FAKEMARK: DEEPFAKE SPEECH ATTRIBUTION WITH WATERMARKED ARTIFACTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deepfake speech attribution remains challenging for existing solutions. Classifier-based solutions often fail to generalize to domain-shifted samples, and watermarking-based solutions are easily compromised by distortions like codec compression or malicious removal attacks. To address these issues, we propose FakeMark, a novel watermarking framework that injects artifact-correlated watermarks associated with deepfake systems rather than predefined bitstring messages. This design allows a detector to attribute the source system by leveraging both injected watermark and intrinsic deepfake artifacts, remaining effective even if one of these cues is elusive or removed. Experimental results show that FakeMark improves generalization to cross-dataset samples where classifier-based solutions struggle and maintains high accuracy under various distortions where conventional watermarking-based solutions fail. Speech samples are available at `https://fakemark-demo.github.io/fakemark-demo/`.[1]

1 INTRODUCTION

Attributing deepfake speech requires identifying the source system used to generate the synthetic samples (Müller et al., 2022; Klein et al., 2025). This is critical for mitigating risks such as copyright violations and malicious use of speech synthesis systems. Most solutions train deep-neural-network-based classifiers (illustrated in the top panel of Figure 1) for system identification (Sun et al., 2023; Wang et al., 2025). They often require to be trained in a discriminative manner with data generated by a rich variety of speech synthesis systems to capture artifact differences. However, such classifiers are known to be sensitive to domain shift and struggle to detect deepfakes generated by unseen systems (Bhagtani et al., 2024; Chen et al., 2025c), as their performance is fundamentally constrained by the variability present in the training data.

Recently, watermarking-based methods become popular as an alternative solution to the attribution task (Cho et al., 2022; Li et al., 2025b; Yang et al., 2025). These solutions involve training a pair of watermark generator and detector (illustrated in the middle panel of Figure 1), where the generator injects an watermark message into the carrier speech that is later extracted by the detector; attribution is achieved by mapping the extracted message to its pre-assigned system label. Although watermarking-based solutions have demonstrated high accuracy on various benchmarks (Liu et al., 2024b; Roman et al., 2024), they can be easily compromised by common distortions and removal attacks (Yang et al., 2024; Kassis & Hengartner, 2025; Yao et al., 2025). In its application to speech, for example, generators are trained to inject watermarks that are inaudible to the human ear. Yet watermark detectors often struggle under neural codec transmissions (Juvela & Wang, 2025; O'Reilly et al., 2025; Özer et al., 2025), whose training objective is compression and high-fidelity reconstruction of audio signals (Défossez et al., 2023; Ju et al., 2024). In deepfake related tasks such as detection (Wu et al., 2025), classifier-based solutions remain robust under neural codecs since deepfake artifacts are preserved to some extent, whereas watermark detectors degrade to near-chance performance as the injected messages are removed during compression.

Presented in this work is our attempt to address the above challenges for robust deepfake speech attribution. Specifically, we ask the following research question: **Can we enhance deepfake traceability by injecting artifact-correlated watermarks**? We hypothesize that correlating watermarks

[1] Codes and model checkpoints will be released upon acceptance.

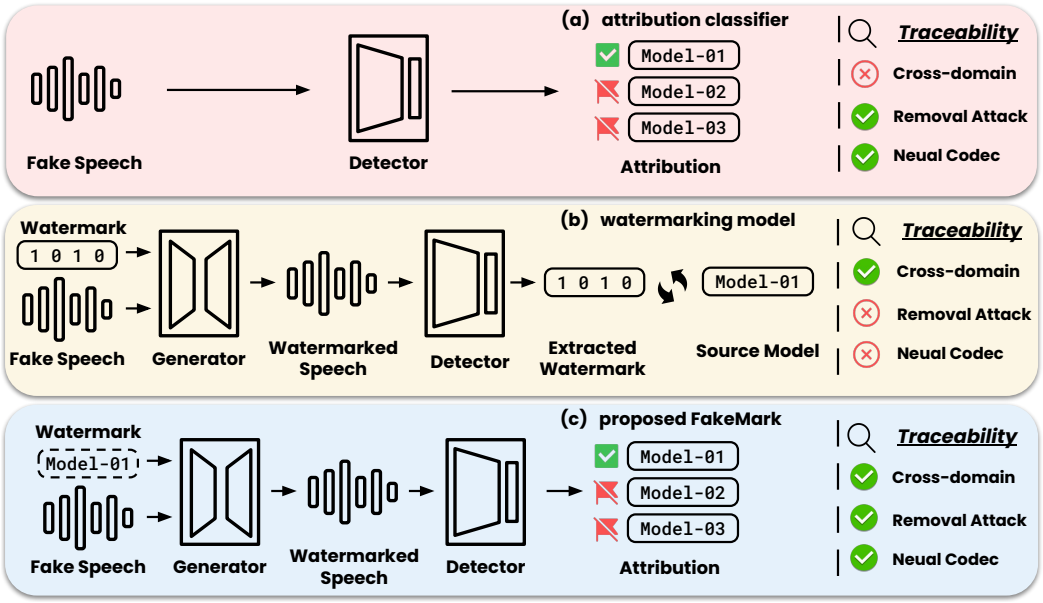

Figure 1: Illustrations of deepfake attribution by (a) classifier-based model, (b) watermarking-based model, and (c) our proposed FakeMark.

with deepfake artifacts can provide improved 1) **robustness towards distortions** by enabling the detector to perform attribution through acoustic artifacts when the watermark message is removed; and 2) **generalization performance** by introducing a watermark generator to assist the typically standalone classifier-based detector when seen artifact patterns are absent. To answer the question, we propose a novel watermarking framework (illustrated in the bottom panel of Figure 1) and evaluate its performance under diverse conditions. Our main contributions are summarized as follows:

- We introduce FakeMark, a novel watermarking framework for deepfake speech attribution. The FakeMark generator injects watermarks that are correlated with acoustic artifacts, allowing the detector to map either the artifacts or the watermarks to their source system.
- We present the first systematic evaluation of deepfake speech attribution using both watermarking- and classifier-based models. We evaluate FakeMark against these baselines on common datasets and under diverse distortions, showing that it improves attribution robustness and generalization in challenging scenarios.

2 Related Works

Speech generation typically follows two paths: text-to-speech (TTS) and voice conversion (VC). In modern TTS, an acoustic model maps the input text (or its derived linguistic features) to an intermediate acoustic representation that is either continuous valued hidden feature vectors or discrete tokens. A neural vocoder is then used to synthesize the speech waveform (Tan et al., 2021). VC follows a partially similar design: it takes an input waveform from a source speaker and renders the same content in a target speaker's voice. The term *artifacts* denotes deviations of synthesized speech from natural speech. Common audible artifacts include (i) alignment errors between text and predicted acoustics that cause word skipping or repetition (Zen et al., 2009); (ii) insufficient modeling of prosody (e.g., incorrect pitch accent (Łańcucki, 2021)), expressiveness (e.g., flat intonation (Liu et al., 2021; Mahapatra et al., 2025)) , and speaker characteristics (e.g., a voice perceptually dissimilar to the target speaker (Chen et al., 2025a; Pan et al., 2022)); and (iii) vocoder artifacts such as buzziness or high-frequency noise (Bak et al., 2023; Sun et al., 2023).

Deepfake attribution. Depending on the specific architectures used, it has been reported that different acoustic models (Bhagtani et al., 2024; Chen et al., 2025b) and vocoders (Sun et al., 2023; Deng et al., 2024) leave distinctive artifacts that can be leveraged for deepfake attribution. Solutions to

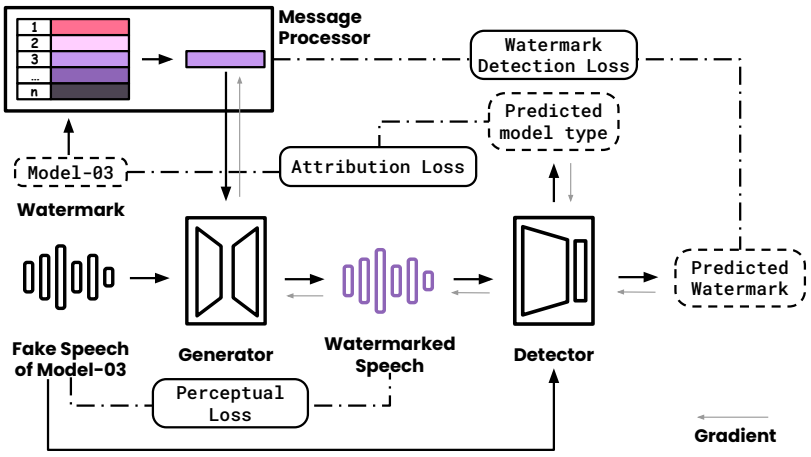

Figure 2: Training pipeline of FakeMark.

the task naturally involve collecting samples from various TTS and VC systems (Müller et al., 2024; Chen et al., 2025c). These samples are then used to train multiclass classifiers to predict the acoustic models or vocoders used for their generation (Klein et al., 2024). However, such supervised training scheme can sometimes cause classifiers to exploit undesired differences in the training data, leading to poor generalization performance on unseen samples. This includes samples generated by seen systems but trained with different languages (Marek et al., 2024) or speakers (Klein et al., 2025), or samples with subtle artifacts like unnatural silences (Chen et al., 2025b) or even generated by the same system with different weights (Stan et al., 2025). Alternative strategies to achieve generalization include estimating model confidence on unseen samples (Klein et al., 2025) or measuring sample similarities in latent space, akin to verification tasks (Negroni et al., 2025).

