Rethinking Fake Speech Detection: A Generalized Framework Leveraging Spectrogram Magnitude

Zihao Liu
Iowa State University
zihaoliu@iastate.edu

Aobo Chen
Iowa State University
aobochen@iastate.edu

Yan Zhang
Iowa State University
yanzh@iastate.edu

Wensheng Zhang
Iowa State University
wzhang@iastate.edu

Chenglin Miao
Iowa State University
cmiao@iastate.edu

***Abstract*—Speech synthesis technologies, driven by advances in deep learning, have achieved remarkable realism, enabling diverse applications across various domains. However, these technologies can also be exploited to generate fake speech, introducing significant risks. While existing fake speech detection methods have shown effectiveness in controlled settings, they often struggle to generalize to unseen scenarios, including new synthesis models, languages, and recording conditions. Moreover, many existing approaches rely on specific assumptions and lack comprehensive insights into the common artifacts inherent in fake speech. In this paper, we rethink the task of fake speech detection by proposing a new perspective focused on analyzing the spectrogram magnitude. Through extensive analysis, we uncover that synthetic speech consistently exhibits artifacts in the magnitude representation of the spectrogram, such as reduced texture detail and inconsistencies across magnitude ranges. Leveraging these insights, we introduce a novel assumption-free and generalized fake speech detection framework. The framework partitions spectrograms into layered representations based on magnitude and detects artifacts across both spatial and discrete cosine transform (DCT) domains using 2D and 3D representations. This design enables the framework to effectively capture fine-grained artifacts and synthesis inconsistencies inherent in fake speech. Extensive experiments demonstrate that the proposed framework achieves state-of-the-art performance on several widely used public audio deepfake datasets. Furthermore, evaluations in real-world scenarios involving black-box Web voice-cloning APIs highlight the framework's robustness and practical applicability, consistently outperforming baseline methods.**

I. Introduction

Speech synthesis has witnessed significant advancements in recent years, driven by breakthroughs in deep learning and neural network architectures. Modern speech synthesis technologies can generate highly realistic audio that mimics the voice of a target speaker with minimal data requirements. These systems leverage advanced techniques such as neural vocoders [1], attention mechanisms [2], and generative adversarial networks (GANs) [3] to produce synthetic speech with natural intonation, rhythm, and prosody. Applications of speech synthesis span a wide range of domains, including audiobooks, personalized learning, and entertainment.

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.231024
www.ndss-symposium.org

However, the rapid evolution of speech synthesis technology has also introduced significant risks. Attackers can exploit these tools to create fake speech or deepfake audio, posing threats in the form of misinformation, identity theft, and fraud. For instance, in August 2019, criminals used AI-based software to impersonate the voice of a U.K. energy firm's CEO, successfully defrauding over $243,000 [4]. Similarly, in October 2021, fraudsters cloned a company director's voice to steal $35 million from a bank [5]. Such incidents demonstrate the growing threat posed by fake speech, which can undermine trust in voice authentication systems, compromise security, and cause significant financial losses.

To address the concerns caused by fake speech, researchers have developed a variety of detection methods. Most existing approaches focus on optimizing combinations of acoustic features and neural architectures [6], [7]. Additionally, some methods rely on specific assumptions [8], [9], such as the presence of liveness cues (e.g., heartbeat, breathing patterns, or microphone artifacts), or require auxiliary meta-labels during training [10], [11], such as the category of synthesis systems or vocoders, to trace the unique fingerprint of specific synthesis tools. While many of these methods achieve satisfactory detection accuracy in intra-dataset settings, they often generalize poorly to unseen speech synthesis systems, languages, or recording environments. In such scenarios, their performance can degrade to random guessing or even below random guessing [12]. Although these limitations have been recognized, and some recent studies have attempted to address detection robustness through multi-task learning [13], [14], speech augmentation [15], [14], or improved sample balancing [15], many of these approaches still depend on vocoder-specific information. This reliance on distinguishing different types of fake speech in feature space limits their effectiveness in truly unseen settings. Furthermore, none of these studies systematically investigate the common, explainable, and characterizable artifacts of fake speech or provide clear explanations for why, how, and where synthetic speech deviates from natural speech. This lack of a deeper understanding of synthesis artifacts not only undermines trust in existing detection methods but also restricts the potential to refine models for better robustness.

The threat posed by fake speech, coupled with the limitations of existing detection methods, raises an urgent question: *Given the diversity and complexity of fake speech data, is there a new perspective that can guide the development of a*

more effective detection method? An ideal solution should be assumption-free and grounded in insights derived from common and explainable artifacts of fake speech. It should not only achieve high detection accuracy across diverse settings but also demonstrate robustness to unseen synthesis techniques, speakers, languages, and recording environments.

In this paper, we rethink the task of fake speech detection and propose a new perspective for developing a more effective and robust detection approach. Specifically, we analyze and address three fundamental questions essential to advancing fake speech detection: (1) How are artifacts in fake speech generated? (2) How do these artifacts manifest in fake speech? and (3) Do these artifacts share generalizable characteristics? These questions are closely tied to the generative origins of artifacts, a dimension that has not been systematically discussed or explored in prior work. Through comprehensive analysis and case study experiments, we find that while fake speech can exhibit various types of artifacts, it consistently displays more pronounced issues when analyzed through *the magnitude representation of the spectrogram*, a perspective that has been largely overlooked in existing studies. A spectrogram represents the time-frequency content of an audio signal, where the *magnitude* encodes the intensity or energy at specific time-frequency points. Our investigation reveals that synthetic speech often lacks texture details and energy, exhibits reduced variance in energy distribution, and sometimes shows repetitive patterns in smaller magnitude ranges. Furthermore, synthetic speech frequently demonstrates inconsistencies in synthesis quality across different magnitude ranges. The artifacts observed in the magnitude representation of the spectrogram are common and can be effectively identified in both the spatial domain (time-frequency representation) and the discrete cosine transform (DCT-frequency) domain.

Building on the above insights, we propose a novel assumption-free and generalized fake speech detection framework. The core idea of the proposed framework is to partition the spectrogram into layered representations based on magnitude and detect artifacts in both the spatial and DCT domains. Specifically, the spectrogram is divided into a sequence of 2D sub-spectrograms, each corresponding to a specific magnitude range, and a 2D DCT spectrum is generated for each sub-spectrogram. To capture dynamic variations across magnitude ranges, these sequences are further organized into video-like 3D representations, where each "video frame" corresponds to a sub-spectrogram or its DCT spectrum. The framework employs ResNet18 [16] for analyzing 2D inputs and TimeSformer [17] for processing 3D representations, with their outputs combined through a multilayer perceptron (MLP) to produce the final prediction. This multi-branch approach enables precise detection of fine-grained artifacts, comprehensive analysis of magnitude-based patterns, and robust identification of synthesis inconsistencies.

The performance of the proposed framework is evaluated on several widely used public audio deepfake datasets. The results demonstrate that it maintains excellent performance in intra-dataset evaluations while exhibiting strong generalizability to unseen speech synthesis models, speakers, and languages. Additionally, the framework is tested in real-world scenarios where fake speech is generated using popular black-box Web voice-cloning APIs. The results show that the framework consistently outperforms baseline methods, highlighting its robustness and practical applicability.

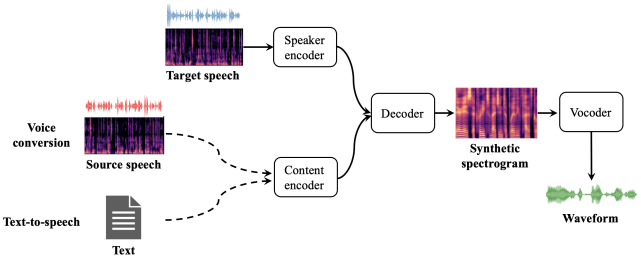


Fig. 1: General pipelines of voice conversion (VC) and text-to-speech (TTS) methods.

II. Background and Related Work

A. Speech Synthesis Attacks

Speech synthesis attacks aim to replicate a target speaker's voice identity using stolen voice samples to generate synthetic speech with the desired voice characteristics and content, which is also known as unauthorized speech synthesis. The history of synthetic speech generation dates back to the 1930s [18], and since then, numerous statistical methods for speech synthesis have been proposed. Recently, with the rapid advancement of Deep Neural Networks (DNNs), DNN-based speech synthesis has garnered significant attention [19], [20] due to its remarkable synthesis quality and accessibility. Figure 1 illustrates the general pipelines of two prevalent types of speech synthesis methods: *voice conversion* (VC) and *text-to-speech* (TTS).

Voice Conversion (VC). VC transforms speech from a source speaker to mimic the vocal characteristics of a target speaker while preserving linguistic content. Most VC systems follow an encoder-decoder framework [21], where a content encoder extracts linguistic features, and a speaker encoder captures the target voice identity. The decoder then synthesizes speech that sounds like the target speaker but retains the original message.

Text-to-Speech (TTS). Similar to VC, TTS extracts linguistic features directly from given texts and employs a similar synthesis framework. Compared to VC, TTS is often more convenient and stealthier for attackers, as some VC requires voice similarity between source and target speakers [22] and may leak the source speaker's identity [23].

Spectrogram Transformation and Vocoder. In speech synthesis pipelines, the spectrogram plays a crucial role. It is a widely used representation of audio for various downstream tasks [24], [25], [26], including speech synthesis, speech translation, or fake speech detection. To compute a spectrogram, the Short-Time Fourier Transform (STFT) is commonly used to divide the raw speech signal into overlapping frames using a window function (e.g., Hamming or Hann). Each

frame is then transformed into the frequency domain using the Discrete Fourier Transform (DFT), producing a complex-valued representation that contains both magnitude and phase information. However, since phase information represents the angular component of the STFT complex output, which is absent in synthetic speech, a vocoder is required to reconstruct the waveform based on a magnitude-only spectrogram (the absolute values of STFT results) with estimated phase information. For simplicity, we refer to the spectrogram as the magnitude-only spectrogram in this paper. Among these vocoders, Griffin-Lim [27] is a classic iterative algorithm that estimates phase information by iteratively minimizing the reconstruction error between the original spectrogram and the spectrogram of the reconstructed waveform. In recent years, many DNN-based vocoders, such as HiFi-GAN [28], MelGAN [29], WaveGlow [30], and WaveRNN [31], have been trained on ground-truth waveform-spectrogram pairs to directly generate waveforms from spectrograms without an iterative process.

B. Fake Speech Detection

Within the broader deepfake defense family [32], [33], [34], [35], [36], [37], [38], fake speech detection serves as the primary defense mechanism against speech synthesis attacks. Typically, detection algorithms extract acoustic features from audio files, such as spectrograms with varying frequency resolutions and derived cepstral features. These extracted features are then passed to a trained classifier, which may be either DNN-based or statistical-based, to generate the final prediction result. In addition, some studies have explored using raw waveforms [39], [12] or speech embeddings extracted from pretrained large audio representation models [40], [41] as inputs, also showing good performance in certain settings.

To enhance fake speech detection, some studies introduce additional explanatory features, such as liveness cues [8], [9] (e.g., heartbeat, breathing patterns, pop noise, or unnatural pauses), as supporting evidence for predictions. Other recent works extend the binary classification task to a multi-task learning framework [10], [14], incorporating objectives such as synthesis model or vocoder classification alongside fake/real classification. These approaches aim to trace the unique fingerprints left by different speech synthesis systems. Among these studies, most focus on optimizing the combination of acoustic features and neural architectures [6], [7]. Others target improvements in training robustness through techniques such as speech augmentation (e.g., adjusting speech speed, pitch, or adding simulated noise) [15], [14] and label-balancing [15]. These advancements have significantly improved detection accuracy in recent years. However, while these methods achieve promising results in intra-dataset evaluations, they still exhibit some key limitations, which we summarize below.

Strong Assumptions and Limited Generalization. Existing detection methods often rely on strong assumptions [42], [43]. For example, some methods detect liveness traces (e.g., heartbeat, breath sounds) or microphone artifacts, typically requiring specific recording conditions (such as distance to the microphone [42]). Others detect artifacts introduced by spectrogram synthesizers or vocoders [11], [44]. For example, some recent works extend the binary classification task to a multi-task learning framework [10], aiming to predict the sample authenticity while tracing its source. However, these approaches have two key limitations. First, reliance on synthesizer labels constrains the detection model's architecture, as the vocoder mappings and categories are fixed during training. As a result, the model struggles to adapt to rapidly evolving synthesis techniques and often overfits to the learned vocoder categories. Second, synthesizer labels are often unavailable, as fake speech commonly circulates online without generation details, and many samples originate from black-box APIs that conceal their synthesis methods to protect intellectual property.

