17.75%	0.7824	32.10%
RawNet2 [29]	1.12%	0.0330	9.49%	0.4192	24.32%
AASIST [39]	0.83%	0.0275	10.51%	0.4884	19.77%
OUR METHOD	**0.30%**	**0.0098**	**2.96%**	**0.2657**	**2.82%**

The best results are highlighted in bold.

Based on the design of the ASVspoof 2021 DF dataset, our chosen comparative methods are exclusively trained on the ASVspoof 2019 LA dataset (high-quality dataset) and tested on the ASVspoof 2021 DF dataset (low-quality compression dataset). To further substantiate the effectiveness of our approach, we trained the ATT ResNet18+XLS-R+Logmet model and the AASIST+XLS-R model using both the high-quality dataset and the compressed dataset (ASVspoof 2019 LA-train-com dataset and ASVspoof 2019 LA-train dataset). As shown in Table IV, the EERs were 3.72% and 3.21% for the sixth and eighth rows respectively. Both models achieved improved detection performance compared to models trained without data augmentation, indicating that incorporating compressed data enhances model robustness. Compared to the above experiments, our proposed method achieves the lowest EER, further indicating that the superior performance of our model is not only due to data augmentation but also due to our overall model design, such as the distillation approach.

Furthermore, we evaluated our model on the ASVspoof 2019 LA database, and the results, as shown in the last row of Table V, indicate an EER of 0.30% and a min t-DCF of 0.0098. This demonstrates that our model not only achieved state-of-the-art (SOTA) results on the ASVspoof 2021 DF dataset but also came very close to the SOTA on the ASVspoof 2019 LA Database. Additionally, we selected three models, RawNet, AASIST, and Dual-Branch Network, which have shown relatively

TABLE VI
THE EXPERIMENTAL RESULTS OF OUR PROPOSED MODEL ON FOUR DIFFERENT DATASETS

Model / Dataset	ResNet18+XLS-R			ResNet18+XLS-R+KD$_{\mathcal{FD}+\mathcal{TD}}$		
	All	Known com-algorithm	Unknown com-algorithm	All	Known com-algorithm	Unknown com-algorithm
19_LA_eval-low	4.94%	2.63%	5.84%	1.33%	1.15%	1.43%
In_the_wild-low	16.93%	13.96%	19.60%	14.97%	13.01%	17.10%
FoR_2s_eval-low	23.14%	21.65%	27.02%	21.48%	20.00%	24.86%
WAVfake-low	ACC:84.07%	ACC:81.51%	ACC:87.33%	ACC:94.52%	ACC:94.11%	ACC:95.03%

In the first three datasets, the evaluation metric used is EER, while on the WAVfake dataset, the evaluation metric is ACC. 'All' denotes the entire dataset, 'known com-alg rithm' indicates data compressed with only the six known compression algorithms, and 'unknown com-algorithm' pertains to data compressed with the four unknown compression algorithms.

good performance on the 19_LA dataset over the past two years, with EERs of 1.12%, 0.83%, and 0.80%, respectively. We evaluated these three models on the ASVspool 2021 DF dataset, and the results are shown in Table V, indicate a sharp decline in their detection performance, with EERs of 24.32%, 19.77%, and 30.28%. This highlights the challenge in the current field of spoofed speech detection where models designed for high-quality data often struggle to effectively detect low-quality data. However, our frequency-time domain model effectively addresses this issue and achieves near SOTA performance on both high-quality and low-quality data.

We also evaluated our model on the ASVspoof 2021 LA dataset and the results are shown in Table V. From the experimental results, we can see that our proposed method also has a positive effect on the 21 LA dataset (both the data in the telephony scenario). The reasons for this we analyze are as follows: First, the two compression algorithms used in our training data are the same as the two coding and decoding methods in the 21 LA dataset, so it improves the detection performance on the 21 LA dataset. Second, the spoofing algorithms in the 21 LA dataset are consistent with those in the 19 LA eval dataset. Since our model was trained on the 19 LA training set and showed excellent detection performance on the 19 LA eval set, it is also effective in detecting the 21 LA dataset.