Speech watermarking models are designed to inject and extract bitstring messages within a speech signal (Li et al., 2025b; Liu et al., 2024b). Depending on the information carried in the message, these models are versatile for various tasks. For example, a watermark can encode a compressed version of the original signal for self-recovery (Quiñonez-Carbajal et al., 2024), or it can be assigned to different users for attribution and copyright protection (Roman et al., 2024; Liu et al., 2024a). In deepfake-related applications, different bitstrings can be assigned to real and fake samples for detection (Wu et al., 2025; Roman et al., 2024) or to counter malicious deepfake manipulations (Li et al., 2025a; He et al., 2025). Beyond bitstring messages, the presence of a watermark itself can represent a zero-bit message indicating whether a sample is real or fake (Juvela & Wang, 2024; Roman et al., 2025). Previous studies have reported that watermarking-based models are vulnerable to distortions such as neural codecs, malicious forgery, or removal attacks (Yang et al., 2024; Liu et al., 2024b). Common strategies to enhance robustness include using codec-based data augmentation during training (Juvela & Wang, 2025) and injecting watermarks into deep latent representations of the speech signal (Ji et al., 2025b).

3 FakeMark

We describe FakeMark pipelines for watermark generation and detection in Sec. 3.1. Objectives used to train the system modules are detailed in Sec. 3.2.

3.1 Pipeline

As illustrated in Figure 2, FakeMark takes two inputs during watermark generation: the speech signal $\boldsymbol{s} \in \mathbb{R}^T$ and the watermark message $w \in \{1, \ldots, C\}$, where T is the number of waveform sampling points and C is the total number of deepfake systems. It outputs a watermarked signal

$\boldsymbol{s}_w \in \mathbb{R}^T$ that carry the watermark message and has the same dimensionality as the input signal. During watermark detection, FakeMark takes $\boldsymbol{s}_w$ and predicts a watermark message w'.

The generation process involves four stages:

1. Given the input watermark message w, the message processor returns the corresponding embedding vector $\boldsymbol{e}_w \in \mathbb{R}^H$ to the generator, where H is the latent dimension.
2. Given the input signal $\boldsymbol{s}$, the generator down-samples it and extracts a compact latent representation $\boldsymbol{H}_s \in \mathbb{R}^{\lfloor \frac{T}{\alpha} \rfloor \times H}$, where α is the downsampling factor.
3. Given the input watermark embedding $\boldsymbol{e}_w$, the generator repeats it along the time axis to form $\boldsymbol{E}_w \in \mathbb{R}^{\lfloor \frac{T}{\alpha} \rfloor \times H}$, then applies voice-activity detection (VAD) to obtain a binary mask $\boldsymbol{m} \in \{0,1\}^{\lfloor \frac{T}{\alpha} \rfloor}$ indicating speech-active frames, and computes the watermark latent $\boldsymbol{H}_w = \boldsymbol{m} \odot \boldsymbol{E}_w$, where $\boldsymbol{H}_w \in \mathbb{R}^{\lfloor \frac{T}{\alpha} \rfloor \times H}$.
4. The generator up-samples $\boldsymbol{H}_s + \boldsymbol{H}_w$ and outputs the watermark signal $\boldsymbol{\delta}_w \in \mathbb{R}^T$. The final watermarked signal is obtained as $\boldsymbol{s}_w = \boldsymbol{s} + \boldsymbol{\delta}_w$, where $\boldsymbol{s}_w \in \mathbb{R}^T$.

Following the generation pipeline, we explore two generator architectures:

- FakeMark^A: follows an encoder-decoder architecture that processes speech waveforms, as used in AudioSeal (Roman et al., 2024);
- FakeMark^T: follows an encoder architecture that processes spectrogram features, as used in Timbre (Liu et al., 2024a). In this setting, $\boldsymbol{H}_s$ is the linear-scale spectrogram obtained via Short-Time Fourier Transform (STFT). Final waveforms are obtained via inverse STFT.

The detection process involves two stages:

1. During training, given the input watermarked signal $\boldsymbol{s}_w$, the detector applies a series of transformations to obtain a distorted version input $\boldsymbol{s}'_w$. This strategy ensures robustness of the watermark injection and detection. Full list of the used transformations and their settings are provided in Appendix A.2. These transformations are disabled during inference.
2. The detector extracts sequence-level feature from the input waveform and predicts the class probabilities $\mathbf{p} \in [0,1]^C$ over C watermark types; the extracted watermark is obtained as $w' = \arg\max_{i \in \{1,\ldots,C\}} p_i$.

We use a common detector architecture that consists of a pre-trained SSL front-end (Pratap et al., 2024) and a fully connected back-end classifier. Detailed architectures are provided in the Appendix A.3.

3.2 Training objectives

All FakeMark modules are optimized with three classes of objectives: 1) attribution loss, to ensure the ability of distinguishing different types of artifacts; 2) detection loss, to maximize the successful injection and detection of watermarks; and 3) perceptual loss, to minimize the perceptual distortion between original and watermarked signals. They are detailed below.

Attribution loss differentiates FakeMark from conventional watermarking approaches. It is computed as the cross-entropy between the ground-truth deepfake system label and the detector's predicted class probabilities over an *unwatermarked* clean signal. This objective is similar to training classifier-based attribution models (Klein et al., 2024; Sun et al., 2023), where the goal is to capture distinct characteristics of deepfakes generated by different systems. The back-propagated attribution loss encourages the detector to distinguish various types of deepfake artifacts and implicitly guides the watermark embeddings from the message processor to correlate with these artifacts. As a result, each watermark embedding encodes the artifact patterns learned from all samples of a specific deepfake system in the training set.

Watermark detection loss ensures that the generator injects watermark messages that can be reliably recognized by the detector. Similar to the training of conventional watermarking models, a

random watermark message is sampled and processed to obtain a watermark embedding, which is then used for watermarked signal generation. The detector predicts class probabilities from this *watermarked* signal, and the watermark detection loss is computed as the cross-entropy between the ground-truth watermark message and the predicted distribution.

By jointly training FakeMark with both attribution and watermark detection losses, the message processor learns to align watermark embeddings with the deepfake artifacts they represent, and the generator-detector pair learns to robustly inject and detect these watermarks. During inference, with the watermark message always chosen to match the ground-truth deepfake system, FakeMark can attribute the source system using both acoustic artifacts and the watermarks. This ensures effective attribution even if one of these cues is compromised.

Perceptual losses promote the imperceptibility of watermarks and the naturalness of watermarked signals. This is achieved by refining the watermarked signal with HiFi-GAN–style losses (Kong et al., 2020), which include a Mel-spectrogram reconstruction loss to enforce the spectral similarity and adversarial discriminator losses to improve speech fidelity.

Additionally, we follow Roman et al. (2024) to refine the watermark signals generated by both generator architectures. We apply l_1 loss and loudness on the watermark signal to decrease its intensity and ensure its robustness towards distortions. We further use a frequency magnitude loss to align the averaged spectral envelope of the watermark signal with that of the clean signal, promoting perceptual similarity and ensuring the watermarks remain less audible.

4 Experiments and Results

We perform in-domain evaluation with seen artifacts and cross-dataset evaluation with unseen artifacts. FakeMark is compared against recent baselines on both clean and distorted signals. In addition to attribution accuracy, we also assess the speech quality of watermarked signals.

4.1 Experimental setup

Datasets. We use the MLAAD_v5 dataset (Müller et al., 2024) for training and evaluation. Following the source tracing challenge protocol (Müller, 2024), our training set comprises 24 TTS systems covering eight languages. To mitigate the influence of undesirable artifacts related to language or speaker (Klein et al., 2024), we group systems with identical architectures into a single class. The resulting training set contains 12 classes, 9 of which appear in the evaluation set. For cross-dataset evaluation, we collected samples generated by five of these systems from ASVspoof5 (Wang et al., 2024) and TIMIT-TTS (Salvi et al., 2023) datasets. Both evaluation sets were randomly sampled with an equal number of files per class. All evaluated system architectures are seen during training; evaluations on unseen architectures are beyond the scope of this work. Dataset details are provided in Appendix A.4.

Baseline systems. We compare our FakeMark with recent and remarkable watermarking models: AudioSeal (Roman et al., 2024) and Timbre (Liu et al., 2024a), and a ResNet34-based classifier (Klein et al., 2025) that takes STFT spectrograms as input. Additionally, we train an SSL-based classifier with the same architecture as FakeMark detector (denoted MMS-300M) to isolate the impact of the watermarking scheme. All baselines were trained with the same training set and augmentation strategy as FakeMark. Watermark message length for AudioSeal and Timbre is set to 4 (equivalent to 2^4 unique bitstrings)-the minimum capacity required to cover 12 classes. Full model configurations and implementation details are in Appendix A.5.