Beyond these limitations, many existing methods exhibit poor generalization performance. While domain shift is a common challenge in machine learning, it is especially severe in fake audio detection. Detection models are highly sensitive to factors such as speaker identity, language, dataset characteristics, input length, synthesis models [45], [10], [12], recording conditions [46], and audio properties like bitrate [47]. For instance, a classifier trained on English speech sees a sharp performance drop when tested on Japanese data [45], even with the same synthesis technique. Similarly, models trained on specific vocoders show high error rates on unseen vocoders or speakers. These issues highlight that detectors often overfit to specific training data or settings, undermining their reliability in real-world applications.

Recent efforts have attempted to improve generalization by applying advanced training strategies such as contrastive learning and speech augmentation to enhance model robustness across varying noise levels, speaking rates, and vocoder transformations. However, these methods still face notable limitations. For example, [14] relies on vocoder labels via a synthesizer prediction stream, which fails to capture artifacts shared across diverse fake speech types. Similarly, [15] addresses generalization by expanding the training corpus, converting real samples into multiple re-synthesized versions using specific vocoders (e.g., HiFi-GAN and WaveGlow) and enforcing balanced mini-batch distributions. Although such approaches outperform many baselines, they still depend on vocoder-specific assumptions and external augmentation techniques, limiting their effectiveness on unseen distributions.

Lack of Artifact Explanation. While many studies focus on fake speech detection, few explore why fake speech contains artifacts or provide insights into how these artifacts manifest. Some research introduces explainable features, such as abnormal pauses [48] or variations in tone and emotion [49], but these low-level artifacts are more common in early-generation synthesis models. Modern systems have largely addressed these issues, producing speech that sounds increasingly natural and human-like. Other works focus on optimizing learning pipelines without adopting a data-driven design perspective. Given the abstract and complex nature of audio signals, detection remains largely opaque. For example, some studies [11], [14], [41] analyze voice embeddings in high-

dimensional feature spaces to identify synthesis-related patterns. However, such high-level features are typically model-specific and lack fine-grained detail, resulting in explanations that are often vague.

III. Detection Challenges

This paper seeks to address the aforementioned limitations by performing a comprehensive analysis of potential artifacts in fake speech and proposing a generalized detection approach that does not depend on specific assumptions or meta-labels (e.g., vocoder type). However, several significant challenges must be overcome.

Audio Signals are Complex. Unlike images or text, audio signals encode rich semantic information across both time and frequency domains, capturing speech content, speaker identity, and recording conditions. These intertwined layers make it challenging to extract reliable acoustic features for fake speech detection. Additionally, due to harmonic effects, speech energy can extend into high frequencies (e.g., up to 8000 Hz), despite the fundamental frequency typically ranging from 90 to 255 Hz [50]. When combined with background noise, these harmonics further increase signal complexity, complicating the identification of artifacts.

Artifacts are Abstract Concepts. Artifacts (subtle signs of unnaturalness) are inherently abstract and difficult to define in speech. While they may appear as abnormal spectrogram textures (e.g., vocoder fingerprints) or irregular energy patterns, such cues are highly dependent on the synthesis model, audio sample, and recording conditions, making them inconsistent and hard to generalize. Moreover, humans are generally less sensitive to audio artifacts than visual ones. For example, visual cues like lip-sync mismatches [51] or shadow inconsistencies [52] are more intuitive and easier to detect. In contrast, audio artifacts often lack perceptual salience, posing greater challenges for identification and labeling.

Data Scarcity and Distribution Gap. Ensuring similar distributions between training and test data remains a major challenge in fake speech detection. Audio datasets are inherently limited due to ethical concerns, restricting diversity in language, content, speakers, and recording environments. While sufficient for general audio tasks like speech recognition, existing datasets lack the scale and variability needed for detecting fake speech. Moreover, domain shift is difficult to overcome by simply increasing data volume, as synthesis techniques evolve rapidly with new vocoders and generation methods, causing trained detectors to lag behind. Although data augmentation, such as adding noise or altering pitch, is sometimes used to increase diversity, these methods risk introducing unnatural artifacts that may compromise authenticity, especially in tasks where signal integrity is critical.

IV. Fake Speech Analysis

To address the aforementioned challenges and develop an effective fake speech detection method, we begin by analyzing fake speech through investigating the following key questions:

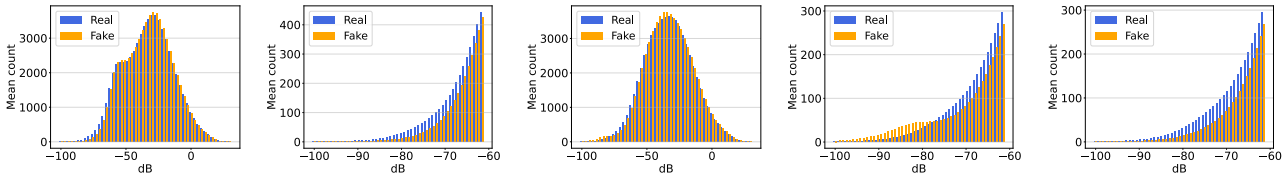
- **Q1:** Can current speech synthesis models generate perfect synthetic speech that is indistinguishable from real speech? If not, what factors prevent perfection, and how are artifacts introduced during the synthesis process?
- **Q2:** What artifacts make fake speech distinguishable, where are they located, and how are they manifested?
- **Q3:** Do these artifacts share generalizable characteristics?

For **Q1**, our investigation shows that current speech synthesis models are incapable of generating perfect synthetic speech, meaning that artifacts in fake speech are nearly unavoidable. The primary reasons are outlined as follows.

First, *artifacts inevitably arise during the vocoding process due to the lack of accurate phase information*. Reconstructing high-quality speech waveforms requires both magnitude and phase components. However, as illustrated in Figure 1, vocoders typically receive only a magnitude spectrogram as input, with no true phase information. In practice, the phase is often approximated or estimated, introducing inherent errors in the reconstruction. These errors are especially pronounced in high-frequency regions (typically associated with smaller magnitudes), which are more sensitive to phase inaccuracies due to their shorter wavelengths [53], as even minor phase shifts can result in significant mismatches in the time domain.

Second, *"naturalness" is not explicitly learnable during the speech synthesis training process*. Most training pipelines for speech synthesis do not directly optimize for perceptual naturalness. Instead, both spectrogram synthesis and vocoder training typically rely on sample-level objectives, aiming to minimize differences between synthetic and ground truth speech using distance-based losses, such as L1 or L2 norms between spectrograms [28], [54]. For example, HiFi-GAN, a GAN-based vocoder, includes a Mel-spectrogram reconstruction loss represented as $\mathcal{L}_{\text{Mel}}(G) = \mathbb{E}_{(x,s)}\left[\|\phi(x) - \phi(G(s))\|_1\right]$, where x is the ground truth waveform, $\phi(\cdot)$ is the waveform-to-spectrogram transformation function, and $G(s)$ is the generated waveform given input state s. This distance-based loss is inherently biased by spectrogram magnitude, as components with larger magnitudes contribute more to the overall loss. From a machine learning perspective, models naturally prioritize focusing on components that have a greater impact on the loss, while neglecting subtle variations. As a result, these underrepresented details may be overlooked, leading to synthesis inconsistencies or speech that overfits specific optimization objectives while disregarding the physical and perceptual principles underlying natural speech.

Third, *dimensionality increase during spectrogram synthesis often introduces artifacts*. This process is analogous to upsampling in image processing, which can introduce artifacts [55]. Simple methods like nearest-neighbor interpolation pad new values based on neighboring pixels, leading to visible distortions or excessive smoothing. Similarly, GAN-based architectures have been shown to introduce artifacts in both pixel and frequency domains during upsampling [55]. For instance, AutoVC [56] performs a transformation $\mathcal{G} : \mathbb{R}^{2\times 256} \rightarrow \mathbb{R}^{80\times T}$, mapping low-dimensional embeddings into an $80 \times T$ Mel-spectrogram. This significant dimensional



(a) WaveFake full (b) WaveFake small (c) LibriSeVoc full (d) LibriSeVoc small (e) LV small (w/o WRNN)

Fig. 2: Distribution of time-frequency point counts across different magnitude ranges.

expansion requires the model to generate details absent from the original representation, often resulting in unnatural energy distributions, spectral inconsistencies, or audible distortions.

The above insights motivate us to explore the potential of detecting fake speech by ***analyzing spectrogram magnitude, a novel perspective that has been largely overlooked in existing fake speech detection research***. Detecting fake speech from the perspective of spectrogram magnitude offers several additional advantages. First, magnitude can roughly separate spoken content (typically encoded in human-perceptible frequency ranges) from noise and synthesis-related artifacts. This enables more **content-independent detection** by focusing on anomalies rather than linguistic variations. Second, magnitude can be analyzed independently of the time axis, supporting more **time-independent methods** that generalize better across speech samples of varying lengths and temporal patterns. Third, partitioning the spectrogram by magnitude ranges retains the original resolution ($F \times T$, where F and T denote the number of frequency and time frames, respectively), offering a **complete and consistent view of the data**, rather than isolating specific segments.

Next, we address the second and third questions (**Q2** and **Q3**) by analyzing the spectrogram magnitude. We begin by computing basic statistics on the distribution of *time-frequency points* across different magnitude ranges, measured in decibels (dB), using multiple datasets. A *time-frequency point* refers to a single magnitude value in the spectrogram matrix, representing the signal's energy at a specific moment in time and a particular frequency.

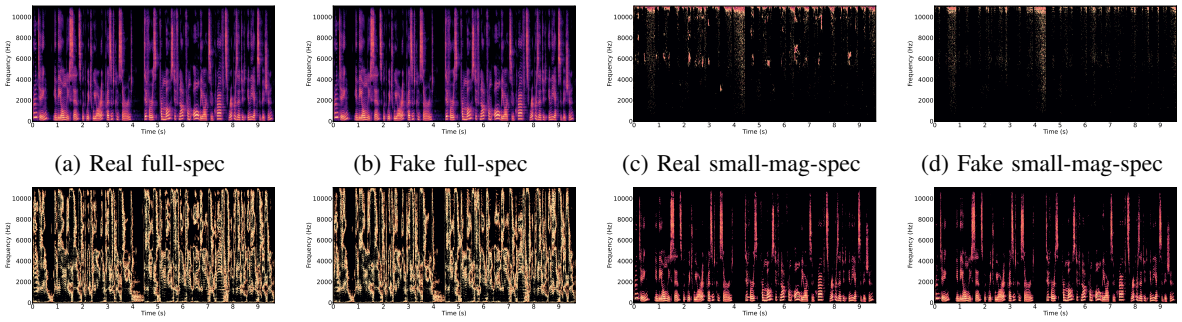
We conduct this analysis using the WaveFake [45] and LibriSeVoc [10] datasets, both of which cover a range of synthesis techniques (WaveFake includes seven vocoders, and LibriSeVoc includes six) making them strong benchmarks for artifact detection. These datasets ensure identical speakers and content across real and fake samples, allowing fair statistical comparison. To standardize the data, all audio is resampled to 16,000 Hz and normalized to 324 time frames (approximately 4 seconds) and 200 frequency frames, yielding spectrograms with 64,800 time-frequency points. We then analyze the magnitude distribution across these points by computing the average number of points within different defined dB intervals for each dataset. This approach provides an overview of how magnitudes are distributed across the spectrogram.

Figure 2a and Figure 2c present the average counts of time-frequency points within different magnitude ranges for the WaveFake and LibriSeVoc datasets (-100 dB to 20 dB, nearly the full magnitude range). The x-axis is divided into 50 bins, and each vertical bar represents the count of the time-frequency points whose corresponding magnitude values are within a specific magnitude range. As we can see, while real and fake speech exhibit similar overall distributions across the entire magnitude range, more pronounced differences emerge within smaller magnitude ranges, as shown in the close-up views in Figure 2b and Figure 2d (-100 dB to -60 dB). For instance, the count of points can differ by more than twofold between real and fake speech in these smaller ranges. **Such significant differences could easily be overlooked without isolating and analyzing specific magnitude ranges.**

In the WaveFake dataset, real speech consistently shows higher counts across nearly all small magnitude ranges (Figure 2b). In contrast, LibriSeVoc exhibits a mixed pattern (Figure 2d): real speech has more points between -80 dB and -60 dB, while fake speech dominates the -100 dB to -80 dB range. Despite this variation, both datasets reveal clear distinctions between real and fake speech in low-magnitude regions. To investigate the cause of this interesting phenomenon in LibriSeVoc, we carefully inspected the distribution of each vocoder and found that WaveRNN largely drives the divergence. Excluding WaveRNN, Figure 2e (LV = LibriSeVoc, WRNN = WaveRNN) shows trends more consistent with WaveFake, where real speech again dominates the small-magnitude range. Figure 4k illustrates a WaveRNN-generated spectrogram, revealing a high-frequency energy boost that inflates point counts.

