WaveVerify: A Novel Audio Watermarking Framework for Media Authentication and Combatting Deepfakes

Aditya Pujari and Ajita Rattani
University of North Texas
Denton, Texas, USA
adityapujari@my.unt.edu; ajita.rattani@unt.edu

Abstract

With the rise of voice synthesis technology threatening digital media integrity, audio watermarking has become a crucial defense for content authentication. However, current solutions often lack robustness to effects like high-pass filtering and temporal modifications and suffer from poor watermark localization. We introduce ***WaveVerify****, a watermarking system that addresses these challenges using a Feature-wise Linear Modulation (FiLM)-based generator for resilient multiband watermark embedding and a Mixture-of-Experts detector for accurate extraction and localization. Our unified training framework enhances robustness by applying multiple distortions per backpropagation step via a dynamic effect scheduler. Evaluations across multiple datasets show WaveVerify outperforms SOTA models like AudioSeal and WavMark, achieving zero Bit Error Rate (BER) under common distortions and MIoU scores of 0.98+ under severe temporal modifications. The parallel FiLM-based generator also reduces training time by ∼80% compared to sequential embedding approaches. Code and pretrained models are available at:* https://github.com/vcbsl/WaveVerify.

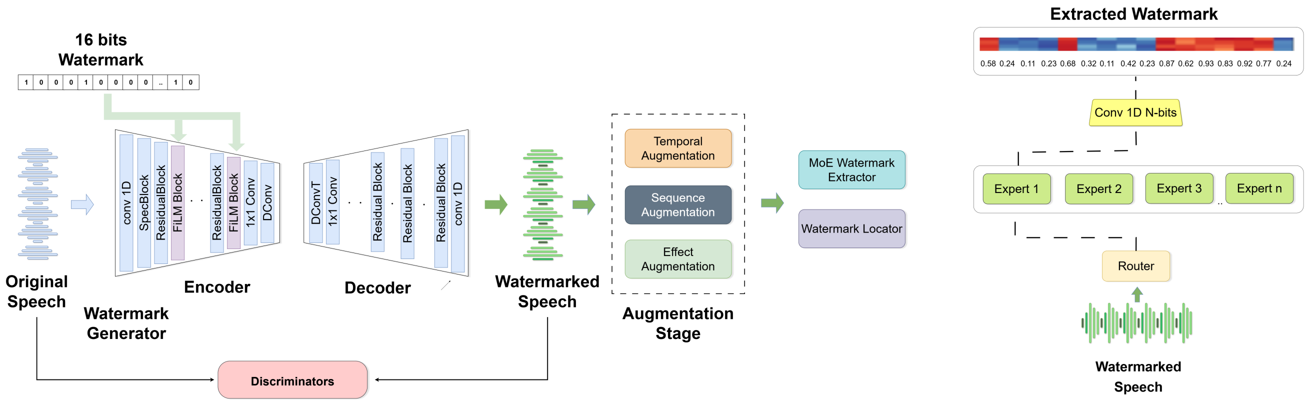
1. Introduction

The rapid advancement of voice generation technologies has enabled the synthesis of speech that is perceptually indistinguishable from genuine human voices [33, 17]. While these innovations facilitate beneficial applications such as personalized text-to-speech systems [37] and voice preservation [16], they have also introduced significant risks, including deepfake impersonation scams [23] and synthetic media-driven disinformation campaigns [7]. Recent reports indicate that in 2024, deepfake fraud attempts surged by over 1,300% compared to 2023 [26], underscoring the urgent need for robust audio content authentication. The financial sector has been particularly impacted, with a loss of over $10 million to voice scams [2] and individual victims reporting losses exceeding $6,000 from AI-generated deepfake calls [35]. In response, regulators and governments worldwide are enacting measures to improve AI content transparency and traceability, emphasizing the development of forensic tools and watermarking techniques as essential strategies to uphold media integrity.

A predominant forensic approach for detecting synthesized audio involves training binary classifiers to distinguish between natural and synthetic speech, as demonstrated in [36, 39]. This method, commonly referred to as *passive detection*, offers a relatively straightforward mitigation strategy. However, its effectiveness is increasingly challenged as generative models continue to improve, narrowing the perceptual and statistical gap between authentic and synthetic audio.

Audio watermarking has thus emerged as a promising alternative for asserting ownership and authenticity of audio media. By embedding imperceptible yet robust signals within audio content, watermarking enables reliable detection and verification [28, 12]. Deep learning-based watermarking methods have addressed many limitations of traditional techniques (e.g., spread spectrum [15], patchwork [38], and echo hiding [4]), improving imperceptibility, retrieval accuracy (measured by Bit Error Rate, BER), and resilience to audio effects. These methods typically employ a generator for watermark embedding, an effect simulator for robustness, and a detector for retrieval [40, 21, 30]. However, most rely on audio effect simulators with fixed distortion parameters, which can limit effectiveness against unforeseen real-world attacks (audio modifications/ manipulations).