Evaluation metrics. Objective evaluation for both attribution and audio quality are performed to evaluate FakeMark and baselines:

- For attribution performance, we report accuracy result. Predicted class for AudioSeal and Timbre is the class whose assigned 4-bit message has the shortest Hamming distance to the detector output (Roman et al., 2024; Liu et al., 2025). For FakeMark and classifier-based models, predicted class is the class with the highest detector probability.
- For audio quality assignment of watermarked signals, we use four different metrics: Scale Invariant Signal to Noise Ratio (SI-SNR) for evaluating noise-level of watermarks; PESQ

to evaluate speech quality for telecom-like scenarios (Rix et al., 2001); ViSQOL for assessing perceptual quality for network-based scenarios (Hines et al., 2012); Production Quality (PQ) to estimate the clarity and fidelity of watermarked signals (Tjandra et al., 2025). Unlike the previous three metrics, PQ does not require clean reference signals.

Distortions and attacks. To evaluate system robustness against distortions and watermark removal attacks, we apply a set of transformations previously shown to have a noticeable impact on either watermark extraction (O'Reilly et al., 2025; Yao et al., 2025; Yang et al., 2024) or deepfake detection (Wu et al., 2025) in the literature. They are applied to the watermarked signals for FakeMark, AudioSeal, and Timbre, and directly to the input signals for ResNet34 and MMS-300M, include:

- Signal processing-based transforms: Pitch shift, playback speed change, and additive noise from MUSAN (Snyder et al., 2015);
- Neural Codec-based waveform compression and regeneration: SpeechTokenizer (Zhang et al., 2024a), FACodec (Ju et al., 2024; Zhang et al., 2024b), and WavToenizer (Ji et al., 2025a);
- Neural Vocoder-based waveform regeneration: HiFi-GAN (Kong et al., 2020), Vocos (Siuzdak, 2024), and BigVGAN (Lee et al., 2023).
- Black-box watermark removal attacks: Overwriting (Yao et al., 2025), where publicly-available, pre-trained Timbre and AudioSeal models are run sequentially to overwrite existing watermarks; and Averaging (Yang et al., 2024), where an average watermark is computed using a pre-trained AudioSeal model and then subtracted from the watermarked signals.

Details of distortions and attacks can be found in Appendix A.6.

4.2 Results

We report deepfake attribution performance in this section, including in-domain evaluation results in Sec. 4.2.1 and cross-dataset evaluation results in Sec. 4.2.2. Speech quality evaluation and additional analysis are reported in Sec. 4.2.2 and Sec. 4.2.4.

4.2.1 Evaluation with seen artifacts

Table 1 presents attribution accuracy for FakeMark and baselines on the MLAAD_v5 test set. Rows represent accuracies under different distortions. Cells are color-coded in grayscale by row: darker shades indicating lower accuracy and lighter shades indicating higher accuracy. We summarize observations related to our research question below.

Table 1: Attribution accuracy results on seen artifacts across distortions and attacks.

	System / Distortion	Proposed Method		Watermarking Baselines		Classifier Baselines	
		FakeMark^A	**FakeMark^T**	**AudioSeal**	**Timbre**	**MMS-300M**	**ResNet34**
	None	1.00	1.00	1.00	1.00	1.00	0.97
Signal Processing	Pitch	0.82	1.00	0.80	0.96	0.27	0.88
	Speed	1.00	1.00	0.85	0.97	1.00	0.92
	Noise	0.63	0.71	0.72	0.60	0.80	0.50
Codec	SpeechTokenizer	0.85	0.99	0.10	0.94	0.92	0.88
	FACodec	0.91	0.99	0.17	0.82	0.92	0.79
	WavTokenizer	0.33	0.34	0.09	0.19	0.39	0.71
Vocoder	HiFi-GAN	0.91	1.00	0.09	1.00	0.94	0.92
	Vocos	0.98	1.00	0.12	1.00	0.98	0.97
	BigVGAN	0.99	1.00	0.28	1.00	1.00	0.97
Removal Attack	Overwriting	0.99	0.95	0.68	0.55	0.95	0.75
	Averaging	0.98	0.99	0.79	1.00	1.00	0.96

FakeMark is robust to strong watermark removal distortions. When no distortion is applied, all models achieve near-perfect accuracy (above 0.97). Across most distortions—except background noise and WavTokenizer—both FakeMark variants maintain high attribution accuracy (above

0.80). In contrast, AudioSeal accuracy drops dramatically under codec (0.09–0.17) and vocoder (0.09–0.28) reconstructions, which is expected given that these distortions are known strong watermark removers (O'Reilly et al., 2025; Juvela & Wang, 2025). Although FakeMark^A shares the same generator architecture as AudioSeal, its detector can still leverage deepfake artifacts for attribution, yielding performance that is similar to that of the MMS-300M classifier.

Our baseline Timbre model demonstrates unexpectedly robust performance across distortions previously reported as vulnerabilities (O'Reilly et al., 2025; Özer et al., 2025). This is likely due to retraining with additional augmentations, including a codec method named EnCodec (Défossez et al., 2023).

FakeMark is robust to watermark removal attacks. Though Timbre performance was enhanced with additional augmentations, both watermarking baselines remain vulnerable to removal attacks: Timbre drops from 1.00 to 0.55 under overwriting, and AudioSeal drops to 0.79 under averaging. Both FakeMark variants are less affected, with the lowest accuracy being 0.95—substantially more robust than the other watermarking baselines.

Additional discussion: models process spectrogram features are generally more robust. Table 1 shows that attribution is easily solved under clean conditions. Even with distortions, most models—except AudioSeal—maintain reliable performance in many scenarios. We also notice that models process spectrogram features are more robust to distortions compared to their counterparts. Watermarking models such as FakeMark^T and Timbre achieve perfect accuracies (1.0) under neural vocoders. The ResNet34 is the only solution that does not reach perfect performance under clean conditions (0.97); however, its lowest accuracy (0.50 under Noise) remains noticeably higher than the MMS-300M's lowest results (0.27 under Pitch shift and 0.39 under WavTokenizer).

Almost all tested models appear sensitive to signal processing–based distortions but relatively more robust to other types of distortions. This is expected, as signal processing transforms directly modify the speech signal and thus alter artifact patterns. By contrast, reconstruction-based distortions primarily regenerate the waveform together with artifacts and, in some cases, watermarks. In the next section, we show that attribution becomes more difficult when the artifact patterns are unseen, particularly for the two classifier-based baselines.

4.2.2 Evaluation with unseen artifacts

Table 2 present cross-dataset evaluation of attribution accuracy of FakeMark and baseline models. Results are presented in a similar format as Table 1. We summarize observations related to our research question below.

Table 2: Attribution accuracy results on unseen artifacts across distortions and attacks.

	System / Distortion	Proposed Method		Watermarking Baselines		Classifier Baselines	
		FakeMark^A	**FakeMark^T**	**AudioSeal**	**Timbre**	**MMS-300M**	**ResNet34**
	None	1.00	1.00	1.00	1.00	0.07	0.12
Signal Processing	Pitch	0.80	1.00	0.72	0.96	0.00	0.10
	Speed	0.99	1.00	0.78	0.98	0.06	0.11
	Noise	0.58	0.63	0.65	0.62	0.03	0.05
Codec	SpeechTokenizer	0.58	0.88	0.07	0.90	0.07	0.10
	FACodec	0.87	0.88	0.08	0.85	0.08	0.05
	WavTokenizer	0.06	0.11	0.03	0.21	0.07	0.07
Vocoder	HiFi-GAN	0.88	0.98	0.08	1.00	0.07	0.11
	Vocos	0.98	1.00	0.09	1.00	0.03	0.11
	BigVGAN	1.00	1.00	0.19	1.00	0.06	0.11
Removal Attack	Overwriting	0.97	0.77	0.70	0.54	0.03	0.05
	Averaging	0.99	1.00	0.73	1.00	0.06	0.10

FakeMark performs robustly under domain shift. Under clean conditions, both FakeMark variants and the watermarking baselines achieve perfect accuracy (1.0), this is consistent with their in-domain results in Table 1. The two classifier-based models perform poorly (0.07 for MMS and 0.12 for ResNet34), likely due to their limited generalization to unseen artifact patterns—even those produced by TTS architectures seen during training. The two classifiers give similar performance

under distortions and attacks, not because their robustness improves in these conditions, but rather because their clean-condition accuracy is already very low.

As in Table 2, the performance of FakeMark and watermarking baselines degrades when input signals are distorted, but the trends remain similar to those in Table 1, with a slight drop in overall accuracy. Given that MMS-300M fails on this domain-shifted data (highest accuracy 0.12), the robustness of FakeMark detector (above 0.80 under most distortions) can be attributed primarily to the watermarks injected by its generator. Unlike classifier-based models, FakeMark detector is influenced more by distortions applied to the carrier signal than by the carrier itself.