While the counts of time-frequency points provide basic insights, they only offer a rough overview. This raises an important follow-up question: *what specific differences exist between real and fake speech in the smaller magnitude ranges?* To explore this, we present visual examples and quantitative results from multiple perspectives to highlight the differences between real and fake speech in spectrogram magnitude.

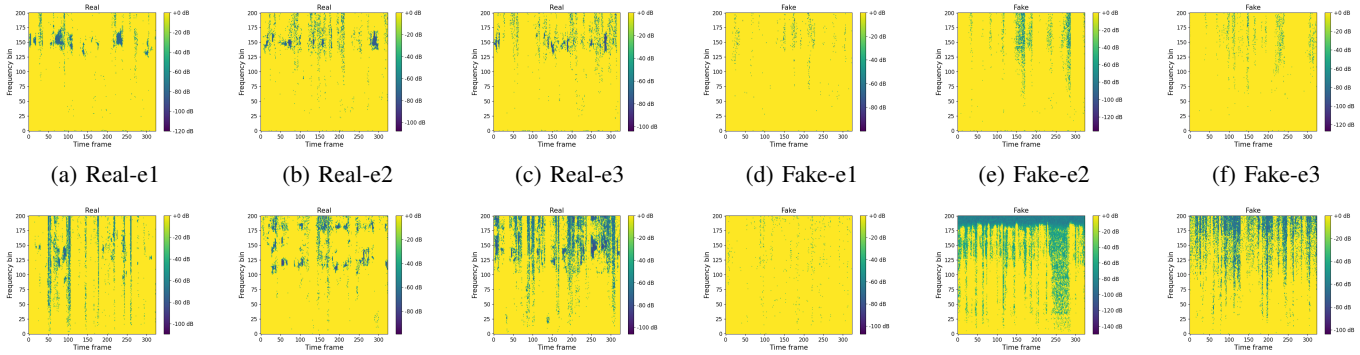
a) Sample-Level Visualizations: We first visualize a real–fake speech pair from the WaveFake dataset (e.g., LJ00001.wav), including full spectrograms and sub-spectrograms for three magnitude ranges: small (-150 dB to -65 dB), middle (-40 dB to -20 dB), and large (-10 dB to 30 dB). The fake sample is generated by HiFi-GAN. As shown in Figure 3, the fake sample lacks detail and energy in the small-magnitude range, while the real sample exhibits rich, irregular textures and clustered patterns. Differences also exist in other ranges, but they are more pronounced in the small-magnitude range. Figure 4 shows additional small-



(a) Real full-spec (b) Fake full-spec (c) Real small-mag-spec (d) Fake small-mag-spec

(e) Real middle-mag-spec (f) Fake middle-mag-spec (g) Real large-mag-spec (h) Fake large-mag-spec

Fig. 3: Spectrograms of real and fake speech samples from the WaveFake dataset, visualized across different magnitude ranges: full range ((a) and (b)), small-magnitude ((c) and (d)), mid-magnitude ((e) and (f)), and large-magnitude ((g) and (h)).



(a) Real-e1 (b) Real-e2 (c) Real-e3 (d) Fake-e1 (e) Fake-e2 (f) Fake-e3

(g) Real-e1 (h) Real-e2 (i) Real-e3 (j) Fake-e1 (k) Fake-e2 (l) Fake-e3

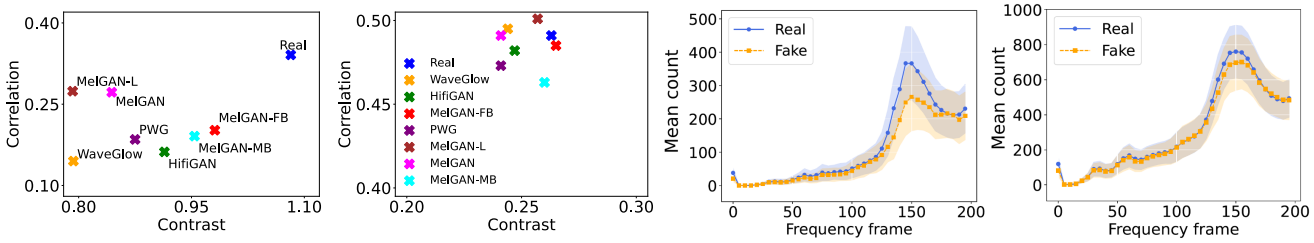
Fig. 4: Examples of small-magnitude spectrograms from WaveFake and LibriSeVoc. (a)–(c): real samples from WaveFake; (d)–(f): fake samples from WaveFake; (g)–(i): real samples from LibriSeVoc; (j)–(l): fake samples from LibriSeVoc.

magnitude spectrograms from WaveFake and LibriSeVoc. We can observe that real speech consistently shows more natural and coherent patterns, while fake speech exhibits noticeable artifacts. In WaveFake, fake samples typically lack texture detail and energy. In LibriSeVoc, fake speech displays varied anomalies, such as extremely sparse distributions (Example (j)) and unnaturally uniform textures (Examples (k) and (l)), where the energy distribution remains overly concentrated or stable across the entire spectrogram. To broaden our analysis, we also visualize real and fake speech samples from two additional widely studied datasets: In-the-Wild [12] and ASVSpoof2019 [57], in Figure 12 of the Appendix. The results consistently show that fake speech exhibits distinct characteristics from real speech in the small-magnitude range.

To further assess whether the observed spectrogram anomalies are generalizable artifacts, we visualize small-magnitude spectrograms from 7 vocoders using the same input and analyze count statistics from 13 vocoders and 5 black-box Web voice-cloning APIs, spanning nine years (2016–2025). The results demonstrate that the lack of detail and energy in small-magnitude ranges is consistently present across vocoders, despite rapid model evolution. More discussion can be found in Section C of the Appendix.

b) Spectrogram Texture and Frequency Analysis: We treat the spectrogram as a one-channel image and apply the Gray-Level Co-occurrence Matrix (GLCM) [58], a widely used statistical tool in image texture analysis, to quantify spatial relationships between time-frequency points. GLCM metrics such as *contrast* and *correlation* measure local pattern variation and dependency: higher values indicate more structured textures, while lower values suggest randomness. On the WaveFake dataset, we compute these metrics for real and fake speech in both small (below -65 dB) and large (-10 dB to 30 dB) magnitude ranges. As shown in Figure 5a, real and fake speech are clearly distinguishable in small magnitudes, whereas in larger magnitudes (Figure 5b), the distinction diminishes. In addition to texture analysis, we examine which frequency bands tend to exhibit missing time-frequency points in fake speech. Figures 5c and 5d present the average frequency-wise point counts under -60 dB and -50 dB, respectively. The results show that fake speech consistently lacks more points in the medium-to-high frequency range (approximately 5000–7000 Hz, around the 150th frequency frame), which typically encodes harmonic and environmental details. This difference is more pronounced in lower-magnitude regions and aligns well with findings in **Q1**.

In summary, our investigation reveals that fake speech generally lacks detail and variation, as evidenced by its texture patterns and energy distribution in smaller-magnitude ranges. While some fake speech samples may exhibit increased energy or clustered patterns, these often serve as additional indicators of fabrication. Despite these observable differences in energy



(a) (b) (c) (d)

Fig. 5: Spectrogram texture and frequency distribution analysis from the WaveFake dataset. (a) GLCM in the small-magnitude range; (b) GLCM in the large-magnitude range; (c) Frequency-wise count under -60 dB; (d) Frequency-wise count under -50 dB.

distribution and pattern details, the concept of "genuineness" remains abstract and cannot be fully captured by these observations and statistics alone. Advanced methods are necessary to effectively uncover hidden artifacts in individual samples, such as energy variation, pattern irregularities, periodic texture repetition, or padding effects, which frequently characterize fake speech samples.

V. Fake Speech Detection Framework

A. Overview of the Proposed Framework

Based on the above analysis, we aim to detect fake speech by focusing on three key aspects: *spectrogram textures*, *spectrogram magnitude distribution*, and *speech synthesis quality consistency across different magnitude ranges*. The core idea is to partition the spectrogram into layered representations based on magnitude and detect artifacts in both the spatial and DCT domains. The motivation for using the DCT domain stems from its proven success in image-based deepfake detection [55], where the fabrication of non-existent details often leaves identifiable traces in the DCT domain. Applying this technique to spectrograms provides a complementary perspective for uncovering subtle synthesis artifacts.

Figure 6 presents an overview of our proposed detection framework. The input speech waveform is first transformed into a spectrogram using STFT. The spectrogram is then partitioned into a layered sequence of 2D sub-spectrograms, each representing a specific range of magnitudes. Correspondingly, the 2D Discrete Cosine Transform (2D-DCT) is applied to each sub-spectrogram to generate a sequence of DCT spectra, resulting in a similar layered representation in the DCT frequency domain. To better capture the dynamic yet continuous variations as magnitude changes, we further develop a novel 3D speech representation. This representation treats the 2D sub-spectrogram sequence or the 2D DCT spectrum sequence as a whole, forming a video-like 3D sub-spectrogram or DCT spectrum stream. In this stream, each "video frame" corresponds to a sub-spectrogram or its DCT spectrum. Consequently, the framework produces four types of input representations: two 2D inputs (the sub-spectrogram and its DCT spectrum) and two 3D inputs (the stream of sub-spectrograms and the stream of DCT spectra). For processing the 2D inputs, we employ ResNet18, a lightweight and widely used convolutional neural network, as the backbone

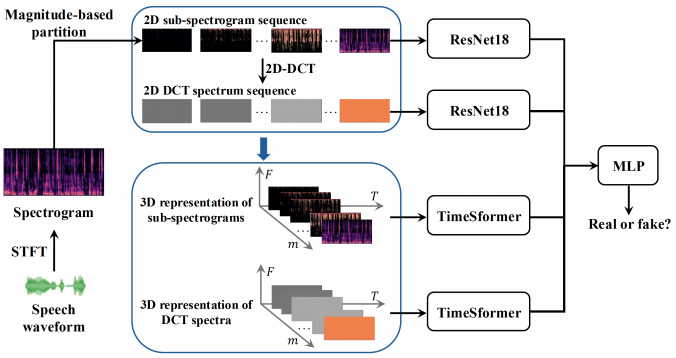


Fig. 6: Overview of the proposed framework.

model. For the 3D inputs, we adopt TimeSformer, a 3D-input classifier based on the Transformer architecture, which has demonstrated outstanding performance in video understanding tasks, such as classification and reasoning. The outputs from these networks are concatenated and passed through a multi-layer perceptron (MLP), which produces the final prediction indicating whether the input speech is fake or real.

This design offers several key advantages. First, it enables the detection of fine-grained artifacts by partitioning the spectrogram into sub-layers based on magnitude ranges. Second, the design incorporates multiple perspectives, analyzing both texture-level patterns and frequency component contributions for a comprehensive evaluation. Third, the proposed framework can identify artifacts not only within individual layers but also across layers, capturing their interdependencies to uncover potential synthesis inconsistencies.

B. Magnitude-Based Spectrogram Partitioning

As introduced in Section II-A, a spectrogram is a widely adopted representation that illustrates how the frequency content of a signal evolves over time. By applying STFT and dividing the speech signal into overlapping frames with a window function, each frame is transformed into the frequency domain using the Discrete Fourier Transform (DFT), resulting in $STFT(f,t) = \sum_{n=0}^{N-1} x[n] \cdot w[n - tH] \cdot e^{-j2\pi fn/N}$, where $x[n]$ is the discrete-time audio signal, $w[\cdot]$ is the window function, t is the frame index, H is the hop size, N is the number of DFT points, and f is the frequency bin index. In this paper, we use $X = [X_{f,t}]_{F \times T}$ to represent the spectrogram,

where $X_{f,t} = |STFT(f,t)|$ is the magnitude of a specific time-frequency point in the spectrogram.