Recent innovations such as WavMark [12] and AudioSeal [28] have improved these foundations. WavMark introduced synchronization patterns for watermark localization, while AudioSeal achieved faster detection speeds and enhanced sample-wise watermark localization. Despite these advances, current watermarking techniques *remain vulnerable* to temporal manipulations (e.g., reversal



(a) End-to-end Training Pipeline.

(b) MoE Watermark Extractor.

Figure 2. (a) The end-to-end training pipeline of WaveVerify, where a FiLM-based generator embeds bits into speech, followed by temporal/sequence/effect augmentations, and extraction via a locator and MoE detector. (b) MoE watermark detector architecture: Input audio is processed by a HILCodec encoder, then routed through multiple expert networks via a learned gating mechanism to produce the final extracted watermark bits.

and speed changes), structural modifications (e.g., trimming and segment shuffling), and obtain limited localization precision, which undermines their effectiveness in adversarial or real-time scenarios.

WaveVerify addresses these aforementioned limitations by rethinking watermark embedding and detection. It uses a Feature-wise Linear Modulation (FiLM)-based generator to adaptively spread watermark features across multiple frequency bands and temporal scales, enhancing robustness against spectral and temporal attacks. For detection, a Mixture-of-Experts (MoE) architecture routes extraction through specialized sub-networks resilient to distortions. Its training pipeline includes aggressive, adaptive augmentations, enabling reliable detection even under severe audio alterations.

The key **innovations** of WaveVerify are listed as follows:

- **A robust FiLM-based generator architecture** that adaptively and uniformly embeds and distributes watermark features across multiple frequency bands and temporal scales to ensure robustness to audio effects and temporal manipulations.

- **A Mixture-of-Experts detector** for resilient watermark extraction by dynamically routing through specialized sub-networks tailored to diverse distortions and manipulations, consequently enhancing robustness to a wide range of unseen audio manipulations.

- **A dynamic augmentation and effect scheduling strategy** that accelerates training and ensures robustness to a wide range of real-world audio effects or distortions.

- **Comprehensive empirical validation** demonstrating state-of-the-art performance in both watermark bit recovery (BER) and localization (MIoU), with significant improvements and robustness against attacks over prior work.

2. Related Work

Audio watermarking has evolved significantly, beginning with traditional signal processing techniques that laid the foundation for imperceptible and robust embedding. Early approaches, such as spread spectrum watermarking, distributed signals across the frequency domain to resist interference [15], while echo hiding techniques embedded watermarks by introducing imperceptible echoes with specific delay parameters [10]. Phase-based methods further advanced robustness by modifying phase components of audio signals [6]. However, these methods struggled against modern audio transformations like compression and filtering, as highlighted by Cox et al. [14] and Bassia et al. [8], underscoring the need for more adaptive solutions.

The advent of deep learning revolutionized audio watermarking by enabling end-to-end optimization of watermark embedding and extraction processes. HiDDeN introduced a pioneering framework for a trainable watermarking solution, leveraging neural networks to enhance robustness against diverse distortions [40]. More recent deep learning-based methods have further improved adaptability and performance by jointly training generator and detector networks, often incorporating adversarial objectives and distortion simulation during training [11, 20]. For instance, IDEAW employs invertible dual-embedding and attack-aware training to improve both capacity and localization, addressing key challenges in neural watermarking [20].

Recent techniques like WavMark [12] and AudioSeal [28] have further improved performance through synchronization patterns and joint optimization of generator-detector networks, respectively. WavMark leverages invertible neural networks for reciprocal encoding and decoding, achieving high capacity and imperceptibility, as well as automatic watermark localization. AudioSeal achieves superior temporal precision and detection speeds but faces challenges with specific attack vectors like high-pass filtering or temporal modifications. Benchmarking studies such as AudioMarkBench confirm that while these methods are robust to many black-box forgery attacks, they remain vulnerable to adaptive watermark-removal strategies, especially those that exploit detector APIs or operate in the spectrogram domain [1].

Despite these advancements, existing watermarking techniques remain vulnerable, particularly to temporal and combined attacks that involve multiple types of distortions (e.g., speed changes combined with spectral filtering, or compression followed by noise addition) [11, 1]. Moreover, recent work has highlighted the limitations of current schemes in the face of generative AI-driven attacks, which can remove watermarks while preserving audio quality, raising concerns about the long-term viability of watermarking for intellectual property protection [11]. While solutions like SilentCipher [30] have explored information-theoretic capacity optimization and adversarial training to counter unseen distortions, their practical application is limited by constraints such as a lack of explicit localization capabilities. These limitations motivate the development of WaveVerify, which addresses these gaps through a novel FiLM-based generator architecture and Mixture-of-Experts detector design.

3. Proposed Method

3.1. WaveVerify Framework Overview

Figure 2 shows the overview of the framework of WaveVerify. The proposed WaveVerify framework introduces a novel end-to-end approach for robust audio watermarking, integrating three synergistic components within a unified Audio Watermarking architecture as illustrated in Figure 2(a). At its core, the framework consists of a Generator, a Locator, and a Detector, each optimized for specific tasks in the watermarking pipeline.