B. Cross-Database Experiments

To assess the generalization of our model, we conducted tests on four different datasets, and the experimental results are shown in Table VI. As there is currently only one compressed dataset, ASVspoof 2021 DF, and our model's primary aim is to improve the detection performance of compressed data, we compressed the data of ASVspoof 2019 LA eval, In_the_wild, for-2seconds and WAVfake to get four new compressed datasets. Furthermore, to assess the robustness of our model, when compressing these four datasets, in addition to using the six known compression algorithms from the training dataset, we also introduced four unknown compression algorithms.

In Table VI, columns two to four represent the detection results without distillation modules, while columns five to seven show the detection results with distillation modules. The second and fifth column displays the results on the entire test dataset, the third and sixth column represents the results on data compressed using the six known compression algorithms, and the fourth and seventh column displays the results on data compressed using the four unknown compression algorithms. It's worth noting that in the WAVfake dataset, there are only synthetic voices and no real voices. Therefore, EER cannot be used as the evaluation metric for this dataset. In this case, accuracy (ACC) is used as the evaluation metric.

From Table VI, it is evident that our model's performance improves to some extent after adding distillation modules on all four different datasets. Additionally, when processing data compressed using unknown algorithms, the detection performance of our model decreases, which is due to the fact that the model has not seen these unknown compression algorithms during the training phase, and is also related to the model's adaptability to specific compression algorithms. However, the degree of performance degradation on data compressed by unknown algorithms is not significant, revealing that our model possesses a certain degree of generalization capability when facing unseen compression methods. Nevertheless, this generalization is still insufficient, and we will explore strategies for further improving it in future research.

It is noteworthy that on the WAVfake-low dataset, the detection performance of data from known compression algorithms is lower than that of data from unknown compression algorithms. Upon careful analysis, we have found that this is due to the poor detection performance of data compressed with the MP2 algorithm, which has lowered the overall accuracy of the known compression algorithm dataset. The low detection performance of MP2-compressed data in the WAVfake-low dataset may be attributed to several factors. Firstly, the WAVfake dataset consists entirely of forged speech, containing more forged artifacts. Additionally, MP2 is an older audio compression format, which is less efficient in terms of compression and audio quality.

C. Ablation Experiments

We studied the quantitative impact of each module of our proposed model on the final result on ASVspoof 2019 LA Database (high quality data) and ASVspoof 2021 DF Database (low quality compressed data). The results are shown in Table VII. We observe that, with the utilization of the feature extraction model XLS-R, the EER on the 19_LA dataset decreases from 7.66% to 0.37%, and on the 21_DF dataset, the EER decreases from 27.30% to 4.86%, as compared to using lfcc features. This indicates that XLS-R, trained on large-scale speech data, exhibits strong robustness across various types of speech. Compared to handcrafted features like LFCC, the combination of XLS-R with

TABLE VII
THE EER RESULTS OF THE ABLATION EXPERIMENTS FOR EACH MODULE OF OUR PROPOSED MODEL ON THE ASVSPOOF 2019 LA DATASET AND ASVSPOOF 2021 DF DATASET

Model \ Dataset	19_LA eval	21_DF eval
ResNet18+LFCC*	7.66%	27.30%
ResNet18+LFCC+KD$_{\mathcal{FD}+\mathcal{TD}}$	6.74%	25.54%
ResNet18+XLS-R*	0.37%	4.86%
ResNet18+XLS-R#	0.60%	3.66%
ResNet18+XLS-R*+#	0.27%	3.79%
ResNet18+XLS-R+KD$_{\mathcal{TD}}$	0.31%	3.18%
ResNet18+XLS-R+KD$_{\mathcal{FD}}$	0.35%	3.11%
ResNet18+XLS-R+KD$_{\mathcal{FD}+\mathcal{TD}}$	**0.30%**	**2.82%**

Denotes the model trained on the asvspoof 2019 la-train dataset, # signifies the model trained on the asvspoof 2019 LA-train-com dataset, *+ # indicates the model concurrently trained on both the asvspoof 2019 LA-train and asvspoof 2019 LA-train-com datasets.