Watermarks injected by FakeMark are robust to removal attacks. From the last two rows of Table 1, we may tentatively hypothesize that FakeMark's robustness against watermark removal attacks was due to the persistence of artifacts. However, results from Table 2 show, when such artifacts are absent in cross-dataset evaluation, both FakeMark variants remain the most robust among watermarking models (lowest accuracy 0.77 under Overwriting, compared to 0.70 for AudioSeal and 0.54 for Timbre). Hence, this robustness is likely to stem from the injection and detection of watermark message, which is designed to correlate with acoustic artifacts. In contrast, removal attacks primarily focus on removing or overwriting fixed patterns in the carrier signal (Yang et al., 2024).

Additional discussion on watermarking in deepfake attribution. Both FakeMark variants and Timbre outperform the two classifiers in nearly all test cases across in-domain (Table 1) and cross-dataset (Table 2) evaluations. Beyond the robustness provided by the system design and training strategies, it is important to note that these solutions are designed for different application scenarios. Classifier-based solutions are passive and require no prior knowledge of the input signal, whereas watermarking-based solutions are proactive and require a message to be injected into the detector input in advance. In the following sections, we further assess the impact of the injected messages on speech quality (Sec. 4.2.3) and detector performance (Sec. 4.2.4).

4.2.3 Evaluation on speech quality and intelligibility

We evaluate the quality and intelligibility of watermarked signals. Results are presented in Table 3. Our observations are summarized below.

Table 3: Comparison of speech quality and intelligibility on watermarked speech signals generated by FakeMark and watermarking-based baselines.

	System	SI-SNR ↑	PESQ ↑	ViSQOL ↑	PQ ↑
Baselines	AudioSeal	36.49	4.55	4.98	6.78
	Timbre	21.79	2.97	4.20	5.67
Proposed	FakeMark^A	35.34	3.79	4.81	6.62
	FakeMark^T	14.97	2.83	4.41	6.18

FakeMark^A achieves second in speech quality. The FakeMark^A performs second only to AudioSeal. Its relatively high SI-SNR (35.34 dB) suggests that the injected watermark has low energy compared to the clean carrier. For other speech quality and fidelity metrics, AudioSeal is the only system achieving a PESQ score above 4 (4.55), while FakeMark^A is slightly lower in ViSQOL (4.98 vs. 4.81) and PQ (6.78 vs. 6.62).

Trade-off between robustness and speech quality. We observe that watermarks injected through spectrogram features (Timbre and FakeMark^T) introduce more distortions to the carrier speech than the approaches that directly process waveforms (AudioSeal and FakeMark^A). Their worse speech quality contrasts with our observations on attribution performance in Sec. 4.2.1, and suggests a trade-off between attribution robustness against distortions and speech quality. The consistent near-perfect performance of Timbre and FakeMark^T is achieved through stronger, more perceptually noticeable watermarks that can survive multiple distortions (shown in Figures 4 and 6 in Appendix A.7). In contrast, AudioSeal's less perceptible watermark introduces minimal distortion to the carrier but is the most vulnerable among the evaluated models. Our proposed FakeMark^A provides strong watermark injection while maintaining relatively high speech quality.

4.2.4 IMPACT OF WATERMARKS ON ATTRIBUTION

In this section, we examine the extent to which injected watermarks improve deepfake traceability. We compare the FakeMark detector's performance on non-watermarked, clean signals and randomly watermarked signals with the results from Table 1, where watermarks are chosen to always match the ground-truth system label. Results are presented in Table 4.

Table 4: Attribution accuracy results of FakeMark detector under different watermarking conditions.

Generator	Test set	MLAAD_v5			ASVspoof + TIMIT-TTS		
	Condition / Distortion	No watermark	Random	Matching	No watermark	Random	Matching
FakeMark^A	None	1.00	1.00	1.00	0.06	1.00	1.00
	Others averaged	0.73	0.80	0.86	0.03	0.77	0.79
FakeMark^T	None	0.99	1.00	1.00	0.05	1.00	1.00
	Others averaged	0.78	0.85	0.91	0.04	0.86	0.84

The injected watermarks improve deepfake traceability. Similar to the classifier baselines in clean conditions, the standalone FakeMark detector achieves near-perfect accuracy (above 0.99) on the MLAAD_v5 test set, but drops to 0.73 (FakeMark^A) and 0.78 (FakeMark^T) under distortions. Adding watermarks improves attribution accuracy for both variants and test conditions, regardless of whether the watermark is randomly assigned or matches the ground-truth label.

Although attribution achieves perfect accuracy with clean signals for both watermark injection conditions, **injecting watermarks that match the ground-truth label outperforms randomly assigned labels under distortions**. For in-domain MLAAD_v5 samples, FakeMark^A improves from 0.80 to 0.86, and FakeMark^T from 0.85 to 0.91. For cross-dataset samples, improvements are small or even absent (FakeMark^T from 0.86 to 0.84), which is expected given that the FakeMark detector primarily depends on the watermark messages when artifact patterns are unseen.

5 CONCLUSION

Motivated by the limitations of both classifier- and watermarking-based solutions for deepfake speech attribution, we proposed a novel watermarking framework FakeMark to enhance deepfake traceability. The core novelty of FakeMark is the injection of artifact-correlated watermarks, which allows the detector to leverage both watermark message and deepfake artifacts for attribution. Our results confirm that such design provides improved generalization and robustness across various seen and unseen datasets and under distortions.

Limitations of this work include considering only fully seen architectures during training and evaluation, which constrains the applicability of watermarking-based attribution when scaling to a large number of unseen deepfake systems. We also observe a trade-off between robustness and speech quality–stronger watermarks introduce more distortions to speech signal. Addressing this trade-off could be an important direction for future work.

REFERENCES

Taejun Bak, Junmo Lee, Hanbin Bae, Jinhyeok Yang, Jae-Sung Bae, and Young-Sun Joo. Avocodo: Generative adversarial network for artifact-free vocoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 12562–12570, 2023.

Kratika Bhagtani, Amit Kumar Singh Yadav, Paolo Bestagini, and Edward J Delp. Attribution of diffusion based deepfake speech generators. In *2024 IEEE International Workshop on Information Forensics and Security*, pp. 1–6. IEEE, 2024.

Sanyuan Chen, Chengyi Wang, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. Neural codec language models are zero-shot text to speech synthesizers. *IEEE Transactions on Audio, Speech and Language Processing*, 33:705–718, 2025a.

Xuanjun Chen, I Lin, Lin Zhang, Haibin Wu, Hung-yi Lee, Jyh-Shing Roger Jang, et al. Towards generalized source tracing for codec-based deepfake speech. *arXiv preprint arXiv:2506.07294*, 2025b.

Xuanjun Chen, I-Ming Lin, Lin Zhang, Jiawei Du, Haibin Wu, Hung yi Lee, and Jyh-Shing Roger Jang. Codec-based deepfake source tracing via neural audio codec taxonomy. In *Interspeech 2025*, pp. 1538–1542, 2025c.

Yongbaek Cho, Changhoon Kim, Yezhou Yang, and Yi Ren. Attributable watermarking of speech generative models. In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3069–3073. IEEE, 2022.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Junlong Deng, Yanzhen Ren, Tong Zhang, Hongcheng Zhu, and Zongkun Sun. VFD-Net: Vocoder fingerprints detection for fake audio. In *2024 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 12151–12155, 2024.

Yulin He, Hongxia Wang, Yiqin Qiu, and Hao Cao. ASSMark: Dual defense against speech synthesis attack via adversarial robust watermarking. *IEEE Signal Processing Letters*, 32:1870–1874, 2025.

Andrew Hines, Jan Skoglund, Anil Kokaram, and Naomi Harte. ViSQOL: The virtual speech quality objective Listener. In *International Workshop on Acoustic Signal Enhancement 2012*, pp. 1–4, 2012.

Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Zehan Wang, Ruiqi Li, and others. WavTokenizer: An efficient acoustic discrete codec tokenizer for audio language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Shengpeng Ji, Ziyue Jiang, Jialong Zuo, Minghui Fang, Yifu Chen, Tao Jin, and Zhou Zhao. Speech watermarking with discrete intermediate representations. In *The 39th Annual AAAI Conference on Artificial Intelligence*, 2025b.

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Yanqing Liu, Yichong Leng, Kaitao Song, Siliang Tang, Zhizheng Wu, Tao Qin, Xiang-Yang Li, Wei Ye, Shikun Zhang, Jiang Bian, Lei He, Jinyu Li, and Sheng Zhao. NaturalSpeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Lauri Juvela and Xin Wang. Collaborative watermarking for adversarial speech synthesis. In *2024 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 11231–11235, 2024.

Lauri Juvela and Xin Wang. Audio codec augmentation for robust collaborative watermarking of speech synthesis. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5. IEEE, 2025.

Andre Kassis and Urs Hengartner. UnMarker: A universal attack on defensive image watermarking. In *2025 IEEE Symposium on Security and Privacy*, pp. 2602–2620. IEEE, 2025.