To enable a more detailed analysis of the spectrogram magnitude, we partition the raw spectrogram X into sub-spectrograms based on predefined magnitude thresholds. For example, a sub-spectrogram $X_{\text{sub}}^{(\tau_{\min},\tau_{\max})} = [X'_{f,t}]_{F\times T}$ corresponding to a specific magnitude range is derived based on

$$X'_{f,t} = \begin{cases} X_{f,t}, & \text{if } \tau_{\min} \le X_{f,t} < \tau_{\max}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $\tau_{\min}$ and $\tau_{\max}$ represent the lower and upper bounds of the magnitude range for the sub-spectrogram, respectively. In this paper, we transform the raw spectrogram into multiple overlapping sub-spectrograms, where smaller magnitudes are always included while larger magnitudes are progressively added. Using this approach, the original spectrogram X is partitioned into a sequence of sub-spectrograms $\{X_{\text{sub}}^{(\tau_0,\tau_1)}, X_{\text{sub}}^{(\tau_0,\tau_2)}, \ldots, X_{\text{sub}}^{(\tau_0,\tau_i)}, \ldots, X_{\text{sub}}^{(\tau_0,\tau_m)}\}$, where τ_0 represents the lower bound of the magnitude, and $\tau_i (1 \le i \le m)$ is the upper bound of the magnitude for each sub-spectrogram. This magnitude-based partitioning method divides the original full spectrogram into m layers, forming a sequence of 2D sub-spectrograms. By providing a multi-layered representation, this approach enables the detector to analyze the data at varying levels of granularity and isolate distinct layers of information embedded within the spectrogram. The magnitude upper bounds can be determined using various methods, such as absolute magnitude values or thresholds that ensure equal increments in time-frequency points (e.g., the first sub-spectrogram contains 5,000 time-frequency points, and the second contains 10,000 time-frequency points).

Additionally, this partitioning approach can produce a video-like 3D representation when the 2D sub-spectrogram sequence is viewed as a whole, with each "video frame" corresponding to a sub-spectrogram that incorporates progressively larger magnitude ranges. This resembles a spectrogram "growing" from a detail-focused representation to a complete representation, allowing the detection model to effectively capture dynamic yet continuous changes in patterns across magnitude layers. This approach can also highlight synthesis quality inconsistencies or abnormal energy loss or concentration within specific magnitude ranges, which deviate from natural patterns.

C. Detecting Artifacts in the DCT Domain

While the STFT spectrogram is a widely used and effective audio feature, it primarily represents the physical frequency components of a signal. However, for fake speech detection, it is also crucial to analyze the "rhythm" of magnitude variations, such as the smoothness, abruptness, or consistency of textures in the spectrogram. Based on our analysis in Section IV, fake speech often lacks natural variation, appearing overly uniform or missing the irregularities typically present in real speech.

Motivated by the observation that up-sampling processes can introduce artifacts in frequency coefficient distributions, we apply the Type-II 2D Discrete Cosine Transform (2D-DCT) [59] to enhance artifact detection in the proposed spectrogram representation. Certain distribution-level artifacts may not be directly observable in the STFT spectrogram alone. The 2D-DCT, unlike its 1D counterpart, captures subtle patterns across both the time and frequency dimensions, such as unnatural variances, periodic stripes, or uniform distributions. This transformation allows the detection of nuanced artifacts while preserving the original sample dimensions. Specifically, we apply the 2D-DCT to each sub-spectrogram. The output expresses a finite set of data points as a combination of cosine functions oscillating at different frequencies.

Suppose the DCT-transformed representation of the spectrogram $X \in \mathbb{R}^{F\times T}$ is denoted as the matrix $D \in \mathbb{R}^{F\times T}$. The 2D-DCT is given by a function $\mathcal{D} : \mathbb{R}^{F\times T} \to \mathbb{R}^{F\times T}$ that maps $X = [X_{f,t}]_{F\times T}$ to its DCT frequency representation $D = [D_{k_f,k_t}]_{F\times T}$, where

$$\begin{aligned} D_{k_f,k_t} =& w(k_f)w(k_t) \sum_{f=0}^{F-1}\sum_{t=0}^{T-1} \\ & X_{f,t} \cos\left[\frac{\pi}{F}\left(f+\frac{1}{2}\right)k_f\right] \cos\left[\frac{\pi}{T}\left(t+\frac{1}{2}\right)k_t\right] \end{aligned} \tag{2}$$

for $\forall k_f = 0,1,2,\ldots,F-1$ and $\forall k_t = 0,1,2,\ldots,T-1$. Here, $w(k_f)$ and $w(k_t)$ denote normalization factors. The resulting matrix D contains coefficients that represent the contribution of different frequency components along the frequency axis and the time axis. Such 2D-DCT transformation can be applied to any sub-spectrogram defined above, resulting in a sequence of DCT spectra. Furthermore, by considering those DCT spectra as whole, this approach also provides a video-like DCT representation, which is similar to the proposed spectrogram-based 3D representation in Section V-B.

D. Detection Using Multi-Level Features and Weighted-Scaling Training

To explore artifacts across both the spatial and DCT domains as well as within different magnitude ranges, we propose a multi-branch detection framework that leverages diverse feature representations to enhance decision-making and robustness.

Specifically, using the magnitude partition approach discussed in Section V-B, the spectrogram of the input speech sample is transformed into a sequence of 2D sub-spectrograms $\{X_{\text{sub}}^{(\tau_0,\tau_1)}, X_{\text{sub}}^{(\tau_0,\tau_2)}, \ldots, X_{\text{sub}}^{(\tau_0,\tau_i)}, \ldots, X_{\text{sub}}^{(\tau_0,\tau_m)}\}$, where each sub-spectrogram corresponds a specific magnitude range. By treating this sequence as a whole, we derive a 3D representation $X_v \in \mathbb{R}^{F\times T\times m}$. Similarly, by applying the 2D-DCT transformation $\mathcal{D}$ (defined in Section V-C) to each sub-spectrogram, we can derive a sequence of DCT spectra $\{X_{\text{sub-dct}}^{(\tau_0,\tau_1)}, X_{\text{sub-dct}}^{(\tau_0,\tau_2)}, \ldots, X_{\text{sub-dct}}^{(\tau_0,\tau_i)}, \ldots, X_{\text{sub-dct}}^{(\tau_0,\tau_m)}\}$ and its corresponding 3D representation $X_v^{\text{dct}} \in \mathbb{R}^{F\times T\times m}$.

For the 2D inputs ($X_{\text{sub}}^{(\tau_0,\tau_i)}$ or $X_{\text{sub-dct}}^{(\tau_0,\tau_i)}$), we employ ResNet18 [16] as the detection backbone. ResNet18 is a lightweight and widely used convolutional neural network

that features residual connections to mitigate gradient vanishing and degradation issues. Its architecture consists of basic blocks incorporating 3×3 convolutional layers, batch normalization, and ReLU activation functions. For the 3D inputs (X_v or X_v^{dct}), we adopt TimeSformer [17] as the detection backbone. Originally developed for video understanding tasks, TimeSformer excels at capturing temporal and spatial interdependencies between video frames and demonstrates strong reasoning capabilities. In our framework, each sub-spectrogram or DCT spectrum is treated as a "video frame", with the key distinction being that a sub-spectrogram or DCT spectrum is a mono-channel input rather than a 3-channel RGB image. The TimeSformer architecture we use consists of 12 Transformer blocks, each equipped with 8 attention heads and an embedding dimension of 768. Each sub-spectrogram or DCT spectrum is divided into 16×16 patches, with positional encodings applied as described in the original paper. By employing a "divided space-time" (or "space-magnitude" in our context) attention mechanism, the model first applies attention on different magnitude ranges we partitioned before and then spatial attention within each partitioned layer to conduct reasoning over different layers while also maintaining focus on individual sub-spectrograms or DCT spectrum.

To ensure each neural network specializes in distinct features while maintaining compatibility with uniform input formats, we adopt a two-step training process. First, we train the four networks (ResNet18 for 2D inputs and TimeSformer for 3D inputs) independently on their respective representations. Afterward, the trained models' parameters are frozen, and their outputs are concatenated and passed through a multilayer perceptron (MLP) for fine-tuning with ground-truth labels. This integration step combines predictions from all branches to produce a comprehensive and robust detection result.

During training, to prevent overfitting and ensure diverse feature learning, we introduce a shuffling mechanism. Sub-spectrograms from multiple speech samples are randomly shuffled to create a new training set:

$$D_s = \text{Shuffle}(\bigcup_{k=1}^{K} \{X_{\text{sub}}^{k(\tau_0,\tau_1)}, X_{\text{sub}}^{k(\tau_0,\tau_2)}, \ldots, X_{\text{sub}}^{k(\tau_0,\tau_i)}, \ldots, X_{\text{sub}}^{k(\tau_0,\tau_m)}\}), \tag{3}$$

where $X_{\text{sub}}^{k(\tau_0,\tau_i)}$ represents the i-th sub-spectrogram of the k-th sample. Similarly, a shuffled set of DCT spectra, denoted as $D_{s-\text{dct}}$, is also created.

For the 2D branches, we employ a weighted loss function that assigns different weights to sub-spectrograms (or DCT spectra). The primary objective is to encourage the network to make accurate predictions even with limited information (i.e., when the magnitude upper bound is smaller). This strategy motivates the model to focus less on content-relevant features (typically found in sub-spectrograms with magnitudes ranging from -10 dB to 30 dB, which predominantly include the fundamental frequency components of speech) and instead prioritize spectrogram details that often reside in higher frequency and smaller magnitude ranges. Since this work does not rely on any vocoder-specific information, we adopt a weighted cross-entropy loss with binary labels (real and fake). The loss is computed across all sub-spectrograms (or DCT spectra) in a batch sampled from the shuffled dataset D_s (or $D_{s-\text{dct}}$) and is defined as $\mathcal{L}_{2D} = -\frac{1}{B}\sum_{j=1}^{B} c_j \left[y_j \log(\hat{y}_j) + (1-y_j)\log(1-\hat{y}_j)\right]$, where B is the batch size, $y_j \in \{0,1\}$ is the ground truth label for the j-th sub-spectrogram (or DCT spectrum) in the batch. $y_j = 1$ denotes the sub-spectrogram (or DCT spectrum) is real (derived from a real speech sample), and $y_j = 0$ denotes it is fake. $\hat{y}_j$ is the prediction for the j-th sub-spectrogram (or DCT spectrum). $c_j \in (0,1)$ is the weight assigned to the j-th sub-spectrogram (or DCT spectrum), which depends on the corresponding magnitude upper bound τ_j. c_j is calculated as: $c_j = \frac{\tau_{\text{MAX}} - \tau_j}{\tau_{\text{MAX}} - \tau_{\text{MIN}}}$, where τ_{MAX} and τ_{MIN} are the reference maximum and minimum magnitudes of the dataset. This weighting mechanism ensures that sub-spectrograms with smaller magnitude upper bounds (i.e., less complete information) are assigned larger weights, forcing the network to pay greater attention to subtle details that often carry important artifacts.

For the 3D branches, where the sequence of sub-spectrograms (or DCT spectra) is treated as a cohesive whole, we use the standard cross-entropy loss for training each TimeSformer. The loss function is defined as $\mathcal{L}_{3D} = \frac{1}{B}\sum_{j=1}^{B} \text{CE}(y_j, \hat{y}_j)$. We also introduce a layer-dropping strategy that randomly removes a subset of magnitude layers from a speech sample during training, aiming to reduce layer overlap and prevent the model from overfocusing on specific layers. More details can be found in Section A of the Appendix.

During the testing phase, the spectrogram of each test speech sample is partitioned into m layers, and the corresponding four types of input representations are derived as described earlier. The outputs from the four branches processing these representations are then concatenated and passed through a MLP to generate the final prediction.

The framework integrates predictions from different sub-spectrograms and branches to produce a robust detection result. By aggregating detection results across sub-spectrograms representing varying magnitude ranges, the framework effectively captures inconsistencies in synthesis quality, enhancing its ability to distinguish fake from real speech.

VI. Performance Evaluation

A. Datasets

We first use the following four public audio deepfake datasets to evaluate the performance.