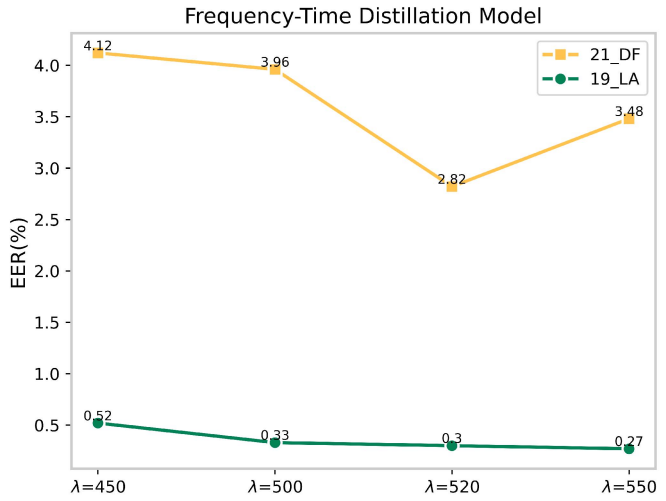


Fig. 6. The EER results for the frequency-time domain distillation model with different values of the time-domain distillation loss's hyper-parameter. The yellow line represents the results on the ASVspoof 2021 DF dataset, while the green line represents the results on the ASVspoof 2019 LA dataset.

the backend classification network enables end-to-end learning by directly extracting feature representations from raw speech data, thereby enhancing its adaptability for the spoofing detection task. Additionally, to demonstrate that the performance improvement of our model is not solely dependent on XLS-R, we applied time-domain distillation and frequency-domain distillation modules to Resnet18+LFCC model. As shown in the third row of Table VII, the results indicate that time-frequency domain distillation also enhances the spoofing detection performance of the Resnet18+LFCC model, further corroborating the effectiveness of this distillation method. Moreover, in order to validate the effect of dataset training strategy on model performance, we trained the Resnet18+XLS-R model using three different dataset training strategies. The results, as shown in rows four, five, and six of Table VII, indicate that the EER on the 21_DF dataset decreased from 4.86% to 3.66% when the model was trained on the compressed dataset compared to the high-quality dataset. However, the EER on the 19_lA dataset increased from 0.37% to 0.60%. This suggests that the model has learned the feature distribution of the compressed dataset, making it better suited to fit the 21_DF dataset. However, due to the lack of some crucial information in the compressed data, the detection performance on the 19_lA dataset experiences a decline. Meanwhile, the model trained on both the high-quality and compressed datasets shows lower EERs on the 19_lA and 21_DF datasets compared to the model trained solely on the high-quality dataset. This demonstrates that combining high-quality and compressed data enhances the model's generalization capability.

Furthermore, the inclusion of the frequency-domain distillation or time-domain distillation modules further improved the performance of the ResNet18+XLS-R model on both the 19_LA and 21_DF datasets. It is clear that our proposed frequency-domain distillation and time-domain distillation modules can individually contribute to the spoofing detection model, improving its performance in both high and low quality data scenarios. By simultaneously integrating both distillation modules, the model's performance reaches its optimal state. On the 19_LA dataset, the EER is merely 0.30%, and on the 21_DF dataset, it is 2.82%. The results of our ablation experiments demonstrate that each module added to our model has a beneficial effect on the overall forgery detection capability. In particular, the frequency domain distillation and time domain distillation modules contribute to the student model's learning from the teacher model in distinct ways, addressing the issue of information loss during data compression. The integration of these two modules yields the most optimal results.

It is worth noting here that why does the student model's detection performance on high-quality data improve after adding the distillation module? We believe that by using low-quality compressed data for distillation training, the student model not only learns some features of compressed data, but also learns features of high-quality data from the teacher model. This increases the diversity of data features learned by the student model, aiding it in better distinguishing between genuine and spoofed data, regardless of whether it's high-quality or low-quality data.

D. Further Analysis

Analysis of time-domain distillation weight λ: We also conducted experiments on the hyper-parameter λ as mentioned earlier, which controls time-domain distillation. Since the loss from frequency-domain distillation is significantly larger than that from time-domain distillation, to balance these two distillation modules, we tried four different values of λ around the $\mathcal{L}_{\mathcal{FD}}/\mathcal{L}_{\mathcal{TD}}$ range. We tested their impact on the overall model performance on the 19_LA and 21_DF datasets, and the results are shown in the Fig 6. It can be observed that the model performs best on the 21_DF dataset when λ is set to 520.