Nicholas Klein, Tianxiang Chen, Hemlata Tak, Ricardo Casal, and Elie Khoury. Source tracing of audio deepfake systems. In *Interspeech 2024*, pp. 1100–1104, 2024.

Nicholas Klein, Hemlata Tak, and Elie Khoury. Open-set source tracing of audio deepfake systems. In *Interspeech 2025*, pp. 1578–1582, 2025.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033, 2020.

Sang-gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. BigVGAN: A universal neural vocoder with large-scale training. In *The Eleventh International Conference on Learning Representations*, 2023.

Haiyun Li, Zhiyong Wu, Xiaofeng Xie, Jingran Xie, Yaoxun Xu, and Hanyang Peng. VoiceMark: Zero-shot voice cloning-resistant watermarking approach leveraging speaker-specific latents. In *Interspeech 2025*, pp. 5108–5112, 2025a.

Yue Li, Weizhi Liu, and Dongdong Lin. TriniMark: A robust generative speech watermarking method for trinity-level attribution. *arXiv preprint arXiv:2504.20532*, 2025b.

Chang Liu, Jie Zhang, Tianwei Zhang, Xi Yang, Weiming Zhang, and Nenghai Yu. Detecting voice cloning attacks via timbre watermarking. In *Network and Distributed System Security Symposium*, 2024a.

Hongbin Liu, Moyang Guo, Zhengyuan Jiang, Lun Wang, and Neil Zhenqiang Gong. AudioMarkBench: Benchmarking robustness of audio watermarking. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b.

Rui Liu, Berrak Sisman, Guanglai Gao, and Haizhou Li. Expressive TTS training with frame and style reconstruction loss. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1806–1818, 2021.

Yixin Liu, Lie Lu, Jihui Jin, Lichao Sun, and Andrea Fanelli. XAttnMark: Learning robust audio watermarking with cross-attention. In *Forty-second International Conference on Machine Learning*, 2025.

Aurosweta Mahapatra, Ismail R. Ulgen, Abinay Reddy Naini, Carlos Busso, and Berrak Sisman. Can emotion fool anti-spoofing? In *Interspeech 2025*, pp. 5628–5632, 2025.

Bartłomiej Marek, Piotr Kawa, and Piotr Syga. Are audio deepfake detection models polyglots? *arXiv preprint arXiv:2412.17924*, 2024.

Nicolas Müller. Using MLAAD for source tracing of audio deepfakes. `https://deepfake-total.com/sourcetracing`, 11 2024.

Nicolas Müller, Franziska Diekmann, and Jennifer Williams. Attacker attribution of audio deepfakes. In *Interspeech 2022*, pp. 2788–2792, 2022.

Nicolas Müller, Piotr Kawa, Wei Herng Choong, Edresson Casanova, Eren Gölge, Thorsten Müller, Piotr Syga, Philip Sperl, and Konstantin Böttinger. MLAAD: The multi-language audio anti-spoofing dataset. *International Joint Conference on Neural Networks*, 2024.

Viola Negroni, Davide Salvi, Paolo Bestagini, and Stefano Tubaro. Source verification for speech deepfakes . In *Interspeech 2025*, pp. 1548–1552, 2025.

Patrick O'Reilly, Zeyu Jin, Jiaqi Su, and Bryan Pardo. Deep audio watermarks are shallow: Limitations of post-hoc watermarking techniques for speech. In *The 1st Workshop on GenAI Watermarking*, 2025.

Jiahui Pan, Shuai Nie, Hui Zhang, Shulin He, Kanghao Zhang, Shan Liang, Xueliang Zhang, and Jianhua Tao. Speaker recognition-assisted robust audio deepfake detection. In *Interspeech 2022*, pp. 4202–4206, 2022.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52, 2024.

Maria T Quiñonez-Carbajal, Rogelio Reyes-Reyes, Volodymyr Ponomaryov, Clara Cruz-Ramos, and Beatriz P Garcia-Salgado. Speech signal authentication and self-recovery based on DTWT and ADPCM. *Multimedia Tools and Applications*, pp. 1–25, 2024.

Antony W. Rix, John G. Beerends, Michael P. Hollier, and Andries P. Hekstra. Perceptual Evaluation of Speech Quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pp. 749–752, 2001.

Robin San Roman, Pierre Fernandez, Hady Elsahar, Alexandre Défossez, Teddy Furon, and Tuan Tran. Proactive detection of voice cloning with localized watermarking. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Robin San Roman, Pierre Fernandez, Antoine Deleforge, Yossi Adi, and Romain Serizel. Latent watermarking of audio generative models. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2025.

Davide Salvi, Brian Hosler, Paolo Bestagini, Matthew C. Stamm, and Stefano Tubaro. TIMIT-TTS: A text-to-speech dataset for multimodal synthetic media detection. *IEEE Access*, 11:50851–50866, 2023.

Hubert Siuzdak. Vocos: Closing the gap between time-domain and Fourier-based neural vocoders for high-quality audio synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

David Snyder, Guoguo Chen, and Daniel Povey. MUSAN: A music, speech, and noise corpus. *arXiv:1510.08484*, 2015.

Adriana Stan, David Combei, Dan Oneata, and Horia Cucu. TADA: Training-free attribution and out-of-domain detection of audio deepfakes. In *Interspeech 2025*, pp. 1543–1547, 2025.

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

Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*, 2021.

Andros Tjandra, Yi-Chiao Wu, Baishan Guo, John Hoffman, Brian Ellis, Apoorv Vyas, Bowen Shi, Sanyuan Chen, Matt Le, Nick Zacharov, et al. Meta Audiobox Aesthetics: Unified automatic quality assessment for speech, music, and sound. *arXiv preprint arXiv:2502.05139*, 2025.

Xin Wang, Héctor Delgado, Hemlata Tak, Jee weon Jung, Hye jin Shim, Massimiliano Todisco, Ivan Kukanov, Xuechen Liu, Md Sahidullah, Tomi H. Kinnunen, Nicholas Evans, Kong Aik Lee, and Junichi Yamagishi. ASVspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale. In *The Automatic Speaker Verification Spoofing Countermeasures Workshop (ASVspoof 2024)*, pp. 1–8, 2024.

Zhigang Wang, Dengpan Ye, Jingyang Li, and Jiacheng Deng. Generalize audio deepfake algorithm recognition via attribution enhancement. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5. IEEE, 2025.

Chia-Hua Wu, Wanying Ge, Xin Wang, Junichi Yamagishi, Yu Tsao, and Hsin-Min Wang. A comparative study on proactive and passive detection of deepfake speech. In *Interspeech 2025*, pp. 5328–5332, 2025.

Pei Yang, Hai Ci, Yiren Song, and Mike Zheng Shou. Can simple averaging defeat modern watermarks? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Xuefeng Yang, Jian Guan, Feiyang Xiao, Congyi Fan, Haohe Liu, Qiaoxi Zhu, Dongli Xu, and Youtian Lin. DualMark: Identifying model and training data origins in generated audio. *arXiv preprint arXiv:2508.15521*, 2025.

Lingfeng Yao, Chenpei Huang, Shengyao Wang, Junpei Xue, Hanqing Guo, Jiang Liu, Phone Lin, Tomoaki Ohtsuki, and Miao Pan. Yours or mine? Overwriting attacks against neural audio watermarking. *arXiv preprint arXiv:2509.05835*, 2025.

Heiga Zen, Keiichi Tokuda, and Alan W Black. Statistical parametric speech synthesis. *speech communication*, 51(11):1039–1064, 2009.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. SpeechTokenizer: Unified speech tokenizer for speech language models. In *The Twelfth International Conference on Learning Representations*, 2024a.

Xueyao Zhang, Liumeng Xue, Yicheng Gu, Yuancheng Wang, Jiaqi Li, Haorui He, Chaoren Wang, Songting Liu, Xi Chen, Junan Zhang, Zihao Fang, Haopeng Chen, Tze Ying Tang, Lexiao Zou, Mingxuan Wang, Jun Han, Kai Chen, Haizhou Li, and Zhizheng Wu. Amphion: An open-source audio, music, and speech generation toolkit. In *2024 IEEE Spoken Language Technology Workshop*, pp. 879–884, 2024b.

Yigitcan Özer, Woosung Choi, Joan Serrà, Mayank Kumar Singh, Wei-Hsiang Liao, and Yuki Mitsufuji. A comprehensive real-world assessment of audio watermarking algorithms: Will they survive neural codecs? In *Interspeech 2025*, pp. 5113–5117, 2025.

Adrian Łańcucki. FastPitch: Parallel text-to-speech with pitch prediction. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6588–6592, 2021.

A Appendix

A.1 The use of large language models

Large Language Models (LLMs) were used only to polish the final version of the manuscript. The authors confirm that their original intention and meaning were not altered during this process.