WaveFake [45]. WaveFake is a self-synthesized fake speech detection dataset. It leverages seven pre-trained neural vocoders: MB-MelGAN, Parallel WaveGAN (PWG) [60], Full-band MelGAN (FB-MelGAN), HiFi-GAN [28], MelGAN [29], MelGAN-large (MelGAN-L), and WaveGlow [30]. It consists of two subsets: English (EN) and Japanese (JP).

LibriSeVoc [10]. LibriSeVoc is another self-synthesized dataset that expands WaveFake by adding vocoders and more speakers. Specifically, it utilizes WaveNet [1], WaveRNN [31],

WaveGrad [61], MelGAN, Parallel WaveGAN (PWG), and DiffWave [54] as vocoders.

In-the-Wild [12]. This dataset comprises speeches from English-speaking celebrities and politicians, collected from diverse media sources, simulating a real-world scenario.

ASVSpoof2021 [62]. ASVSpoof2021 features spoofing samples generated using various voice conversion (VC) and text-to-speech (TTS) techniques. We use the train/dev sets to train all detection models and the evaluation set (DF21) to assess the performance of the detection algorithms.

B. Baseline Methods

RawNet2 [39]. RawNet2 is a hybrid model that combines convolutional neural networks (CNNs) and gated recurrent units (GRUs). It extracts speech representations directly from raw audio signals and has demonstrated reliable performance, particularly in the ASVSpoof2019 challenge.

LFCC-LCNN [63]. Linear Frequency Cepstral Coefficients (LFCC) are computed using a linear filterbank on the spectrogram. We adopt a lightweight CNN as the detection model.

RawNet2-Voc [10]. This method uses RawNet2 as the backbone detection model and extends the binary classification task to multi-task learning, incorporating a synthesizer prediction task to enhance detection performance.

Wav2Vec2 [40]. Wav2Vec2 is a large-scale pretrained model developed by Meta AI. It is designed to learn universal speech signal representations for various downstream speech tasks, including fake speech detection.

VoiceRadar [41]. VoiceRadar analyzes the frequency distribution of speech embeddings extracted from HuBERT [64], a self-supervised speech representation model, and leverages these insights to augment the detector training process.

Dual-Stream [14]. Similar to RawNet2-Voc, this approach uses a multi-task framework with a synthesizer prediction stream for classifying vocoders and a content stream for predicting vocoder-independent pseudo-labels (e.g., speech speed and codec type), enhanced by contrastive learning.

Trident [15]. It utilizes Wav2Vec2 as the front end and integrates various strategies into the downstream task design. It incorporates speech re-synthesis and augmentation to enhance training diversity and sample balance.

C. Evaluation Metric and Other Settings

Evaluation Metric. Following prior studies [14], [10], we adopt **Equal Error Rate (EER)** as the primary metric. EER is the point on the ROC curve where the False Acceptance Rate (FAR) equals the False Rejection Rate (FRR). A lower EER indicates better performance.

Data Preprocessing and Other Settings. All speech samples are resampled to a uniform 16kHz sampling rate, and their raw waveforms are either trimmed or padded to a length of 4 seconds. We adopt the STFT spectrogram as the base feature, converting it to logarithmic scale for deep learning tasks. The number of FFT points is set to 400, resulting in spectrogram representations of size 200×324 for all speech samples. For both 2D and 3D input branches, we define dB upper bounds of -70, -65, -60, -55, -45, -35, -10, and 30, yielding eight individual image-like sub-spectrograms (2D) and video-like spectrogram streams (3D) with eight frames. We apply the 2D-DCT transformation to all representations, generating corresponding outputs of the same shape. The Adam optimizer is used for both ResNet18 and TimeSformer. Batch sizes are set to 128 for ResNet18 and 16 for TimeSformer, with learning rates of 0.0005 and 0.0001, respectively. Other setting details can be found in Section A of the Appendix.

TABLE I: Detection EER (%) in the intra-dataset setting.

Method	Dataset		
	WaveFake	LibriSeVoc	Wild
RawNet2	6.8	6.6	0.9
LFCC-LCNN	0.2	1.3	0.4
Wav2Vec2	0.4	1.4	1.1
VoiceRadar	1.2	2.0	0.7
Trident	0.7	0.9	1.2
RawNet2-Voc	3.9	3.1	-
Dual-Stream	0.2	**0.5**	-
Ours	**0.1**	**0.5**	**0.3**

D. Performance on Public Datasets

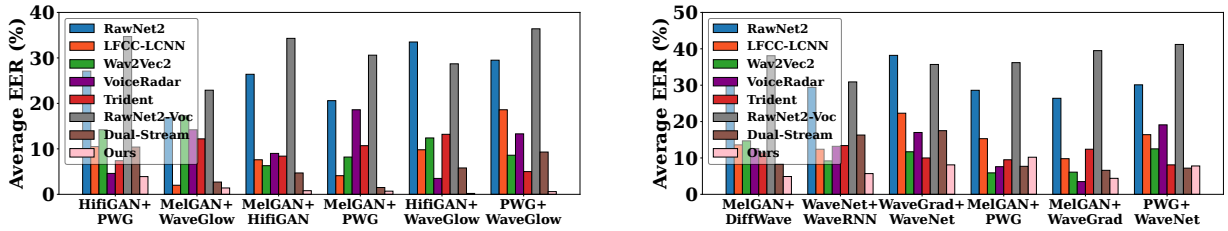
1) Evaluation in the Intra-Dataset Setting: We begin by evaluating the proposed framework in the Intra-dataset setting, where the training and testing sets are split from the same dataset. Specifically, we follow the 6/2/2 train/dev/test splits used in [14] and [10] for the WaveFake, LibriSeVoc, and In-the-Wild datasets. Table I presents the intra-dataset evaluation results for these datasets, showing that our framework achieves the best detection performance. Note that RawNet2-Voc and Dual-Stream are not applicable to the In-the-Wild dataset due to the absence of vocoder information. We also observe that most detection algorithms achieve an EER of less than 5% across the three datasets. This strong performance can be attributed to the ability of these methods to capture effective features that distinguish fake and real samples.

2) Evaluation in the Cross-Method Setting: This setting simulates real-world scenarios where unseen or newly released vocoders are not included during the training of the detection model. To evaluate the generalizability of our proposed framework, we consider two scenarios: (1) **Leave-One-Out**, where the detection model is trained on samples generated by all vocoders except one within the dataset and tested on the unseen vocoder, and (2) **Leave-Most-Out**, where the detection model is trained on samples generated by only two vocoders within the dataset and tested on samples generated by all remaining unseen vocoders.

Table II presents the results for the Leave-One-Out scenario on the WaveFake dataset. As described in Section VI-A, the WaveFake dataset utilizes seven pre-trained neural vocoders. The results show that our proposed framework achieves the best average EER on this dataset in the Leave-One-Out scenario, despite being assumption-free and not relying on vocoder labels during training. Compared to baseline methods that utilize vocoder information during training, such as

TABLE II: Detection EER (%) on the WaveFake dataset in the Leave-One-Out scenario.

Method	Unseen vocoder							Average
	MelGAN	MelGAN-FB	MelGAN-MB	MelGAN-L	HiFiGAN	PWG	WaveGlow	
RawNet2	2.3	24.2	18.3	4.8	20.9	7.4	17.3	13.6
LFCC-LCNN	3.6	44.1	12.6	1.1	4.7	0.9	9.8	11.0
Wav2Vec2	6.9	10.2	3.1	3.0	12.6	9.7	2.9	6.9
VoiceRadar	5.3	7.3	17.6	10.2	9.4	4.4	13.8	9.7
Trident	3.2	15.6	2.7	4.0	5.9	5.1	1.9	5.5
RawNet2-Voc	0.6	40.2	9.3	27.4	36.3	22.4	30.0	23.7
Dual-Stream	9.6	1.2	0.6	**0.1**	8.7	4.1	**0.2**	3.5
Ours	**0.1**	**0.3**	**0.1**	**0.1**	**1.0**	**0.2**	0.4	**0.3**



(a) The WaveFake dataset

(b) The LibriSeVoc dataset

Fig. 7: The average detection EER (%) in the Leave-Most-Out scenario.

RawNet2-Voc and Dual-Stream, our framework demonstrates superior transferability. In Section B of the Appendix, we also present the results for this scenario on the LibriSeVoc dataset in Table VII, which further demonstrates that our framework achieves the best overall performance.

The results also reveal a key limitation of baseline methods: while they may perform well on specific vocoders, they often exhibit significant performance degradation on unseen vocoders not included during training. For example, as shown in Table II, LFCC-LCNN and RawNet2-Voc achieve EERs of 0.9% on PWG and 0.6% on MelGAN, respectively, but their EERs increase dramatically to 44.1% and 40.2% on MelGAN-FB when MelGAN-FB is excluded during training. This demonstrates that these methods fail to generalize effectively and struggle to develop a robust understanding of fake speech across unseen vocoder categories.

Another observation is that incorporating vocoder prediction tasks does not consistently improve performance. For example, while RawNet2-Voc (leveraging vocoder information during training) lowers the EER by 2.9% and 3.5% under the regular intra-dataset setting (Table I) on the WaveFake and LibriSeVoc datasets, respectively, its average EER increases by approximately 10% on both datasets in the Leave-One-Out scenario.

Furthermore, both Trident and Dual-Stream achieve better performance than other baseline methods in terms of average EER, as they incorporate advanced training strategies and speech augmentation techniques. However, they still fail to achieve consistently low EERs across all unseen vocoders. It is worth noting that both methods rely on re-synthesizing or external tools to expand the training data. In contrast, our proposed framework is purely data-driven and does not depend on any external methods. Despite this, our method consistently achieves the best or highly competitive results in most cases.

For the Leave-Most-Out scenario, the performance of our proposed framework compared to baseline methods is shown in Figure 7. In this scenario, the detection model is trained on samples generated by only two vocoders (shown on the horizontal axis in Figure 7) within the dataset and tested on samples generated by all remaining unseen vocoders. Figure 7 reports the average EER across the unseen vocoders for each case. From the results, we observe that our proposed framework demonstrates stronger transferability on the WaveFake dataset, achieving the best performance in all cases, even when trained on only two vocoders. On the LibriSeVoc dataset, our framework also achieves the best overall performance, although a few baseline methods outperform it in specific combinations of two vocoders used for training, such as MelGAN+PWG and PWG+WaveNet. Most other baseline methods exhibit unstable performance, with average EERs fluctuating significantly depending on the vocoder combinations used during training.

3) Evaluation in the Cross-Language/Dataset Settings: Next, we evaluate the proposed framework in cross-language and cross-dataset settings, with the results presented in Table III. For the cross-language setting, we assess performance using the English (ENG) and Japanese (JP) subsets of the WaveFake dataset. As shown in Table III, ENG→JP denotes a model trained on the English subset and tested on the Japanese subset, while JP→ENG represents the reverse scenario. For the cross-dataset setting, we consider two scenarios. In the first scenario (denoted as LV→WF), the model is trained on LibriSeVoc and tested on WaveFake, and in the second scenario (denoted as ASV19→DF21), the model is trained on ASVSpoof2019 and tested on DF21, which is derived from

TABLE III: The average detection EER (%) in cross-language/dataset settings.

Method	ENG→JP	JP→ENG	LV→WF	ASV19→DF21	Average
RawNet2	29.3	41.6	21.3	23.0	28.8
LFCC-LCNN	**6.1**	18.8	8.6	22.4	14.0
Wav2Vec2	15.2	12.3	26.4	29.1	20.8
VoiceRadar	14.9	22.3	10.7	19.1	16.8
Trident	18.6	14.1	21.2	**16.8**	17.7
RawNet2-Voc	36.5	34.7	23.9	22.4	29.4
Dual-Stream	22.7	16.9	5.4	18.6	15.9
Ours	12.6	**8.3**	**4.7**	18.0	**10.9**

ASVSpoof2019. It includes many newer and unseen synthesis techniques not present in the training set. Table III shows that our framework achieves the best overall performance, with the lowest average EER of 10.9%. Notably, the performance of our framework in the cross-language setting demonstrates the robustness of small-magnitude features, which appear less dependent on content or language. Interestingly, the basic LFCC-LCNN model achieves the second-best overall performance, surpassing advanced baseline methods like Dual-Stream and Trident, which have shown strong results in other settings. However, all methods, including ours, show a general decline in performance when evaluated across different datasets and languages, likely due to inherent challenges such as variations in speakers, recording conditions, and data preprocessing techniques across datasets.