The Generator trained adversarially alongside a Discriminator, employs an encoder-decoder architecture with Feature-wise Linear Modulation (FiLM) [25] for watermark embedding. This design choice is crucial for computational efficiency, as conventional bottleneck embedding approaches require significantly more training time due to the sequential nature of processing through the entire network. By leveraging FiLM for adaptive modulation at multiple hierarchical levels, our approach distributes the watermark embedding process across the network architecture, enabling more efficient parallel computation and reducing overall training time by approximately 80% compared to our implemented bottleneck-based alternatives (embedding watermark at the bottleneck layers) while maintaining robust watermark integration. Direct training time comparisons with existing methods, such as AudioSeal, are not feasible due to variations in hardware configurations and the lack of publicly available or executable implementation code.

The framework's Locator enables precise identification of watermark presence or absence for each individual audio sample (i.e., sample-level identification) of watermarked regions while maintaining minimal computational overhead. Complementing these components, the Detector utilizes a sophisticated architecture (approximately 4.5M parameters) featuring a Mixture-of-Experts (MoE) module with multiple specialized expert networks (referred to as subnetworks), significantly enhancing the model's capacity to diverse audio attacks (detailed in Figure 2(b)).

To further improve robustness against audio edits and reduce training time, the framework incorporates temporal augmentations and audio effects, using a dynamic effect scheduler, during the end-to-end training stage. Next, we discuss these individual components of the WaveVerify framework as follows:

3.2. Watermark Embedding and Modulation

Previous watermarking approaches like AudioSeal [28] use ”direct linear extension” to embed watermarks by simply repeating fixed message-derived vectors across the temporal dimensions of intermediate audio features. This method lacks resilience to temporal modifications in audio, resulting in suboptimal and uneven distribution of watermark information across different time segments. We embed watermarks across the entire audio signal without using voice activity detection. Addressing these limitations, we introduce a sophisticated hierarchical modulation architecture that operates at multiple temporal scales as follows:

- A message processing module employs a multi-layer perceptron (MLP) to transform the n-bit watermark sequence into adaptive modulation parameters (scale γ and shift β vectors), enabling dynamic control over the watermark embedding process

- The framework implements multi-scale feature modulation throughout the encoder's hierarchical layers, utilizing frequency-specific FiLM (Feature-wise Linear Modulation) [25] layers to apply adaptive modulation across multiple frequency bands, enhancing the robustness of the watermarking to temporal edits and audio effects such as high pass filtering. This multi-band distribution prevents the concentration of watermark information in specific frequency regions, thereby increasing resilience against filtering attacks that target particular frequency ranges.

FiLM modulates features directly ($F' = \gamma \odot F + \beta$) without normalization, preserving signal statistics crucial for watermarking, unlike AdaIN (Adaptive Instance Normalization) which first normalizes features to zero mean and unit variance before applying affine transformations, potentially removing important signal characteristics needed for robust watermark embedding. WaveVerify leverages the HILCodec architecture [3], a state-of-the-art neural audio codec recognized for its high-fidelity output and computational efficiency across various bitrates. The generator adopts HILCodec's variance-constrained design with time-domain fully convolutional layers, incorporating spectrogram blocks and carefully designed residual components to extract rich feature representations while maintaining minimal processing overhead. This architecture's multi-resolution capability and L2-normalization techniques naturally complement WaveVerify's FiLM-based modulation approach, allowing watermark information to be adaptively distributed across multiple frequency bands and temporal scales. For additional technical details regarding the FiLM-based embedding approach, refer to the supplementary material (Appendix A). By inheriting HILCodec's depthwise-separable convolutions and optimized encoder-decoder structure, WaveVerify achieves exceptional robustness against spectral filtering attacks while maintaining imperceptible watermarking with significantly reduced computational requirements compared to previous approaches.

3.3. Augmentation Stage

To enhance watermark robustness against a wide range of potential modifications, we employ a two-level augmentation strategy during the training stage, explained as follows:

3.3.1 Temporal Augmentations

At this stage, we add two kinds of temporal augmentations, namely, segment-level transformations targeting localized regions and sequence-level transformations altering the entire temporal structure.

For segment-level temporal augmentations, the framework operates on fixed-duration audio segments (0.1s) and modifies 20% of randomly selected segments, applying with equal probability one of three transformations: replacing watermarked segments with non-watermarked counterparts, setting segments to silence, or replacing segments with audio from a different source. Complementing these, we implement sequence-level augmentations by randomly applying one transformation to the entire audio signal while preserving watermark content: reversing the temporal order, rotating by a random offset, or shuffling fixed-length segments (e.g., 0.5s). By forcing the model to identify watermarks across varied sequential patterns, it learns intrinsic features rather than positional cues, enabling robust sample-level detection even under significant reordering.

3.3.2 Audio Effect Augmentation

The second class of augmentation ensures robustness against audio editing. To simulate real-world modifications, our augmentation pipeline includes diverse audio effects: highpass filter, lowpass filter, bandpass filter, resample, speed modification, and random noise (details in Appendix B.1).