A.2 List of augmentations during training

Below is the list of transformations used as augmentation strategies during all system training in our experiments. They are reproduced from the AudioSeal (Roman et al., 2024) pipeline. During training, each transformation is selected at random with equal probability and the chosen transform is applied to the current mini-batch. They include:

1. EnCodec: Inputs are resamples to 24kHz, compressed and reconstructed with EnCodec (Défossez et al., 2023) with $nq = 16$, and resampled back to 16kHz.
2. Speed: Playback speed of input signal is changed randomly between 0.9 and 1.1.
3. Resample: Inputs are resampled to 32kHz and resampled back to 16kHz.
4. Echo: A delay and less loud copy of the original is added to the input signal. Delay time is randomly sampled between 0.1 and 0.5 seconds, volume of the copied signal is randomly chosen between 0.1 and 0.5.
5. White noise: Gaussian noise with standard deviation fixed at 0.001 is added to the input signals.
6. Pink noise: Pink noise with standard deviation fixed at 0.01 is added to the input signals.
7. Lowpass filtering: A lowpass filter is applied to the input signal with a cutoff frequency at 5kHz.
8. Highpass filtering: A highpass filter is applied to the input signal with a cutoff frequency at 500Hz.
9. Bandpass filtering: A bandpass filter is applied to the input signal with a lower cutoff frequency of 300Hz and an upper cutoff frequency of 8kHz.
10. Smoothing: Inputs are smoothed using a moving average filter with a variable window size between 2 and 10.
11. Boost: Amplitude of input signal is multiplied by 1.2.
12. Duck: Amplitude of input signal is multiplied by 0.8.
13. AAC: Input signal is encoded in AAC format at 128kbps bitrate.

14. MP3: Input signal is encoded in MP3 format at 128kbps bitrate.
15. Identity: Returns the unprocessed input signal.

A.3 FAKEMARK MODULE ARCHITECTURES

FakeMark^A We adopt the original AudioSeal generator architecture. The encoder uses a 1D convolution (32 channels, kernel size 7) followed by four convolutional blocks, each containing a residual unit (two kernel-3 convolutions with skip connection, doubling channels during down-sampling) and a down-sampling convolution (stride S, kernel $K = 2S$; $S = 2, 4, 5, 8$). It concludes with a two-layer LSTM and a final 1D convolution (128 channels, kernel 7) using ELU activations. The decoder mirrors the encoder with transposed convolutions and reversed strides. The latent dimension H is 128.

FakeMark^T We adopt the Timbre encoder architecture but with larger size and hidden dimension (128). A 1024-point Short-Time Fourier Transform (STFT) with 256 hop length is applied to obtain the magnitude spectrogram and phase of the input signal. The magnitude is fed to the 5-layer Carrier Encoder to obtain the encoded carrier feature, which is then concatenated with the original magnitude and the repeated watermark embedding $\boldsymbol{E}_w$. This combined feature is passed to the 5-layer Watermark Embedder to generate the magnitude spectrogram of watermark signal. The watermarked magnitude spectrogram is obtained by adding watermark magnitude with original clean magnitude. This is different to the original Timber implementation where the Watermark Embedder directly outputs the watermarked magnitude. The watermarked signal is reconstructed via inverse STFT using the original phase and watermarked magnitude. The same original phase is also used for generating watermark waveform with watermark magnitude. The latent dimension H is 513.

Detector We use an identical detector architecture for both FakeMark generators. The detector contains a pre-trained wav2vec model (namely the MMS-300M) as front-end. It extract a 1024-dimensional sequence-level representations from the input signal. These representations are then passed through a global average pooling layer to aggregate temporal information, followed by a fully connected layer that produced the output probabilities of 12 classes.

A.4 DATASETS DETAILS

Both the MLAAD_v5 dataset and source tracing challenge protocol can be downloaded from `https://deepfake-total.com/sourcetracing`.

The ASVspoof5 dataset can be downloaded from `https://huggingface.co/datasets/jungjee/asvspoof5`.

The TIMIT-TTS dataset can be downloaded from `https://zenodo.org/records/6560159`.

A.5 TRAINING AND IMPLEMENTATION DETAILS

AudioSeal We use the official AudioSeal implementation from `https://github.com/facebookresearch/audioseal`.

Timbre We use the official Timbre implementation from `https://github.com/TimbreWatermarking/TimbreWatermarking`.

FakeMark For FakeMark training, the learning rate was linearly increased to 1×10^{-4} over the first 2,000 mini-batches, and then linearly decayed to 0 at the 50,000th mini-batch, where training stops. All input signals were resampled to 16 kHz if necessary. The waveform amplitude of training samples was randomly adjusted according to the Active Speech Level (ASL) based on ITU-T P.56. Training data were dynamically sampled by grouping files of similar durations and zero-padding them to form mini-batches, with a maximum batch duration of 40 seconds. Files longer than 10 seconds were randomly trimmed to durations between 6 and 10 seconds during training.

Table 5: Summary of TTS models, Class ID, watermark bits, and number of samples in train, validation, and test sets of MLAAD_v5 dataset.

TTS Model	Class ID	Watermark Bits	Train	Validation	Test
Mars5	0	(0,1,0,0)	275	23	300
MeloTTS	1	(0,0,1,0)	274	22	300
Metavoice-1B	2	(1,1,1,0)	267	29	300
facebook-mms-tts-deu	3	(1,1,0,0)	265	31	300
tts_models-en-ljspeech-fast_pitch	4	(1,0,1,1)	277	23	0
tts_models-it-mai_female-glow-tts	5	(1,0,1,0)	277	18	0
griffin_lim	6	(0,1,1,1)	1359	125	300
suno-bark	7	(0,0,0,1)	137	16	79
suno-bark-small	7	(0,0,0,1)	126	19	221
tts_models-en-ljspeech-tacotron2-DCA	8	(1,1,1,1)	272	25	49
tts_models-fr-mai-tacotron2-DDC	8	(1,1,1,1)	264	34	65
tts_models-de-thorsten-tacotron2-DDC	8	(1,1,1,1)	261	36	64
tts_models-en-ljspeech-tacotron2-DDC	8	(1,1,1,1)	142	11	32
tts_models-en-ljspeech-tacotron2-DDC_ph	8	(1,1,1,1)	135	11	90
tts_models-en-ljspeech-speedy-speech	9	(1,0,0,0)	268	28	0
tts_models-it-mai_male-vits	10	(0,0,1,1)	272	26	44
tts_models-fr-css10-vits	10	(0,0,1,1)	270	27	62
tts_models-it-mai_female-vits	10	(0,0,1,1)	269	29	60
tts_models-lt-cv-vits	10	(0,0,1,1)	264	34	53
tts_models-de-css10-vits-neon	10	(0,0,1,1)	264	35	60
tts_models-en-ljspeech-vits–neon	10	(0,0,1,1)	261	37	21
tts_models-multilingual-multi-dataset-xtts_v2	11	(1,1,0,1)	1898	185	154
tts_models-multilingual-multi-dataset-xtts_v1.1	11	(1,1,0,1)	1623	157	128
vixTTS	11	(1,1,0,1)	280	19	18

Table 6: TTS models, source dataset, Class IDs, watermark bits, and sample counts for cross-dataset evaluation.

TTS Model	Source Dataset	Class ID	Watermark Bits	Number of Samples
A01-GlowTTS	ASVspoof5	5	(1,0,1,0)	160
A07-FastPitch	ASVspoof5	4	(1,0,1,1)	160
fastpitch	TIMIT-TTS	4	(1,0,1,1)	160
glowtts	TIMIT-TTS	5	(1,0,1,0)	160
A11-Tacotron2	ASVspoof5	8	(1,1,1,1)	160
A29-XTTS	ASVspoof5	11	(1,1,0,1)	160
A08-VITS	ASVspoof5	10	(0,0,1,1)	137
vits	TIMIT-TTS	10	(0,0,1,1)	23

Validation was performed every 500 mini-batches, and the best model was selected based on the lowest sum of attribution loss and watermark detection loss. Test samples were neither amplitude-adjusted nor trimmed.

The balancing weights for training were set as follows: attribution loss, 10.0; watermark detection loss, 10.0; HiFi-GAN losses, 1.0 (with L_1 spectrogram loss weight 1.0 and feature matching loss weight 1.0); AudioSeal perceptual losses, 0.1 for L_1 loss, 10.0 for loudness loss, and 1.0 for frequency magnitude loss.

AudioSeal was trained on 6 NVIDIA A100 GPUs. The left training were performed on a single NVIDIA H100 GPU.

MMS-300M Classifier We adopt the same architecture as the FakeMark detector and use the same codebase and training procedure, except that the maximum learning rate is set to 1×10^{-5} and the batch size is fixed at 16. Training stops after 30,000 mini-batches. Best model selection is based on the classification accuracy on validation set.