Applicability to different backbones: As shown in Table VIII, we evaluated the general applicability of our time-frequency domain knowledge distillation method using two backbone networks different from ResNet18. To enhance model detection performance, we still employed XLS-R as the feature extractor.

TABLE VIII
PERFORMANCE OF THE PROPOSED FTDKD WITH DIFFERENT BACKBONES

Mode	21_DF eval EER
SpecRNet [68]+XLS-R*	7.35%
SpecRNet [68]+XLS-R*+#	6.89%
SpecRNet [68]+XLS-R+KD$_{\mathcal{FD}+\mathcal{TD}}$	**4.27%**
AASIST+XLS-R*	4.57%
AASIST+XLS-R*+#	3.21%
AASIST+XLS-R+KD$_{\mathcal{FD}+\mathcal{TD}}$	**3.08%**

*denotes the model trained on the asvspoof 2019 LA-train dataset, *+# indicates the model concurrently trained on both the asvspoof 2019 LA-train and asvspoof 2019 LA-train-com datasets.

We trained these two networks on both the original 19_LA training set and the original 19_LA training set augmented with compressed data. Subsequently, we applied the time-frequency domain knowledge distillation to both networks. The results in Table VIII demonstrate that, for different backbone networks, our distillation method enhances their forgery detection performance on low-quality compressed data compared to simple data augmentation methods. This indicates that our method is highly applicable to various backbone networks.

VI. CONCLUSION

In this paper, we propose a deepfake audio detection model based on frequency-time domain knowledge distillation for low-quality compressed deepfake audio. Low-quality compressed audio typically loses high-frequency information and time domain details during compression. Therefore, we propose a knowledge distillation approach to enhance the detection performance of low-quality compressed audio. Specifically, we employ the pre-trained self-supervised audio feature extractor, XLS-R, to extract high-quality and generalizable audio features. We employ data distillation, training the teacher model with high-quality data and the student model with low-quality compressed data. This allows the student model to learn high-frequency information and time domain details through frequency and time domain distillation from the teacher model. The experimental results on the ASVspoof 2021 DF dataset demonstrate the high effectiveness of our proposed method for low-quality compressed data, achieving an EER of 2.82%, which outperforms all individual systems. Furthermore, our approach exhibits outstanding performance on the ASVspoof 2019 LA dataset, with an EER of 0.30%, showcasing the versatility of our model for both high-quality and low-quality data. In the future, we will continue to explore deepfake detection methods for different types of data in real-world environments.

REFERENCES

[1] C. Wang et al., "Neural codec language models are zero-shot text to speech synthesizers," 2023, *arXiv:2301.02111*.

[2] Z. Khanjani, G. Watson, and V. P. Janeja, "Audio deepfakes: A survey," *Front. Big Data*, vol. 5, 2023, Art. no. 1001063.

[3] Z. Lv, S. Zhang, K. Tang, and P. Hu, "Fake audio detection based on unsupervised pretraining models," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 9231–9235.

[4] N. M. Müller, P. Czempin, F. Dieckmann, A. Froghyar, and K. Böttinger, "Does audio deepfake detection generalize?," 2022, *arXiv:2203.16263*.

[5] Z. Wu et al., "ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge," in *Proc. 16TH Ann. Conf. Int. Speech Commun. Assoc.*, 2015, vol. 1-5, pp. 2037–2041.

[6] Z. Wu et al., "ASVspoof: The automatic speaker verification spoofing and countermeasures challenge," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 4, pp. 588–604, Jun. 2017.

[7] A. Nautsch et al., "ASVspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech," *IEEE Trans. Biom. Behav. Ident. Sci.*, vol. 3, no. 2, pp. 252–265, Apr. 2021.

[8] J. Yamagishi et al., "ASVspoof 2021: Accelerating progress in spoofed and deepfake speech detection," 2021, *arXiv:2109.00537*.