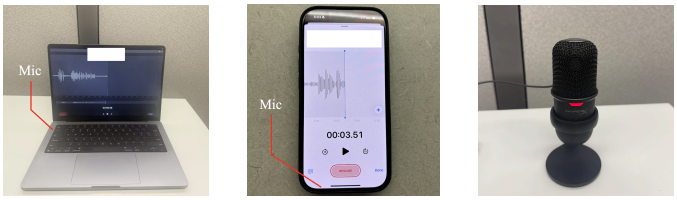
E. Real-World Evaluation

Although public datasets cover a range of speech synthesis technologies, they may not fully represent the capabilities of modern free or commercial voice-cloning tools. These web APIs are highly accessible and often support zero-shot synthesis, requiring only a short voice clip (approximately 10 seconds) and input text to generate high-quality synthetic speech. However, limited research has examined the threats posed by these APIs. To address this gap, we conduct a case study by recruiting real-world participants and collecting both real recordings and synthetic samples generated via publicly available voice-cloning APIs[1].

Web Voice-Cloning APIs. We select 5 publicly available voice-cloning tools: ElevenLabs [65], Speechify [66], FishAudio [67], PlayHT [68], and DupDub [69]. These APIs are chosen via a Google search using keywords like "free voice-cloning" and "free AI speech synthesis". Selection criteria include accessibility and ease of use, favoring APIs that require minimal setup. These tools often incorporate state-of-the-art speech synthesis techniques with strong adaptability and performance, allowing them to mimic the voices and accents of unseen speakers with remarkable accuracy.

Data Collection. We recruit 13 English-speaking participants (8 male and 5 female) aged between 19 and 45 years. Each participant provides 10 voice recordings of the Rainbow Passage [70], a widely used corpus containing common English syllables. Each recording contains 1-2 phrases

[1]Ethics considerations are discussed in Section VIII.



(a) Laptop Mic (b) Phone Mic (c) USB Mic

Fig. 8: Recording devices.

TABLE IV: Detection EER (%) for Web voice-cloning APIs.

Method	Voice-cloning API					Average
	ElevenLabs	FishAudio	DupDub	Speechify	PlayHT	
RawNet2	46.4	72.5	54.1	69.7	41.0	56.7
LFCC-LCNN	38.6	24.7	43.1	32.5	35.4	34.9
Wav2Vec2	24.9	56.2	38.9	37.5	44.0	40.3
VoiceRadar	23.5	32.9	19.7	13.4	19.3	21.8
Trident	24.7	25.8	29.4	31.9	18.5	26.1
Ours	**6.8**	**9.4**	**8.3**	**12.6**	**4.5**	**8.3**

from the passage. Recordings are made using a MacBook (6 participants), iPhone (4), or USB microphone (3), with microphones positioned 6–12 inches from the mouth in various environments (bedrooms, offices, a classroom). The adopted recording devices are shown in Figure 8. For speech synthesis, we randomly select 2 recordings per participant to generate synthetic speech using voice-cloning APIs, with the remaining 8 used as real samples. All synthesized content is also drawn from the Rainbow Passage. In total, we collected 130 fake and 104 real speech samples.

Detection Performance. We train our detection model using a hybrid dataset combining the training sets of WaveFake, LibriSeVoc, In-the-Wild, and ASVSpoof2021, totaling 50,000 samples evenly drawn from the four sources. To account for real-world compression artifacts, we follow [14] and augment the training set with two common formats: MP3 and AAC. For baseline comparison, we include RawNet2, LFCC-LCNN, Wav2Vec2, VoiceRadar, and Trident. RawNet2-Voc and Dual-Stream are excluded due to their reliance on vocoder labels, which are unavailable in the In-the-Wild dataset. Table IV presents detection results on synthetic speech generated using participants' voices. The results show that our method consistently outperforms all baselines, demonstrating strong generalization to black-box Web voice-cloning APIs where the synthesis process is entirely unknown.

F. Performance under Adaptive Attacks

In practice, an attacker may be aware of the proposed framework and attempt an adaptive attack to evade detection. To evaluate the robustness of our framework under such conditions, we consider the following adaptive attack strategies.

- **Projected Gradient Descent (PGD) Attack.** PGD [71] is an adversarial attack method that perturbs the original input to induce misclassification (e.g., crafting adversarial speech to evade detection). However, applying a universal perturbation across all branches and all layers of our proposed

TABLE V: Detection EER (%) under adaptive attacks.

Vocoders for training	PGD-R	PGD-T	Edit-2L	Edit-4L
MelGAN + HifiGAN	1.4	0.8	2.6	1.3
PWG + WaveGlow	1.0	0.9	1.8	0.6

framework is nearly infeasible. This is primarily due to the non-differentiable nature of the layer partitioning operation, where perturbations optimized for one layer may lose their effectiveness after being divided into sub-layers. Moreover, generating a single perturbation that successfully attacks both the 2D and 3D branches is extremely challenging, given their architectural differences and the fact that they operate in distinct domains (pixel and DCT), and to our knowledge this has not been addressed by existing adversarial methods. As a case study, we therefore perform independent PGD attacks on the ResNet18 and TimeSformer, constraining the perturbation norm to 4 for both inputs.

- **Post-Editing Attack.** In this strategy, the attacker attempts to enhance the realism of synthetic speech by introducing fabricated details. To evade detection, the attacker may replace small-magnitude sub-spectrogram layers of the synthetic speech with those extracted from real samples. Specifically, we consider scenarios where the attacker replaces the two or four lowest-magnitude layers.
- **Victim-Overfitting Attack.** While most victims are typically not included in the training set of speech synthesis models, we consider a highly targeted scenario where an attacker fine-tunes a synthesis model using a victim's voice to improve the synthesis quality for that individual. As a case study, we adopt AutoVC [56] and SV2TTS [20] as representative VC and TTS models, respectively. We select 20 speakers from the VCTK [72] and LibriSpeech datasets [73] (none of whom are present in the original training sets) to serve as specific victims for this evaluation.

Detection Performance. For the PGD and post-editing attacks, we evaluate detection performance under the leave-most-out setting using the WaveFake dataset. Specifically, we consider two training scenarios with different vocoders: one using MelGAN+HiFiGAN and the other using PWG+WaveGlow, while testing is performed on samples from the remaining unseen vocoders. Table V reports the detection EER under these two adversarial strategies. As an example, we consider the detection model trained on the PWG and WaveGlow vocoders. For the PGD attack, the EERs (%) are 1.0 and 0.9 when targeting ResNet18 (PGD-R) and TimeSformer (PGD-T), respectively. For the post-editing attack, the EERs (%) are 1.8 and 0.6 when modifying the two smallest layers (Edit-2L) and four smallest layers (Edit-4L), respectively. The results indicate that our method remains robust, even when attackers are aware of the detection framework and attempt to bypass it. For the victim-overfitting attack, the EER before fine-tuning is 1.9 and becomes 1.7 after fine-tuning with the victim's voice. This shows that personalized fine-tuning fails to meaningfully degrade detection performance, reinforcing the challenge of achieving undetectable synthesis.

TABLE VI: Detection EER (%) for the ablation study.

Detection task	w/o 2D-spec	w/o 2D-dct	w/o 3D-spec	w/o 3D-dct
Intra-dataset	4.8	0.3	2.7	0.7
Cross-method	5.1	6.5	3.4	8.5

G. Ablation Study

Our framework comprises four components to handle distinct input representations. To assess their contributions to detection performance, we conduct an ablation study in four scenarios: 'w/o 2D-spec', 'w/o 2D-dct', 'w/o 3D-spec', and 'w/o 3D-dct'. These scenarios involve omitting components responsible for processing 2D sub-spectrograms, 2D DCT spectra, 3D sub-spectrograms, and 3D DCT spectra, respectively. The evaluation, performed on intra-dataset and cross-method detection tasks using the LibriSeVoc dataset, is summarized in Table VI. For comparison, the original framework achieves average EERs of 0.5% and 4.6% on these tasks. The results reveal that the 2D sub-spectrogram sequence is critical for intra-dataset detection performance. As the fundamental time-frequency representation of a speech signal, it encapsulates rich and detailed information, and its omission results in a significant performance drop. Similarly, the 3D representation of sub-spectrograms plays an essential role, with its removal leading to a notable increase in EER. In the cross-method detection task, the components related to DCT representations demonstrate their importance in enhancing detection generalization. The 3D representation of DCT spectra is particularly impactful, as its absence leads to the most significant performance decline. This result underscores the strength of the DCT frequency domain in identifying subtle, distribution-level artifacts. By capturing general patterns and hidden inconsistencies, the DCT components effectively generalize across unseen synthesis methods.

Additionally, we study the impact of the number of sub-spectrogram layers and sample length, with supplementary results provided in Section B of the Appendix.

VII. Conclusion

In this paper, we addressed the limitations of existing fake speech detection methods by introducing a novel assumption-free and generalized framework centered on the magnitude representation of spectrograms. Our analysis reveals consistent artifacts in fake speech, such as reduced texture detail, repetitive patterns, and inconsistencies across magnitude ranges. Leveraging these insights, our framework partitions spectrograms into layered magnitude-based representations and detects artifacts in both spatial and DCT domains using 2D and 3D inputs. Extensive experiments on public datasets show that our method achieves state-of-the-art performance with strong generalizability to unseen models, speakers, and languages, and its robustness is further validated in real-world scenarios involving black-box Web voice-cloning APIs.

VIII. Ethics Considerations

Our real-world evaluation was approved by the IRB. All participants provided informed consent via a consent form presented at the start of the study. Participants were anonymized, and no sensitive identifiers or account information were stored or shared during the experiments. Participation was voluntary, and they could withdraw at any time without penalty. The recording process took approximately 5–10 minutes, and each participant received $10 as compensation for their time.

References

[1] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu *et al.*, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, vol. 12, 2016.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[4] "The wall street journal." [Online]. Available: https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402

[5] "Forbes." [Online]. Available: https://www.forbes.com/sites/thomasbrewster/2021/10/14/huge-bank-fraud-uses-deep-fake-voice-tech-to-steal-millions/?sh=2fdf0c417559

[6] A. Gomez-Alanis, A. M. Peinado, J. A. Gonzalez, and A. M. Gomez, "A gated recurrent convolutional neural network for robust spoofing detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 1985–1999, 2019.

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

[8] L. Blue, K. Warren, H. Abdullah, C. Gibson, L. Vargas, J. O'Dell, K. Butler, and P. Traynor, "Who are you (i really wanna know)? detecting audio deepfakes through vocal tract reconstruction," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 2691–2708.

[9] Q. Wang, X. Lin, M. Zhou, Y. Chen, C. Wang, Q. Li, and X. Luo, "Voicepop: A pop noise based anti-spoofing system for voice authentication on smartphones," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2062–2070.

[10] C. Sun, S. Jia, S. Hou, and S. Lyu, "Ai-synthesized voice detection using neural vocoder artifacts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 904–912.

[11] X. Yan, J. Yi, J. Tao, C. Wang, H. Ma, T. Wang, S. Wang, and R. Fu, "An initial investigation for detecting vocoder fingerprints of fake audio," in *Proceedings of the 1st International Workshop on Deepfake Detection for Audio Multimedia*, 2022, pp. 61–68.

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

[13] Y. Xie, H. Cheng, Y. Wang, and L. Ye, "Domain generalization via aggregation and separation for audio deepfake detection," *IEEE Transactions on Information Forensics and Security*, 2023.

[14] K. Zhang, Z. Hua, Y. Zhang, Y. Guo, and T. Xiang, "Robust ai-synthesized speech detection using feature decomposition learning and synthesizer feature augmentation," *IEEE Transactions on Information Forensics and Security*, 2024.

[15] T.-P. Doan, H. Dinh-Xuan, T. Ryu, I. Kim, W. Lee, K. Hong, and S. Jung, "Trident of poseidon: A generalized approach for detecting deepfake voices," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 2222–2235.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *ICML*, vol. 2, no. 3, 2021, p. 4.

[18] J. Mullennix and S. Stern, *Computer Synthesized Speech Technologies: Tools for Aiding Impairment: Tools for Aiding Impairment*. IGI Global, 2010.

[19] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen *et al.*, "Hierarchical generative modeling for controllable speech synthesis," *arXiv preprint arXiv:1810.07217*, 2018.

[20] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. Lopez Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *Advances in neural information processing systems*, vol. 31, 2018.

[21] T.-h. Huang, J.-h. Lin, and H.-y. Lee, "How far are we from robust voice conversion: A survey," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 514–521.