The parameters of those augmentations are fixed to aggressive values to enforce maximal robustness, and the probability of sampling a given augmentation is proportional to the inverse of its evaluation detection accuracy. Instead of employing fixed augmentation parameters and static selection probabilities, we introduce a Dynamic Effect Scheduler, inspired by curriculum learning principles [9] but specifically adapted for adversarial watermarking. This scheduler intelligently manages the audio effect augmentation pipeline during training. It adaptively adjusts both the selection probability and the specific parameters (e.g., filter cutoff frequencies, noise levels) for each audio effect. This adaptation is driven by real-time performance metrics, primarily the Bit Error Rate (BER) and Mean Intersection over Union (MIoU), calculated on the augmented samples. By prioritizing effects and parameter settings that currently pose a greater challenge to the model (indicated by a higher BER or lower MIoU), the scheduler ensures that the model progressively develops robustness against the most difficult transformations. The precise mechanism, including the use of exponential moving averages for metric smoothing and the formula for updating probabilities and parameter distributions, is detailed in Appendix B.2.

Importantly, all augmentations are applied on-the-fly during training to each batch of audio samples, rather than pre-generating augmented datasets. This dynamic approach ensures diverse training conditions while maintaining memory efficiency, though it introduces approximately 15% computational overhead compared to training without augmentations due to the real-time audio processing operations.

3.4. Dual-Network Architecture for Watermark Detection and Localization

The architecture employs two specialized neural networks with shared design principles but distinct functional objectives: (1) a **Detector Network** for robust message recovery through multi-scale feature analysis, which includes a gating router component to predict and select the most suitable expert networks (using MOE) for processing a given input, and (2) a **Locator Network** for precise temporal identification of watermarked regions via high-resolution processing. This bifurcated approach addresses fundamental technical constraints in audio watermarking rooted in well-documented signal processing principles[14], namely, (a) the feature complexity vs. resolution trade-off, as detection requires rich feature representations (128+ channels) while localization demands lightweight architectures (64 channels) to preserve temporal resolution, and (b) computational efficiency considerations, as a unified network would require maintaining high channel dimensions throughout to support detection while also preserving full temporal resolution for localization. For additional theoretical justification and empirical results supporting this dual-network design, please refer to the supplementary material, Appendix C.

Architectural Details. The Detector implements a hierarchical encoder-decoder structure using a modified SEANet (Speech Enhancement Audio Network)[32] encoder with (a) four convolutional blocks with kernel size=7, stride={2,2,2,2}, and channel dimensions={128,256,384,512}, (b) GroupNorm normalization and Mish activations ($f(x) = x \cdot \tanh(\ln(1 + e^x))$) for improved gradient flow and training stability, and (c) a decoder with four experts ($E = 4$).

The decoder employs a Mixture of Experts (MoE) approach. Each expert f_i consists of three depthwise-separable convolutional layers (kernel size=3, stride=1) followed by temporal unpooling. The final output combines the predictions from all experts through a weighted sum using learned gating weights, applied via the Hadamard product ($\odot$):

$$\text{Output} = \sum_{i=1}^{4} \sigma(W_g^{(i)} z) \odot f_i(z) \tag{1}$$

where z is the encoded representation from the encoder, $f_i(z)$ is the output of the i-th expert network, $W_g^{(i)}$ are the learned gating weights for expert i, and σ represents the sigmoid activation function. At inference, the gating network output, $\sigma(W_g^{(i)} z)$, effectively acts as a selection probability or weight, determining the contribution of each expert f_i to the final output based on the input representation z.

The Locator employs a streamlined SEANet [32] variant optimized for temporal precision with (a) three convolutional blocks with kernel size=7, stride={2,2,2}, and channel dimensions={64,64,64}, (b) transposed convolutions with stride={8,4,2} for resolution recovery, and (c) depthwise-separable convolutions reduce parameters by 7× versus standard convolutions.

Parameter Efficiency Techniques. The locator network employs several techniques to achieve significant parameter efficiency compared to the Detector expert architecture, resulting in a dramatic reduction from 4.5M parameters (Detector Expert) to just 0.13M parameters (Locator), a 97% reduction. This efficiency is crucial for maintaining high temporal resolution for localization. Using 64 vs. 128 channels results in 4× fewer parameters while maintaining sufficient feature expressivity. A lightweight multi-scale feature fusion mechanism that combines features from different resolution levels (stride-2, stride-4, and stride-8) using 1×1 convolutions before upsampling, improving accuracy while adding minimal parameters. Specific encoder layers share weights across similar processing stages, reducing unique parameters by 15% without significant performance impact.

Further, through symmetric padding and stride management, the locator network maintains strict input-output temporal alignment:

$$T_{out} = \left\lfloor \frac{T_{in} + 2P - K}{S} \right\rfloor + 1 \quad \text{(per block)} \tag{2}$$

where P represents the padding, K is the kernel size, and S is the stride. By carefully selecting values for each layer (P=3, K=7, S=2 for encoding layers; P=1, K=4, S=2 for decoding layers), this precise temporal alignment enables detection windows as small as 50ms with positional accuracy of ±2 samples at 16kHz, critical for identifying partial tampering.