ResNet34 Classifier We use a standard ResNet34 architecture with a temporal statistics pooling layer (TSPL) to extract a 128-dimensional embedding from the input signal, followed by a fully connected layer for prediction. The input is a randomly selected 4-second segment of the original signal, padded if shorter. Following Klein et al. (2025), we use 80-dimensional log linear filter-bank (LFB) features of the speech signal, computed with a 400-sample window, 160-sample hop, and a 400-point FFT. We further compute delta (Δ) and double-delta ($\Delta\Delta$) features, and apply cepstral mean and variance normalization (CMVN), yielding a final feature dimension of 240. The model is trained using the Large Margin Cosine Loss with default settings from the implementation in `https://github.com/YirongMao/softmax_variants/blob/master/model_utils.py#L103`. All hyperparameters are identical to those used for MMS-300M training, except that the maximum learning rate is set to 1×10^{-4}.

A.6 List of distortions during evaluation

The settings of distortion and watermark removal attacks are:

1. Pitch shift: Pitch is randomly shifted between -1 and 1 semitones.
2. Playback speed: Original speed is adjust to a number randomly sampled between 0.95 and 1.05.
3. Noise: Random noise from MUSAN noise recordings is applied at 0dB SNR.
4. BigVGAN: Using code and pre-trained weight from `https://github.com/NVIDIA/BigVGAN`. Input signals are resampled to 24kHz, passed to BigVGAN vocoder, and resampled back to 16kHz.
5. HiFi-GAN: Using the pre-trained weights from `https://huggingface.co/speechbrain/tts-hifigan-libritts-16kHz`.
6. Vocos: Using code and pre-trained weight (vocos-mel-24khz) from `https://github.com/gemelo-ai/vocos/tree/main`. Input signals are resampled to 24kHz, passed to Vocos, and resampled back to 16kHz.
7. SpeechTokenizer: Using code and pre-trained weight (speechtokenizer_hubert_avg) from `https://github.com/ZhangXInFD/SpeechTokenizer`.
8. FACodec: Using code and pre-trained weight from `https://huggingface.co/amphion/naturalspeech3_facodec`.
9. WavTokenizer: Using code and pre-trained weight (WavTokenizer-small-600-24k-4096) from `https://huggingface.co/amphion/naturalspeech3_facodec`. Input signals are resampled to 24kHz, passed to WavTokenizer, and resampled back to 16kHz.
10. Overwriting: Input signals are sequentially passed through pre-trained Timbre and AudioSeal models three times to obtain the watermarked signal.
11. Averaging: Data samples from the zh-CN subset of the Common Voice dataset are processed using the pre-trained AudioSeal model. The resulting watermark signals for each sample are summed and averaged, and this averaged watermark is then subtracted from the input signal. We did not apply the Averaging attack with pre-trained Timbre model because its generator directly outputs the watermarked signal rather than estimating a separate watermark.

A.7 Visualizations of speech signals

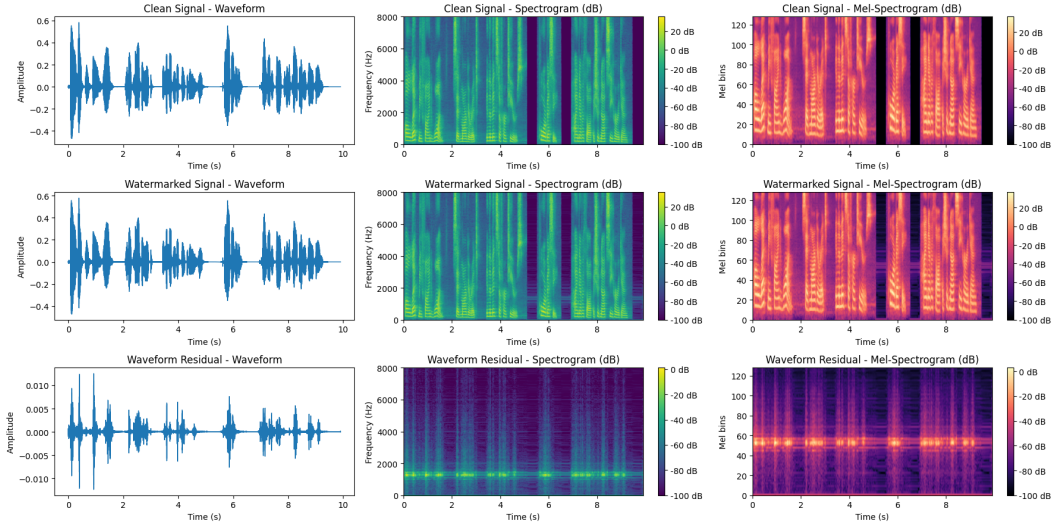

Figure 3: Visualization of AudioSeal watermarking on MLAAD-en-tts_models-en-ljspeech-tacotron2-DDC-northandsouth_27_f000104.

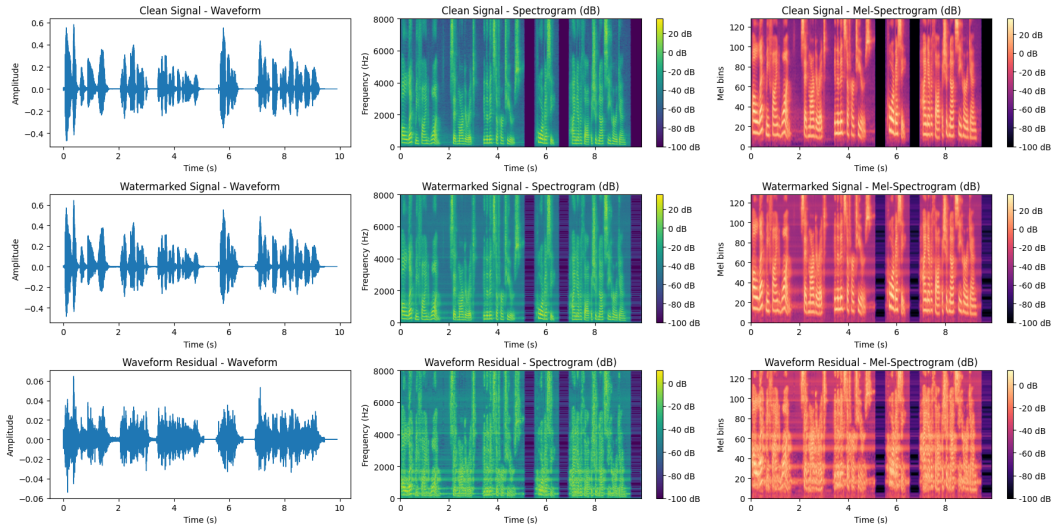

Figure 4: Visualization of Timbre watermarking on MLAAD-en-tts_models-en-ljspeech-tacotron2-DDC-northandsouth_27_f000104.

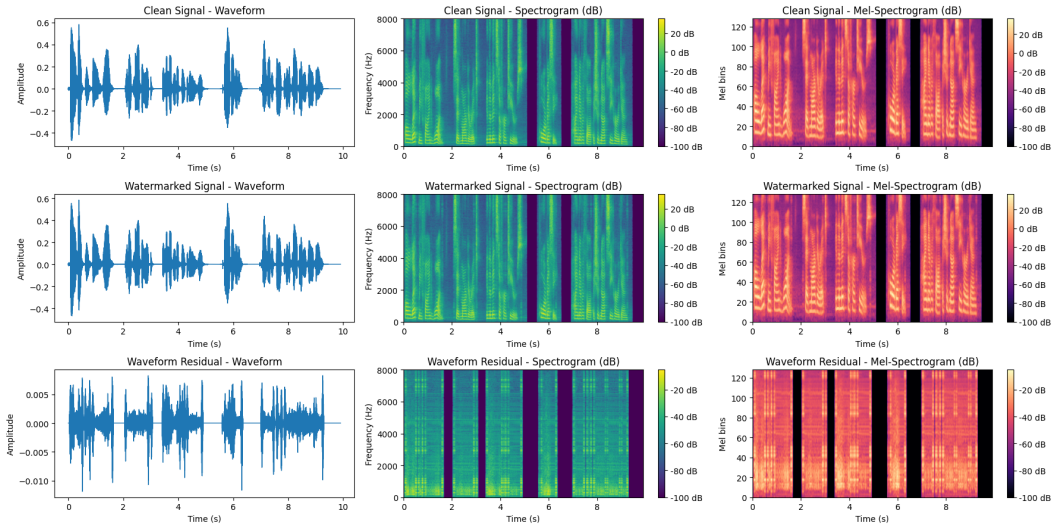

Figure 5: Visualization of FakeMark^{A} watermarking on MLAAD-en-tts_models-en-ljspeech-tacotron2-DDC-northandsouth_27_f000104.