[9] J. Yi et al., "Add 2022: The first audio deep synthesis detection challenge," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 9216–9220.

[10] J. Yi et al., "Add 2023: The second audio deepfake detection challenge," 2023, *arXiv:2305.13774*.

[11] H. Wu et al., "Partially fake audio detection by self-attention-based fake span discovery," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 9236–9240.

[12] L. Wang, B. Yeoh, and J. W. Ng, "Synthetic voice detection and audio splicing detection using se-res2net-conformer architecture," in *Proc. 13th Int. Symp. Chin. Spoken Lang. Process.*, 2022, pp. 115–119.

[13] Z. Zeng and Z. Wu, "Audio splicing localization: Can we accurately locate the splicing tampering?," in *Proc. 13th Int. Symp. Chin. Spoken Lang. Process.*, 2022, pp. 120–124.

[14] J. Yang and R. K. Das, "Long-term high frequency features for synthetic speech detection," *Digit. Signal Process.*, vol. 97, 2020, Art. no. 102622 .

[15] P. Liu, Z. Zhang, and Y. Yang, "End-to-end spoofing speech detection and knowledge distillation under noisy conditions," in *Proc. Int. Joint Conf. Neural Netw.*, 2021, pp. 1–7.

[16] S. Woo et al., "Add: Frequency attention and multi-view based knowledge distillation to detect low-quality compressed deepfake images," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, pp. 122–130.

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

[18] Q. Fu, Z. Teng, J. White, M. E. Powell, and D. C. Schmidt, "Fastaudio: A learnable audio front-end for spoof speech detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 3693–3697.

[19] S. Novoselov, A. Kozlov, G. Lavrentyeva, K. Simonchik, and V. Shchemelinin, "STC anti-spoofing systems for the ASVspoof 2015 challenge," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2016, pp. 5475–5479.

[20] J. Li, H. Wang, P. He, S. M. Abdullahi, and B. Li, "Long-term variable Q transform: A novel time-frequency transform algorithm for synthetic speech detection," *Digit. Signal Process.*, vol. 120, 2022, Art. no. 103256.

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

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

[23] C. Wang, J. Yi, J. Tao, C. Y. Zhang, S. Zhang, and X. Chen, "Detection of cross-dataset fake audio based on prosodic and pronunciation features," in *Proc. Interspeech*, 2023.

[24] Z. Mostaani and M. M. Doss, "On breathing pattern information in synthetic speech," in *Proc. Interspeech*, 2022, pp. 2768–2772.

[25] Y. Xie, Z. Zhang, and Y. Yang, "Siamese network with wav2vec feature for spoofing speech detection," in *Proc. Interspeech*, 2021, pp. 4269–4273.

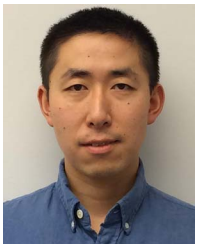
[26] J. M. Martín-Doñas and A. Álvarez, "The vicomtech audio deepfake detection system based on wav2vec2 for the 2022 add challenge," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 9241–9245.