[22] C.-y. Huang, Y. Y. Lin, H.-y. Lee, and L.-s. Lee, "Defending your voice: Adversarial attack on voice conversion," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 552–559.

[23] J. Deng, Y. Chen, Y. Zhong, Q. Miao, X. Gong, and W. Xu, "Catch you and i can: Revealing source voiceprint against voice conversion," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 5163–5180.

[24] H. Chen, W. Jin, Y. Hu, Z. Ning, K. Li, Z. Qin, M. Duan, Y. Xie, D. Liu, and M. Li, "Eavesdropping on black-box mobile devices via audio amplifier's emr," in *Proceedings of the 2018 Annual International Conference on Network and Distributed System Security (NDSS)*, 2024.

[25] T. Du, S. Ji, J. Li, Q. Gu, T. Wang, and R. Beyah, "Sirenattack: Generating adversarial audio for end-to-end acoustic systems," in *Proceedings of the 15th ACM Asia conference on computer and communications security*, 2020, pp. 357–369.

[26] H. Guo, Y. Wang, N. Ivanov, L. Xiao, and Q. Yan, "Specpatch: Human-in-the-loop adversarial audio spectrogram patch attack on speech recognition," in *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*, 2022, pp. 1353–1366.

[27] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.

[28] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in neural information processing systems*, vol. 33, pp. 17 022–17 033, 2020.

[29] K. Kumar, R. Kumar, T. De Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. De Brebisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *Advances in neural information processing systems*, vol. 32, 2019.

[30] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[31] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.

[32] R. Wang, F. Juefei-Xu, M. Luo, Y. Liu, and L. Wang, "Faketagger: Robust safeguards against deepfake dissemination via provenance tracking," in *Proceedings of the 29th ACM international conference on multimedia*, 2021, pp. 3546–3555.

[33] Z. Sha, Z. Li, N. Yu, and Y. Zhang, "De-fake: Detection and attribution of fake images generated by text-to-image generation models," in *Proceedings of the 2023 ACM SIGSAC conference on computer and communications security*, 2023, pp. 3418–3432.

[34] Z. Zhang, Q. Yang, D. Wang, P. Huang, Y. Cao, K. Ye, and J. Hao, "Mitigating unauthorized speech synthesis for voice protection," in *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, 2023, pp. 13–24.

[35] Z. Zhang, D. Wang, Q. Yang, P. Huang, J. Pu, Y. Cao, K. Ye, J. Hao, and Y. Yang, "Safespeech: Robust and universal voice protection against malicious speech synthesis," *arXiv preprint arXiv:2504.09839*, 2025.

[36] Y. Wang, H. Guo, G. Wang, B. Chen, and Q. Yan, "Vsmask: Defending against voice synthesis attack via real-time predictive perturbation," in *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2023, pp. 239–250.

[37] Y. Wen, A. Innuganti, A. B. Ramos, H. Guo, and Q. Yan, "Sok: How robust is audio watermarking in generative ai models?" *arXiv preprint arXiv:2503.19176*, 2025.

[38] Y. Hu, Z. Jiang, M. Guo, and N. Z. Gong, "A transfer attack to image watermarks," *arXiv preprint arXiv:2403.15365*, 2024.
[39] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, "End-to-end anti-spoofing with rawnet2," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6369–6373.
[40] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.
[41] K. Kumari, M. Abbasihafshejani, A. Pegoraro, P. Rieger, K. Arshi, M. Jadliwala, and A.-R. Sadeghi, "Voiceradar: Voice deepfake detection using micro-frequency and compositional analysis," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2025.
[42] E. Wenger, M. Bronckers, C. Cianfarani, J. Cryan, A. Sha, H. Zheng, and B. Y. Zhao, "" hello, it's me": Deep learning-based speech synthesis attacks in the real world," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 235–251.
[43] S. Shaw, B. Nassi, and L. Schönherr, "Generated audio detectors are not robust in real-world conditions," 2024.
[44] F. Li, Y. Chen, H. Liu, Z. Zhao, Y. Yao, and X. Liao, "Vocoder detection of spoofing speech based on gan fingerprints and domain generalization," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 6, pp. 1–20, 2024.
[45] J. Frank and L. Schönherr, "Wavefake: A data set to facilitate audio deepfake detection," *arXiv preprint arXiv:2111.02813*, 2021.
[46] B. E. Koenig and D. S. Lacey, "Forensic authenticity analyses of the header data in re-encoded wma files from small olympus audio recorders," *Journal of the Audio Engineering Society*, vol. 60, no. 4, pp. 255–265, 2012.
[47] S. Borzì, O. Giudice, F. Stanco, and D. Allegra, "Is synthetic voice detection research going into the right direction?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 71–80.
[48] N. V. Kulangareth, J. Kaufman, J. Oreskovic, and Y. Fossat, "Investigation of deepfake voice detection using speech pause patterns: Algorithm development and validation," *JMIR biomedical engineering*, vol. 9, p. e56245, 2024.
[49] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, "Emotions don't lie: An audio-visual deepfake detection method using affective cues," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 2823–2832.
[50] "Voice frequency." [Online]. Available: https://en.wikipedia.org/wiki/Voice_frequency
[51] C.-Z. Yang, J. Ma, S. Wang, and A. W.-C. Liew, "Preventing deepfake attacks on speaker authentication by dynamic lip movement analysis," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1841–1854, 2020.
[52] V. Nguyen, T. F. Yago Vicente, M. Zhao, M. Hoai, and D. Samaras, "Shadow detection with conditional generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4510–4518.
[53] A. V. Oppenheim and J. S. Lim, "The importance of phase in signals," *Proceedings of the IEEE*, vol. 69, no. 5, pp. 529–541, 1981.
[54] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020.
[55] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *International conference on machine learning*. PMLR, 2020, pp. 3247–3258.
[56] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5210–5219.
[57] A. Nautsch, X. Wang, N. Evans, T. H. Kinnunen, V. Vestman, M. Todisco, H. Delgado, M. Sahidullah, J. Yamagishi, and K. A. Lee, "Asvspoof 2019: spoofing countermeasures for the detection of synthesized, converted and replayed speech," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 252–265, 2021.
[58] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.
[59] J. Fridrich, "Digital image forensics," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 26–37, 2009.
[60] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.
[61] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," *arXiv preprint arXiv:2009.00713*, 2020.
[62] X. Liu, X. Wang, M. Sahidullah, J. Patino, H. Delgado, T. Kinnunen, M. Todisco, J. Yamagishi, N. Evans, A. Nautsch *et al.*, "Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2507–2522, 2023.
[63] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "Stc antispoofing systems for the asvspoof2019 challenge," *arXiv preprint arXiv:1904.05576*, 2019.
[64] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 29, pp. 3451–3460, 2021.
[65] "Elevenlab." [Online]. Available: https://elevenlabs.io/
[66] "Speechify." [Online]. Available: https://speechify.com/
[67] "Fishaudio." [Online]. Available: https://fish.audio/
[68] "Playht." [Online]. Available: https://play.ht/
[69] "Dupdub." [Online]. Available: https://www.dupdub.com/
[70] G. Fairbanks, "The rainbow passage," *Voice and articulation drillbook*, vol. 2, pp. 127–127, 1960.
[71] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.
[72] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, vol. 6, p. 15, 2017.
[73] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.
[74] P. Kawa, M. Plata, M. Czuba, P. Szymański, and P. Syga, "Improved deepfake detection using whisper features," *arXiv preprint arXiv:2306.01428*, 2023.
[75] K. Borodin, V. Kudryavtsev, D. Korzh, A. Efimenko, G. Mkrtchian, M. Gorodnichev, and O. Y. Rogov, "Aasist3: Kan-enhanced aasist speech deepfake detection using ssl features and additional regularization for the asvspoof 2024 challenge," *arXiv preprint arXiv:2408.17352*, 2024.
[76] N. Subramani and D. Rao, "Learning efficient representations for fake speech detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5859–5866.

APPENDIX

A. Other Experimental Settings

All experiments are conducted on two NVIDIA RTX 6000 GPUs. Early stopping is applied if the model's performance does not improve for three consecutive epochs. For DCT implementation, we adopt the DCT function from the SciPy library with orthonormal normalization factors. We set the τ_{MIN} and τ_{MAX} to -150 dB and 50 dB as the reference minimum and maximum magnitudes for calculating the sample weight during weighted-scaling training. For the baseline method Dual-stream, we randomly select 10 speed-codec combinations during training for augmentations, with speed from the range of 0.5 to 2.0 times and the codec as (aac, ops, mp3). For Trident [15], we also choose $k = 3$ speech augmentation techniques and $v = 3$ vocoder augmentation

TABLE VII: Detection EER (%) on the LibriSeVoc dataset in the Leave-One-Out scenario.

Method	Unseen vocoder						Average
	DiffWave	MelGAN	PWG	WaveGrad	WaveNet	WaveRNN	
RawNet2	28.8	6.9	17.6	30.1	24.9	40.7	24.8
LFCC-LCNN	6.4	37.9	**1.1**	7.2	20.4	43.3	19.4
Wav2Vec2	5.9	18.9	3.7	6.5	1.8	15.7	8.8
VoiceRadar	18.4	10.5	7.6	4.1	19.3	23.7	13.9
Trident	7.3	28.4	5.3	12.1	2.2	**7.4**	10.5
RawNet2-Voc	40.0	33.4	34.8	17.3	36.6	45.2	34.6
Dual-stream	7.1	8.2	6.8	**3.4**	6.0	15.6	7.9
Ours	**4.1**	**3.4**	3.6	5.8	**1.2**	9.4	**4.6**

methods used in the paper. For real-world experiments with Web voice-cloning APIs, due to the limited sample size, we select an equal number of real speech samples for testing EER, repeat the process 5 times and report the average.

During training, we only use half of the sub-spectrograms for the 2D-input branches and randomly drop one layer for the 3D-input branches to reduce content overlap and prevent Timesformer from overfitting to specific layers. For the 3D branches, we specifically drop the 1st, 3rd, 5th, or 7th layers. After training, we freeze the weights of the four detectors and fine-tune an MLP using the multiple predictions of a speech sample with its ground-truth label. This step incorporates magnitude-layer information to enhance decision-making, akin to position encoding in natural language tasks. Specifically, we generate 8 predicted vectors each for sub-spectrogram and DCT-spectrum sequences using ResNet18, and 4 vectors each for the corresponding 3D representations by dropping specific layers, resulting in 24 (8+8+4+4) vectors per sample. These vectors are then used to fine-tune the MLP, focusing on both the independence and consistency of predictions across different magnitude ranges. During testing, the same process aggregates predictions from the four branches via the MLP to classify samples as real or fake.

B. Additional Experimental Results

Evaluation in the Cross-Method Setting. In Table VII, we present the results for the Leave-One-Out scenario on the LibriSeVoc dataset.

Additional Baselines and Metrics. In addition to the baselines in Table II, we further evaluate three DNN-based detection methods: Whisper [74], ASSIST3 [75], and EfficientCNN [76], following the main setups described in their papers. We consider the first two combinations of the leave-most-out setting on the WaveFake dataset for evaluation. Table VIII presents the results under this setting, showing a similar phenomenon in which unseen vocoders are difficult to handle for baseline methods. To examine whether our method is effective on both positive and negative samples, we evaluate the TPR, TNR, and F1 scores under the first combination (HifiGAN+PWG) in the leave-most-out setting of WaveFake. The corresponding TPR, TNR, and F1 scores are 0.97, 0.95, and 0.98, respectively, demonstrating that our method effectively distinguishes real and fake speech samples with a high detection rate. In addition, we evaluate the detection efficiency. The overhead of the pipeline is 0.23s for a 4s speech clip.

TABLE VIII: Detection EER (%) on the WaveFake dataset in the leave-most-out scenario.