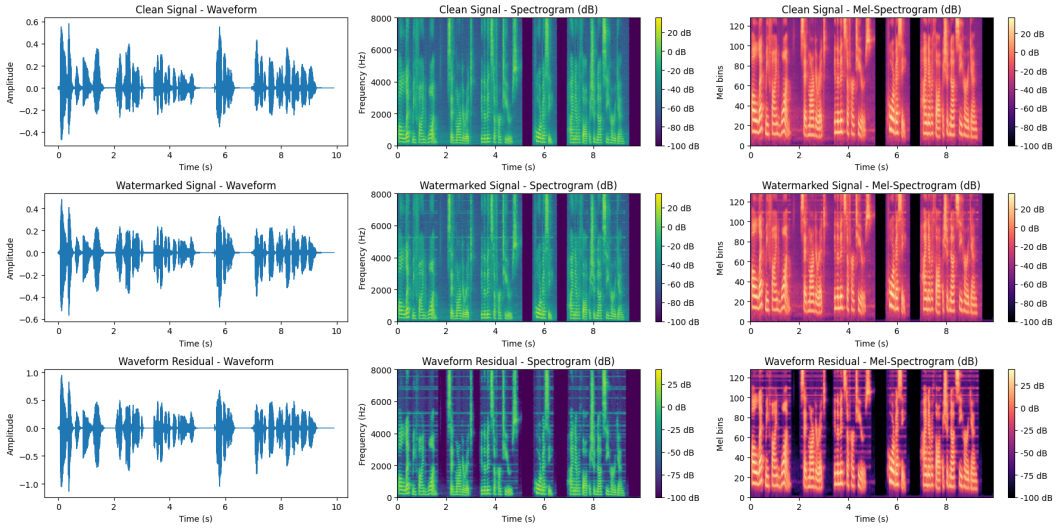

Figure 6: Visualization of FakeMark^{T} watermarking on MLAAD-en-tts_models-en-ljspeech-tacotron2-DDC-northandsouth_27_f000104.

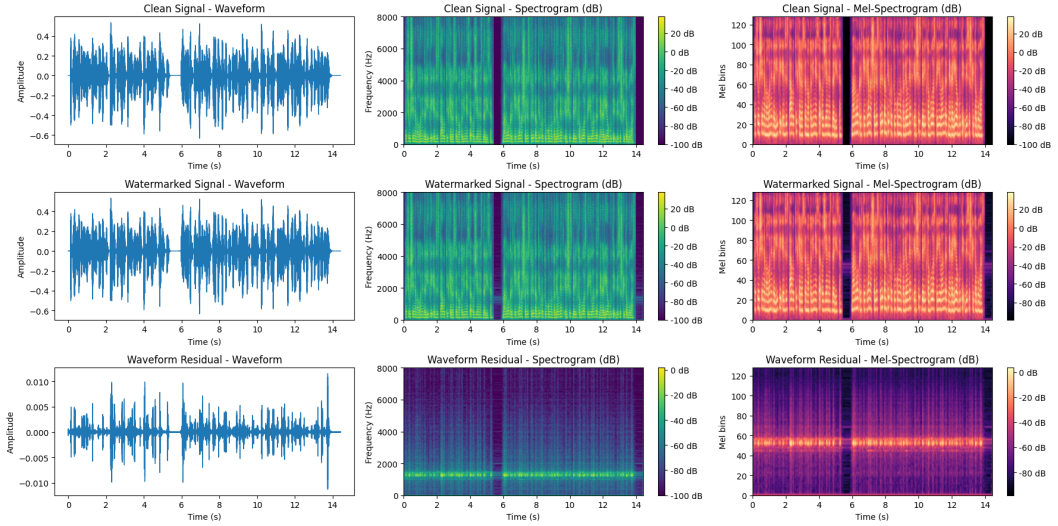

Figure 7: Visualization of AudioSeal watermarking on MLAAD-lt-tts_models-lt-cv-vits-emerald_city_of_oz_03_f000037.

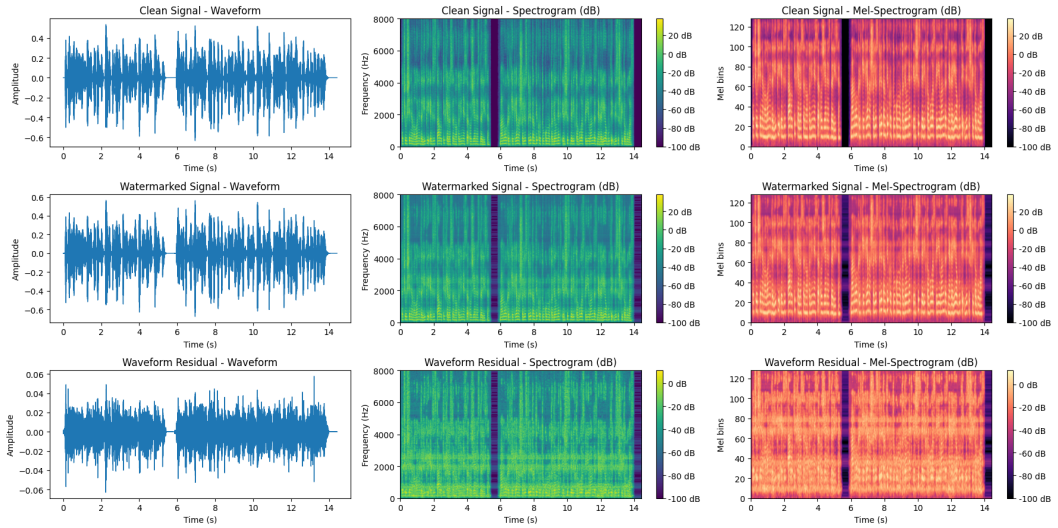

Figure 8: Visualization of Timbre watermarking on MLAAD-lt-tts_models-lt-cv-vits-emerald_city_of_oz_03_f000037.

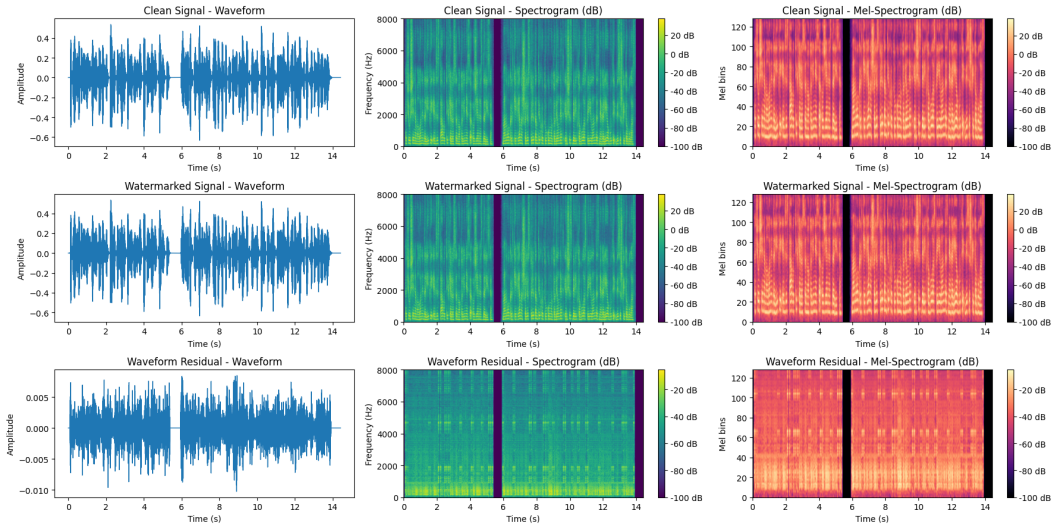

Figure 9: Visualization of FakeMark^A watermarking on MLAAD-lt-tts_models-lt-cv-vits-emerald_city_of_oz_03_f000037.

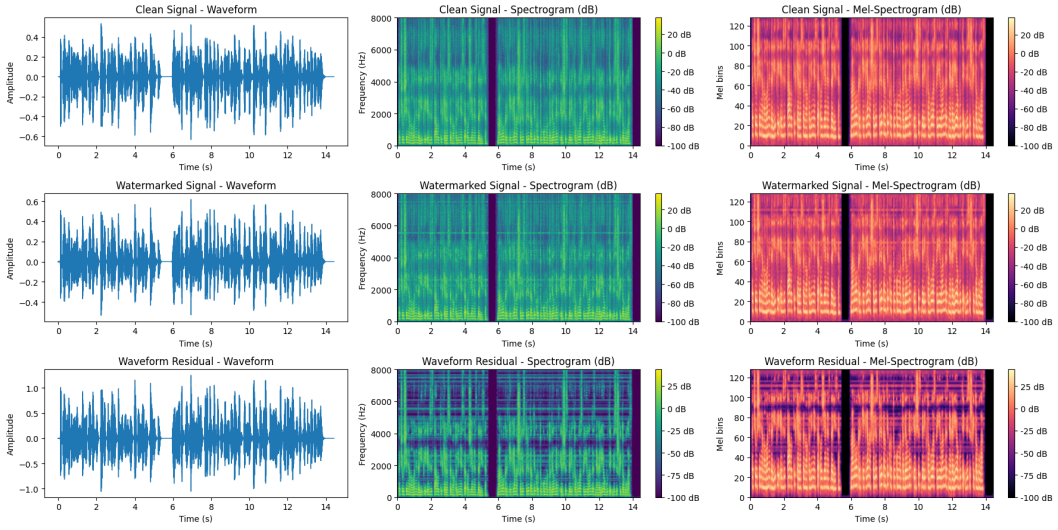

Figure 10: Visualization of FakeMark^T watermarking on MLAAD-lt-tts_models-lt-cv-vits-emerald_city_of_oz_03_f000037.