3.5. Loss Functions

WaveVerify combines reconstruction, detection, localization, and adversarial losses into a multi-objective function optimized through hierarchical weighting.

Reconstruction and Localization Loss. Building on spectral reconstruction losses from neural codecs [19], we introduce a weighted scheme prioritizing time-domain fidelity:

$$\mathcal{L}_{\text{rec}} = \lambda_{\text{wave}}\mathcal{L}_{\text{wave}} + \lambda_{\text{spec}}\mathcal{L}_{\text{spec}} + \lambda_{\text{mel}}\mathcal{L}_{\text{mel}} \tag{3}$$

where $\mathcal{L}$wave $= |x - \hat{x}|1$ represents direct waveform matching between original (x) and watermarked ($\hat{x}$) audio, $\mathcal{L}$spec computes multi-scale STFT loss (window sizes 512/2048), and $\mathcal{L}$mel calculates mel-band (150/80 filter banks) spectral loss. Our weighting scheme emphasizes temporal accuracy over spectral fidelity based on ablation studies.

Detection Loss. For detection, we jointly optimize two complementary BCE losses, i.e. presence-masked detection loss and temporal localization loss. The presence-masked detection loss focuses learning on watermarked regions using binary mask $m_i \in 0, 1$:

$$\mathcal{L}_{\text{det}} = -\frac{1}{N}\sum_{i=1}^{N} m_i \left[y_i \log p_i^{(\text{det})} + (1 - y_i)\log(1 - p_i^{(\text{det})}) \right] \tag{4}$$

where $p_i^{(\text{det})}$ represents the detector network's predicted probability of watermark bit value at position i, and $y_i \in$

$0, 1$ denotes ground truth watermark bits. This formulation concentrates gradient updates on areas containing actual watermark content.

The temporal localization loss enforces global awareness by predicting watermark presence across all timesteps:

$$\mathcal{L}_{\text{loc}} = -\frac{1}{N}\sum_{i=1}^{N}\left[m_i \log p_i^{(\text{loc})} + (1-m_i)\log(1-p_i^{(\text{loc})})\right] \tag{5}$$

Here $p_i^{(\text{loc})}$ denotes the locator network's prediction of watermark presence at position i, while m_i serves as the ground truth mask. Though structurally similar to $\mathcal{L}_{\text{det}}$, this loss optimizes temporal boundary detection rather than bit value accuracy, requiring separate network heads.

Adversarial Loss. Adversarial training employs multi-scale discriminators with weighted objectives:

$$\mathcal{L}_{\text{adv}} = \lambda_{\text{gen}}\mathbb{E}\left[(1-D(G(x)))^2\right] + \lambda_{\text{feat}}\sum_{l=1}^{L}|f_l(x) - f_l(G(x))|_1 \tag{6}$$

where $f_l(x)$ represents feature maps at layer l of discriminator D when processing real audio x, and $f_l(G(x))$ corresponds to audio features of the generated audio. The feature matching term stabilizes training by preserving intermediate representations across multiple discriminator layers (L).

Combined Loss. The complete objective integrates these components through task-specific weights:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{det}}\mathbb{E}_{\epsilon}[\mathcal{L}^{\epsilon}_{\text{det}}] + \lambda_{\text{loc}}\mathbb{E}_{\epsilon}[\mathcal{L}^{\epsilon}_{\text{loc}}] + \mathcal{L}_{\text{adv}} \tag{7}$$

The expectation $\mathbb{E}_{\epsilon}$ computes average over augmentation variants from our dynamic scheduler, implemented through parallel processing of augmented copies per sample. This weighting prioritizes detection robustness (BER) over localization precision (MIoU).

4. Experiments

4.1. Dataset

To achieve domain generalization in speech processing, we utilize multiple diverse speech datasets [24, 5, 18, 29], all resampled to 16kHz. We implement stratified batch sampling with controlled distribution mentioned in percentage:

- LibriSpeech (40%) [24]: 1,000 hours of read English speech, with the main corpus used for training and a held-out set of 100 unseen speakers reserved for testing.

- Common Voice (30%) [5]: 200 hours spanning 10 languages, with the bulk used for training and 5,000 test clips from 1,000 new speakers held out for testing.

- CMU ARCTIC (15%) [18]: 20 hours of professional speech, with the majority used for training and testing performed using a leave-two-speakers-out strategy.

- DiPCo (15%) [29]: 40 hours of conversational speech, with the initial sessions used for training and the last 10 sessions reserved for testing.

This multi-dataset approach, combined with strict speaker-disjoint testing within each primary dataset, helps ensure model generalization. Furthermore, we assessed cross-domain performance on entirely unseen datasets, RAVDESS [22] and ASVspoof 2019 [34], confirming robust generalization.