[27] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[28] M. Todisco, H. Delgado, and N. W. Evans, "A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients," Odyssey, vol. 2016, pp. 283–290, 2016.
[29] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with rawnet2," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* 2021, pp. 6369–6373.
[30] E. Conti et al., "Deepfake speech detection through emotion recognition: A semantic approach," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 8962–8966.
[31] T.-P. Doan, L. Nguyen-Vu, S. Jung, and K. Hong, "BTS-E: Audio deepfake detection using breathing-talking-silence encoder," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.
[32] H. Ling, L. Huang, J. Huang, B. Zhang, and P. Li, "Attention-based convolutional neural network for ASV spoofing detection," in *Proc. Interspeech*, 2021, pp. 4289–4293.
[33] Y. Zhang12, W. Wang12, and P. Zhang12, "The effect of silence and dual-band fusion in anti-spoofing system," in *Proc. Interspeech*, 2021, pp. 4279–4283.
[34] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.
[35] X. Li, X. Wu, H. Lu, X. Liu, and H. Meng, "Channel-wise gated res2net: Towards robust detection of synthetic speech attacks," 2021, *arXiv:2107.08803*.
[36] X. Li et al., "Replay and synthetic speech detection with res2net architecture," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 6354–6358.
[37] K. Ma, Y. Feng, B. Chen, and G. Zhao, "End-to-end dual-branch network towards synthetic speech detection," *IEEE Signal Process. Lett.*, vol. 30, pp. 359–363, 2023.
[38] C. Wang et al., "Fully automated end-to-end fake audio detection," in *Proc. 1st Int. Workshop Deepfake Detection Audio Multimedia*, 2022, pp. 27–33.
[39] J.-W. Jung et al., "AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 6367–6371.
[40] C. Fan et al., "Dual-branch knowledge distillation for noise-robust synthetic speech detection," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 32, pp. 2453–2466, 2024.
[41] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[42] N. Komodakis and S. Zagoruyko, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proc. Int. Conf. Learn. Representations*, 2017.
[43] S. You, C. Xu, C. Xu, and D. Tao, "Learning from multiple teacher networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 1285–1294.
[44] L. Huang et al., "Teach-DETR: Better training DETR with teachers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 15759–15771, Dec. 2023.
[45] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3713–3722.
[46] J. Xue et al., "Learning from yourself: A self-distillation method for fake speech detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.
[47] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*.
[48] Y. Ren, H. Peng, L. Li, X. Xue, Y. Lan, and Y. Yang, "Generalized voice spoofing detection via integral knowledge amalgamation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 2461–2475, 2023.
[49] J. Lu, Y. Zhang, W. Wang, Z. Shang, and P. Zhang, "One-class knowledge distillation for spoofing speech detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2024, pp. 11251–11255.
[50] J. Xue, C. Fan, J. Yi, J. Zhou, and Z. Lv, "Dynamic ensemble teacher-student distillation framework for light-weight fake audio detection," *IEEE Signal Process. Lett.*, vol. 31, pp. 2305–2309, 2024.
[51] C. Fan, S. Dong, J. Xue, Y. Chen, J. Yi, and Z. Lv, "Frequency-mix knowledge distillation for fake speech detection," 2024, *arXiv:2406.09664*.
[52] A. Babu et al., "XLS-R: Self-supervised cross-lingual speech representation learning at scale," 2021, *arXiv:2111.09296*.
[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[54] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 12449–12460, 2020.
[55] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10285–10295.
[56] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 1735–1742.
[57] X. Wang et al., "ASVSpoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Comput. Speech Lang.*, vol. 64, 2020, Art. no. 101114.
[58] X. Liu et al., "ASVspoof 2021: Towards spoofed and deepfake speech detection in the wild," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 2507–2522, 2023.
[59] R. Reimao and V. Tzerpos, "FoR: A dataset for synthetic speech detection," in *Proc. Int. Conf. Speech Technol. Hum.- Comput. Dialogue*, 2019, pp. 1–10.
[60] J. Frank and L. Schönherr, "Wavefake: A data set to facilitate audio deepfake detection," 2021, *arXiv:2111.02813*.
[61] T. Kinnunen et al., "t-DCF: A detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification," in *Proc. Speaker Odyssey Speaker Lang. Recognit. Workshop*, 2018.
[62] T. Chen, E. Khoury, K. Phatak, and G. Sivaraman, "Pindrop labs' submission to the ASVspoof 2021 challenge," in *Proc. Ed. Autom. Speaker Verification Spoofing Countermeasures Challenge*, 2021, pp. 89–93.
[63] A. Tomilov, A. Svishchev, M. Volkova, A. Chirkovskiy, A. Kondratev, and G. Lavrentyeva, "STC antispoofing systems for the asvspoof2021 challenge," in *Proc. ASVspoof Workshop*, 2021, pp. 61–67.
[64] X. Wang and J. Yamagishi, "Investigating self-supervised front ends for speech spoofing countermeasures," in *Proc. Speaker Lang. Recognit. Workshop*, 2021.
[65] B. Wang, Y. Ma, Y. Tang, R. Wang, and M. Zhang, "Enhancing synthesized speech detection with dual attention using features fusion," in *Proc. Int. Conf. Comput. Appl. Technol.*, 2023, pp. 104–109.
[66] H. Tak, M. Todisco, X. Wang, J.-W. Jung, J. Yamagishi, and N. Evans, "Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation," 2022, *arXiv:2202.12233*.
[67] G. Hua, A. B. J. Teoh, and H. Zhang, "Towards end-to-end synthetic speech detection," *IEEE Signal Process. Lett.*, vol. 28, pp. 1265–1269, 2021.
[68] P. Kawa, M. Plata, and P. Syga, "SpecrNet: Towards faster and more accessible audio deepfake detection," in *Proc. IEEE Int. Conf. Trust Secur. Privacy Comput. Commun.*, 2022, pp. 792–799.