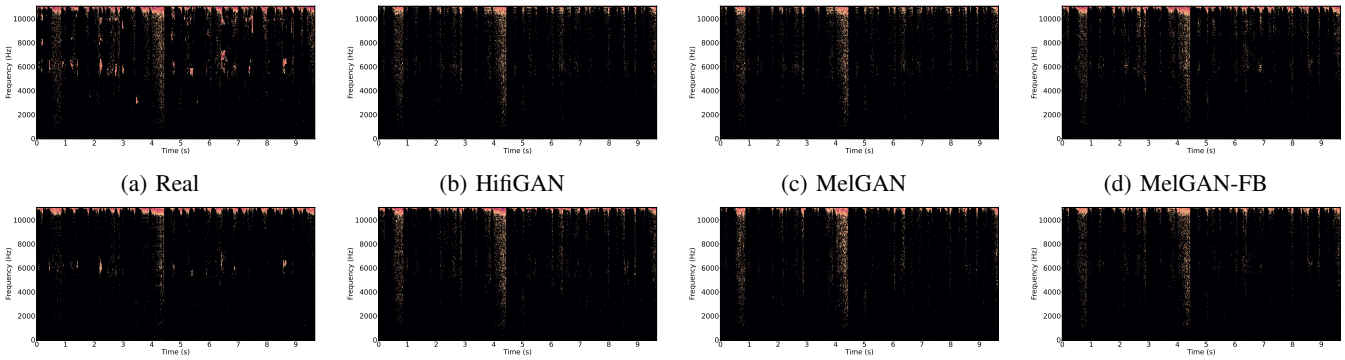
	HifiGAN+PWG	MelGAN+WaveGlow
Whisper (SpecRNet)	21.3	32.7
Whisper (LCNN)	32.0	12.3
AASIST3	18.7	8.2
EfficientCNN	13.0	10.4
Ours	**3.9**	**1.4**

TABLE IX: Detection EER (%) across varying numbers of sub-spectrogram layers on the LibriSeVoc dataset.

	# of layers			
	4	6	8	10
Inner-dataset	1.8	0.7	0.5	0.9
Cross-method	8.6	6.5	4.6	4.2

Impact of the Number of Sub-Spectrogram Layers. In Section VI, we adopt a setting where the spectrogram is partitioned into 8 sub-spectrogram layers. To evaluate the impact of the number of layers on detection performance, we analyze four cases with layer counts of 4, 6, 8, and 10. The results of this analysis are presented in Table IX. From this table, we observe that using a smaller number of layers leads to a noticeable performance decline in both tasks. This is likely due to the reduced granularity in the magnitude domain, where larger and smaller magnitudes become excessively blended, limiting the model's ability to identify fine-grained patterns. Conversely, increasing the number of layers excessively can result in overly sparse and redundant inputs, which may not effectively contribute to DNN training and could negatively impact model performance.

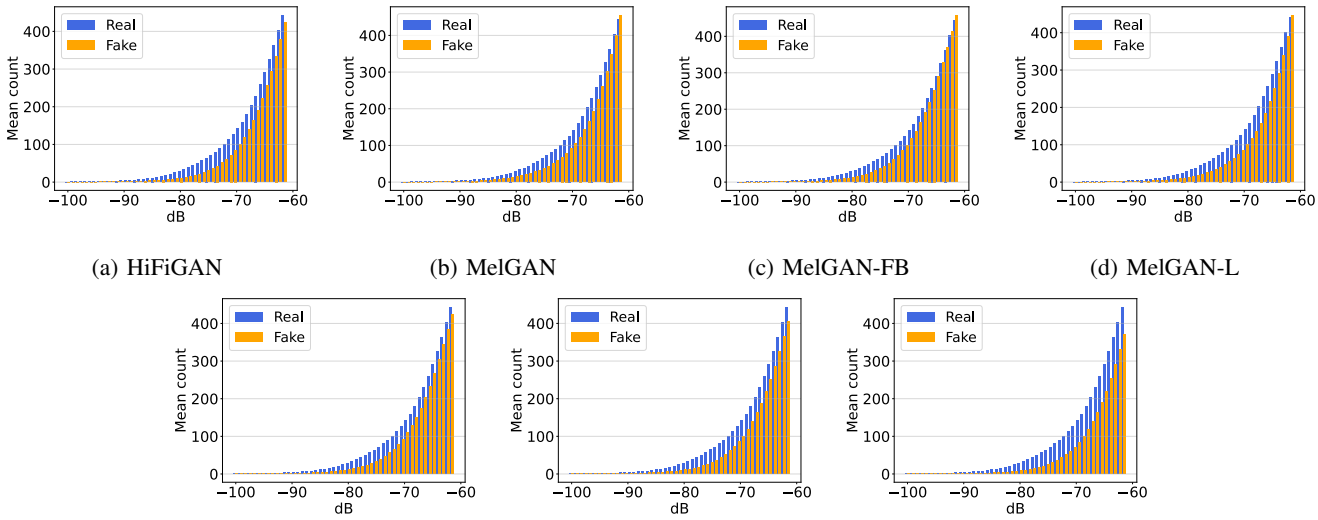
Impact of Sample Length. While standard speech samples often contain rich content, shorter clips, such as voice commands in IoT applications, are also common. We evaluate whether our detection framework can handle these short clips. Since the magnitude domain captures the "depth" of speech signals, it remains informative regardless of sample length. We



(a) Real (b) HifiGAN (c) MelGAN (d) MelGAN-FB

(e) MelGAN-L (f) MelGAN-MB (g) PWG (h) WaveGlow

Fig. 9: Small-magnitude sub-spectrograms generated with different vocoders in the WaveFake dataset.



(a) HiFiGAN (b) MelGAN (c) MelGAN-FB (d) MelGAN-L

(e) MelGAN-MB (f) PWG (g) WaveGlow

Fig. 10: Distribution of time-frequency point counts in small-magnitude region across all vocoders in the WaveFake dataset.

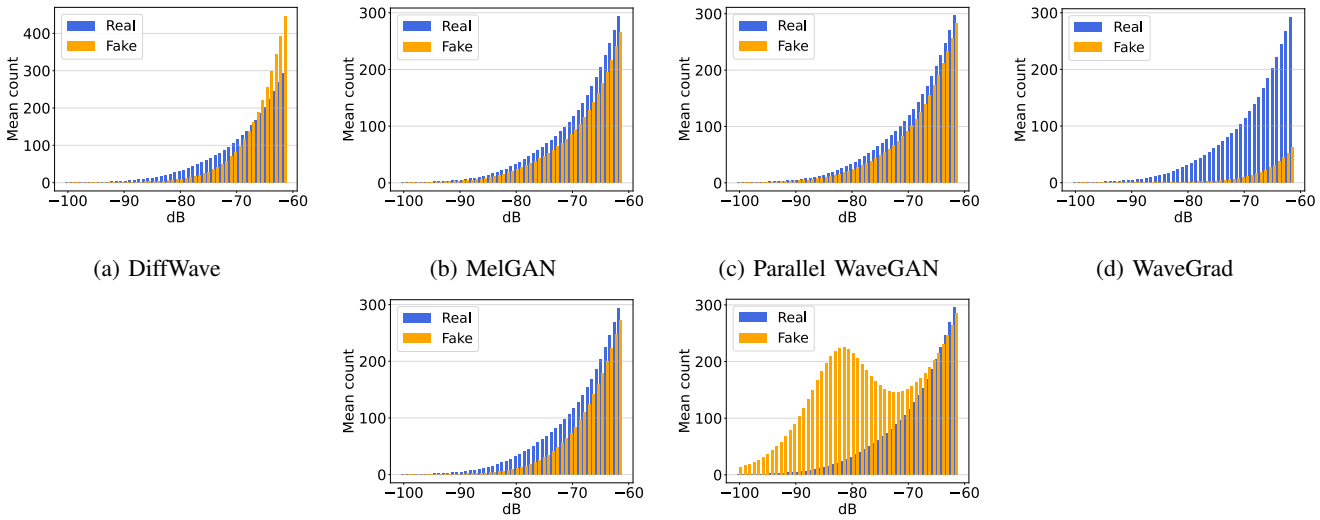
test clips of 2s and 3s under the Leave-One-Out setting. The average EER(%) is 1.4 for 2s and 0.8 for 3s, confirming our method's robustness across varying sample lengths.

C. More Statistics for Fake Speech Analysis

Impact of Vocoder. In Figure 9, we visualize the small-magnitude spectrograms of real and fake speech generated by seven different vocoders using the same sample from the WaveFake dataset. The results demonstrate the phenomenon of missing details and energy is consistent across different vocoders.

Figures 10 and 11 show the average count of time-frequency points within the small-magnitude range (below -60 dB) for different vocoders in the WaveFake and LibriSeVoc datasets, respectively. In the WaveFake dataset, real samples consistently exhibit higher counts of time-frequency points compared to fake samples, indicating that real audio retains more fine-grained details in the small-magnitude region. In the LibriSeVoc dataset, while real samples generally display higher counts in the small-magnitude range, an exception is observed with WaveRNN, where fake samples show increased counts in specific magnitude ranges. Given the rapid evolution of speech technologies, we also extend our evaluation beyond the 13 vocoders (e.g., WaveNet (2016) and HiFi-GAN (2020)) to include recent black-box Web voice-cloning APIs: ElevenLabs [65], Speechify [66], FishAudio [67], PlayHT [68], and DupDub [69]. Our analysis reveals that the mean counts of time-frequency points in the small, middle, and large magnitude ranges are 2308/1325, 25010/26689, and 8610/8386 for real/fake speech samples, respectively. These results suggest that even the latest models still exhibit notable artifacts in the small-magnitude range.

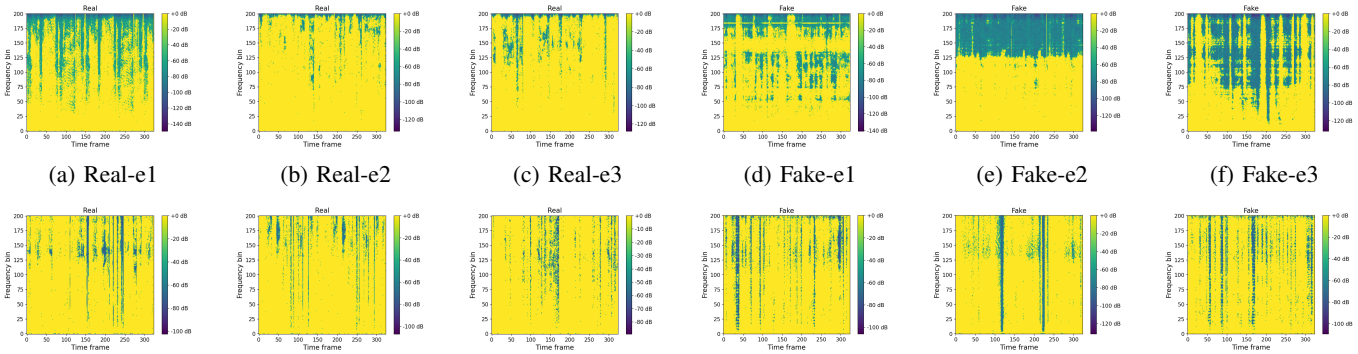
Examples from More Datasets. In Figure 12, we visualize both fake and real speech samples from two additional widely studied datasets: In-the-Wild [12] and ASVSpoof2019 [57], which provide a diverse basis for analysis. For each dataset, we provide three examples of real speech and three examples of fake speech. From Figure 12, we can observe several



(a) DiffWave (b) MelGAN (c) Parallel WaveGAN (d) WaveGrad

(e) WaveNet (f) WaveRNN

Fig. 11: Distribution of time-frequency point counts in small-magnitude region over all vocoders in the LibriSeVoc dataset.



(a) Real-e1 (b) Real-e2 (c) Real-e3 (d) Fake-e1 (e) Fake-e2 (f) Fake-e3

(g) Real-e1 (h) Real-e2 (i) Real-e3 (j) Fake-e1 (k) Fake-e2 (l) Fake-e3

Fig. 12: Examples of small-magnitude spectrograms from In-the-Wild and ASVSpoof2019. (a)–(c): real samples from In-the-Wild; (d)–(f): fake samples from In-the-Wild; (g)–(i): real samples from ASVSpoof2019; (j)–(l): fake samples from ASVSpoof2019.

characteristics of fake speech, while real speech tends to exhibit more consistent and natural patterns. In the In-the-Wild dataset, many fake speech samples display unnatural vertical or horizontal stripes or overly concentrated energy at higher frequencies. In the ASVSpoof2019 dataset, Example (k) exhibits abnormal vertical energy bars. However, some fake speech samples, such as Examples (j) and (l), also retain texture details, making them more challenging to distinguish directly within the spatial domain of sub-spectrograms. Nevertheless, upon closer inspection, Examples (j) and (l) reveal repetitive or monotonously consistent texture patterns. This observation underscores potential artifacts at the distribution level and highlights the need for more advanced strategies to detect traces of fabrication hidden within spectrograms.