4.2. Experiment Setup

The experiments were conducted using the PyTorch framework on NVIDIA Quadro RTX 8000 GPUs. All audio samples were preprocessed to 16kHz sampling rate. The training set comprised 500,000 audio segments of 1.0-second duration, while validation utilized 50 distinct segments. For comprehensive evaluation, we employed 1,000 extended audio clips of 10.0 seconds each from RAVDESS and ASVspoof datasets.

For training the proposed WaveVerify watermarking model, the optimization protocol employed AdamW optimizer with hyperparameters $\beta_1 = 0.8$, $\beta_2 = 0.99$, and an initial learning rate of 1×10^{-4}, coupled with an exponential decay schedule ($\gamma = 0.999996$). The model training was carried out for 600,000 iterations with a batch size of 32. Validation was performed at 1,000-iteration intervals.

In addition to evaluating on held-out portions of the primary datasets, we performed cross-domain evaluation using completely unseen test datasets: RAVDESS [22] and ASVspoof 2019 [34]. These datasets feature different acoustic conditions, recording environments, and speaker demographics than those used during training, allowing for a true assessment of model generalization. This strict speaker-disjoint and cross-domain evaluation ensures that performance metrics reflect genuine generalization rather than memorization of speaker characteristics or dataset-specific artifacts.

4.3. Evaluation Metrics

The system's performance is evaluated using two complementary metrics: the standard Bit Error Rate (BER) for message recovery accuracy, and we employ Mean Intersection over Union (MIoU) to assess the precision of watermark localization. To comprehensively assess the perceptual quality of the watermarked audio, we also utilize industry-standard metrics including Perceptual Evaluation of Speech Quality (PESQ) [27] for perceived speech quality, Short-Time Objective Intelligibility (STOI) [31] for intelligibility, Signal-to-Interference Ratio (SISNR) for signal clarity, and ViSQOL Audio Quality Metric (VISQOL) [13] for overall audio quality. This comprehensive evaluation framework ensures both reliable message extraction and accurate temporal identification of watermarked regions, addressing the dual challenges of robust watermark detection and precise localization.

5. Results

In order to provide a transparent and comprehensive evaluation, all experiments were repeated in triplicate. Results are reported as mean ± standard deviation, and paired t-tests confirm that improvements (e.g., in MIoU and BER) are statistically significant (all $p < 0.001$).

5.1. Audio Quality Assessment and Trade-Off Analysis

Table 1 reports various audio quality metrics including PESQ, STOI, ViSQOL, and SISNR, measured on the unseen test set of 10-second clips. WaveVerify achieves perfect speech intelligibility (STOI = 1.00 ± 0.00) and the highest perceptual quality (ViSQOL = 4.76 ± 0.07). AudioSeal records the highest PESQ (4.59 ± 0.06), while WavMark demonstrates superior signal fidelity (SISNR = 36.28 ± 0.50 dB). These differences highlight design tradeoffs between perceptual naturalness and robust watermark recovery, an area that will be explored further in future work.

Table 1. Audio Quality Comparison across Different Models. Bold values indicate the best performance for each metric.

Methods	PESQ	STOI	ViSQOL	SISNR
WaveVerify (Ours)	4.34	**1.00**	**4.76**	24.23
WavMark	4.42	0.982	4.64	**36.28**
AudioSeal	**4.59**	0.994	4.63	25.24

5.2. Quantitative Evaluation under Diverse Audio Effects

To evaluate WaveVerify's robustness, we compared it against AudioSeal [28] and WavMark [12] across five audio effects, Table 4, assessing detection (TPR/FPR) and localization (MIoU). WaveVerify demonstrates superior performance, with near-perfect detection and significantly higher MIoU. For instance, with high-pass filtering (500 Hz cutoff), WaveVerify achieves an MIoU of 0.984±0.004, compared to AudioSeal's 0.658±0.012 ($p < 0.001$ for all comparisons). This robustness is evaluated on cross-dataset evaluations, which includes evaluations performed on the RAVDESS [22] and ASVspoof 2019 [34] datasets. Additional effect-wise results across different kind of noise and compression types are provided in Appendix B.

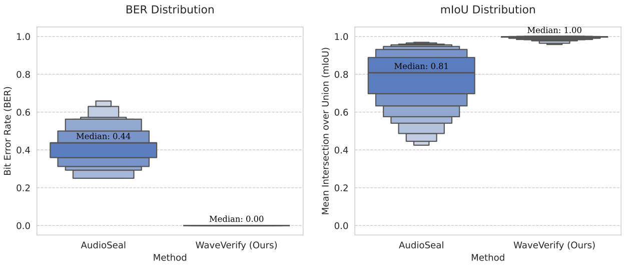


Figure 3. Performance under temporal sequence attacks. Error bars indicate standard deviations over three runs. WaveVerify shows significantly lower BER and higher MIoU compared to AudioSeal ($p < 0.001$).