Bo Wang (Member, IEEE) received the B.S. degree in electronic and information engineering and the M.S. and Ph.D. degrees in signal and information processing from the Dalian University of Technology, Dalian, China, in 2003, 2005, and 2010, respectively. From 2010 to 2012, he was a Postdoctoral Research Associate with the Faculty of Management and Economics, Dalian University of Technology. He is currently a Professor with the School of Information and Communication Engineering, Dalian University of Technology. His current research interests include multimedia processing and security and artificial intelligence security.

Yeling Tang received the B.S. degree in software engineering from China West Normal University, Nanchong, China, in 2022. She is currently working toward the M.S. degree with the School of Information and Communication Engineering, Dalian University of Technology, Dalian, China. Her main research interests include audio deepfake detection and deep learning.



Fei Wei (Member, IEEE) received the Ph.D. degree in electrical engineering from the State University of New York at Buffalo, Buffalo, NY, USA, in 2020. He is currently a Research Scientist with Alibaba Group, China. He was a Research Fellow with the National University of Singapore, Singapore, and as a Postdoctoral Research Scholar with Arizona State University, Tempe, AZ, USA. His research interests include machine learning, privacy and security, and network information theory.

Zhongjie Ba (Member, IEEE) received the Ph.D. degree in computer science and engineering from the State University of New York at Buffalo, Buffalo, NY, USA, in 2019. He is currently a Professor with the School of Cyber Science and Technology, College of Computer Science and Technology, Zhejiang University, Hangzhou, China. He was a Postdoctoral Researcher with the School of Computer Science, McGill University, Montreal, QC, Canada. Results have been authored or coauthored in peer-reviewed top conferences and journals, including S&P, CCS, NDSS, INFOCOM, and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. His current research interests include deepfake detection, forensic analysis of multimedia content, and privacy-enhancing technologies in the context of Internet of Things. He is an Associate Editor of IEEE INTERNET OF THINGS JOURNAL and the Technical Program Committee of several conferences in the field of Internet of Things and wireless communication.

Kui Ren (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from Worcester Polytechnic Institute, Worcester, MA, USA. He is currently a Professor and the Dean of College of Computer Science and Technology, Zhejiang University, Hangzhou, China, where he also directs the Institute of Cyber Science and Technology. Before that, he was SUNY Empire Innovation Professor with the State University of New York at Buffalo, Buffalo, NY, USA. He has authored or coauthored extensively in peer-reviewed journals and conferences. His current research interests include data security, IoT security, AI security, and privacy. He was the recipient of many recognitions including Guohua Distinguished Scholar Award of ZJU, IEEE CISTC Technical Recognition Award, SUNY Chancellorâs Research Excellence Award, Sigma Xi Research Excellence Award, NSF CAREER Award, and received the Test-of-time Paper Award from IEEE INFOCOM and many Best Paper Awards from IEEE and ACM, including ACM MobiSys, IEEE ICDCS, IEEE ICNP, IEEE Globecom, ACM/IEEE IWQoS. His h-index is 93, with a total citation exceeding 47,000 according to Google Scholar. Kui is a Fellow of ACM. He is a frequent reviewer for funding agencies internationally and serves on the editorial boards of many IEEE and ACM journals. Among others, he is the Chair of SIGSAC of ACM China Council, a member of ACM ASIACCS steering committee, and a member of S&T Committee of Ministry of Education of China.