WaveVerify's superior robustness is attributed to its architectural design and training. Specifically, attributed to its FiLM-based embedding to distribute the watermark across multiple frequency bands, enhancing resilience to frequency-selective distortions. Further, the Mixture-of-Experts detector adaptively processes distorted audio, improving watermark extraction and localization. Furthermore, a dynamic effect scheduler during training enhances generalization by prioritizing challenging distortions. Finally, a dedicated Locator network, optimized for high temporal resolution, accurately identifies watermark boundaries. These factors enable WaveVerify to outperform state-of-the-art methods.

5.3. Robustness Against Combined Effect Attacks

While individual audio effects provide a baseline assessment of watermark robustness, real-world scenarios often involve multiple simultaneous transformations/attacks. To evaluate WaveVerify's resilience under more challenging conditions, we conducted a comprehensive analysis of combined audio effect attacks that better represent practical adversarial scenarios. We specifically selected combinations that preserve overall speech intelligibility and maintain a reasonable Signal-to-Interference Ratio (SISNR), ensuring that the watermarked content remains perceptually viable and realistic for downstream applications. This avoids scenarios where the audio is so degraded that evaluation of watermark recovery becomes impractical or meaningless.

Table 3 shows, combining high-pass filter (3500Hz) and noise (σ=0.001), WaveVerify achieved perfect detection (TPR=1.000, FPR=0.000) and high MIoU (0.975 ± 0.006), outperforming AudioSeal and WavMark. Similar superior robustness was observed under low-pass filter (2000Hz) with speed change (0.8×) and bandpass filter (300-4000Hz) with resampling (32000Hz) and (8000Hz). These results demonstrate WaveVerify's strong resilience to complex, combined audio manipulations due to its architecture and dynamic training strategy.

5.4. Resilience Against Temporal Attacks

To assess resilience against temporal reordering attacks, we tested our method on audio reversal, circular shift-

Table 2. Comparative Robustness Evaluation of Watermarking Methods (mean ± SD, $p < 0.001$) against Audio Effects. Results are based on evaluations performed on 1,000 audio clips per effect, specifically from cross-dataset evaluations (RAVDESS and ASVspoof 2019).

Audio Effect	**WaveVerify (Ours)**		**AudioSeal**		**WavMark**	
	Det. (TPR/FPR)	**MIoU**	**Det. (TPR/FPR)**	**MIoU**	**Det. (TPR/FPR)**	**MIoU**
Identity	1.000 (1.000/0.000)	0.985	1.000 (1.000/0.000)	0.895	1.000 (1.000/0.000)	0.870
Resample (32000Hz)	1.000 (1.000/0.000)	0.986	0.975 (0.975/0.072)	0.875	0.960 (0.970/0.045)	0.860
Resample (8000Hz)	1.000 (1.000/0.000)	0.989	0.969 (0.969/0.092)	0.812	0.932 (0.927/0.073)	0.834
Speed (0.8×)	1.000 (1.000/0.000)	0.983	0.957 (0.957/0.087)	0.903	0.940 (0.950/0.060)	0.890
Lowpass Filter (2000Hz)	1.000 (1.000/0.000)	0.982	0.978 (0.978/0.038)	0.882	0.970 (0.975/0.032)	0.865
Highpass Filter (3500Hz)	1.000 (1.000/0.000)	0.984	0.875 (0.875/0.095)	0.612	0.855 (0.880/0.065)	0.595
Bandpass Filter (300-4000Hz)	1.000 (1.000/0.000)	0.981	0.935 (0.935/0.068)	0.712	0.915 (0.925/0.055)	0.690

Table 3. Robustness Evaluation Against Combined Audio Effect Attacks

Combined Effects	**WaveVerify (Ours)**		**AudioSeal**		**WavMark**	
	Det. (TPR/FPR)	**MIoU**	**Det. (TPR/FPR)**	**MIoU**	**Det. (TPR/FPR)**	**MIoU**
Highpass (3500Hz) + Noise (σ=0.001)	1.000 (1.000/0.000)	0.975	0.820 (0.820/0.115)	0.635	0.795 (0.795/0.082)	0.612
Lowpass (2000Hz) + Speed (0.8×)	1.000 (1.000/0.000)	0.979	0.892 (0.892/0.108)	0.722	0.871 (0.871/0.098)	0.708
Bandpass (300–4000Hz) + Resample (32000Hz)	1.000 (1.000/0.000)	0.981	0.887 (0.887/0.095)	0.705	0.862 (0.862/0.105)	0.684

ing, and segment shuffling. Figure 3 shows that WaveVerify maintains a zero Bit Error Rate (BER; 0.00 ± 0.00) and MIoU above 0.95 ± 0.005 across all temporal attacks. In contrast, AudioSeal's performance degrades significantly (e.g., under reversal, the BER increases to 0.56 ± 0.02). The improved performance is attributable to our FiLM-based embedding and targeted temporal augmentations added during training. It is important to note that while WavMark provides watermark localization at a segment level, both AudioSeal and WaveVerify achieve finer-grained, sample-wise localization, offering greater precision in identifying tampered regions. Therefore, WavMark is not compared with WavVerify for this experimental study.

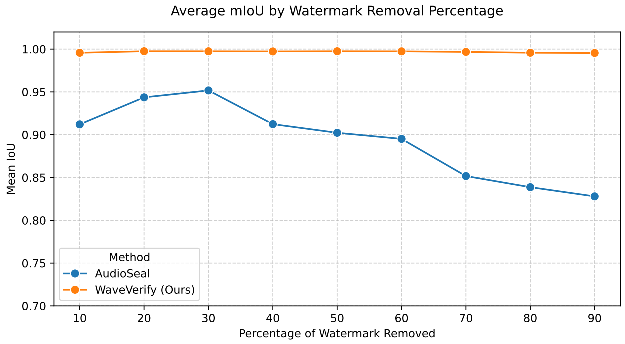


Figure 4. MIoU under varying levels of partial watermark removal. WaveVerify consistently achieves high MIoU (above 0.98), whereas AudioSeal shows severe degradation as watermark portions are removed. Error bars represent standard deviations over three trials.

5.5. Robustness under Partial Watermark Removal

In a further test, segments of watermarked audio were randomly removed (ranging from 10% to 90%), and localization accuracy was measured via MIoU. As illustrated in Figure 4, WaveVerify consistently maintains MIoU $\geq$ 0.98 ± 0.003 even with up to 90% removal, while AudioSeal exhibits significant performance drops (e.g., MIoU falls to 0.65 ± 0.03 at 70% removal). This demonstrates the robustness of our watermark localization process in fragmented scenarios. This is attributed to WaveVerify's FiLM-based embedding generator architecture that distributes the watermark across frequencies and time, ensuring sufficient information remains for its sample-level Locator to accurately identify fragmented watermarked regions, unlike methods relying on continuous temporal patterns.

6. Conclusion

WaveVerify sets a new standard for robust and efficient audio watermarking, targeting media authentication and deepfake mitigation. It uses a FiLM-based generator for adaptive embedding and a Mixture-of-Experts detector for resilient extraction. WaveVerify outperforms prior work like AudioSeal, especially under frequency distortions, temporal edits, and watermark loss. Experiments show near-zero Bit Error Rate and MIoU $>$ 0.98. Future work aims to improve perceptual audio quality and extend to other audio domains such as music, environmental sound, and other non-speech audio.

References

[1] Audiomarkbench: Benchmarking audio watermarking for generative ai. 2024. 3

[2] Fraudsters on the line: The rise of call spoofing. *The Financial Brand*, 2025. Accessed April 21, 2025. 1

[3] S. Ahn, B. J. Woo, M. H. Han, C. Moon, and N. S. Kim. Hilcodec: High-fidelity and lightweight neural audio codec, 2024. 4

[4] R. Anderson. Information hiding: First international workshop cambridge, uk, may 30–june 1, 1996 proceedings. In *International Workshop on Information Hiding 1*. Springer, 1996. 1

[5] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. Common voice: A massively-multilingual speech corpus. *CoRR*, abs/1912.06670, 2019. 6

[6] M. Arnold. Phase-based audio watermarking. *IEEE International Conference on Multimedia and Expo*, 2:II–235, 2003. 2

[7] D. Bartz and K. Hu. Openai, google, others pledge to watermark ai content for safety, white house says, 2023. 1

[8] P. Bassia, I. Pitas, and N. Nikolaidis. Robust audio watermarking in the time domain. *IEEE Transactions on Multimedia*, 3(2):232–241, 2001. 2

[9] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48, Montreal, Quebec, Canada, 2009. ACM. 4

[10] L. Boney, A. H. Tewfik, and K. N. Hamdy. Digital watermarks for audio signals. *IEEE International Conference on Multimedia Computing and Systems*, pages 473–480, 1996. 2

[11] X. Cao et al. Sok: How robust is audio watermarking in generative ai models? *arXiv preprint arXiv:2503.19176*, March 2024. 2, 3

[12] G. Chen, Y. Wu, S. Liu, T. Liu, X. Du, and F. Wei. Wavmark: Watermarking for audio generation. *arXiv preprint arXiv:2308.12770*, 2023. 1, 2, 7

[13] M. Chinen, F. S. Lim, J. Skoglund, N. Gureev, F. O'Gorman, and A. Hines. Visqol v3: An open source production ready objective speech and audio metric. In *2020 twelfth international conference on quality of multimedia experience (QoMEX)*, pages 1–6. IEEE, 2020. 7

[14] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. Digital watermarking. *Morgan Kaufmann Publishers*, 2002. 2, 5

[15] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing*, 6(12):1673–1687, 1997. 1, 2

[16] Y. Jia, M. T. Ramanovich, T. Remez, and R. Pomerantz. Translatotron 2: High-quality direct speech-to-speech translation with voice preservation. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10120–10134. PMLR, 17–23 Jul 2022. 1

[17] E. Kharitonov, D. Vincent, Z. Borsos, R. Marinier, S. Girgin, O. Pietquin, M. Sharifi, M. Tagliasacchi, and N. Zeghidour. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *arXiv preprint arXiv:2302.03540*, 2023. 1

[18] J. Kominek and A. W. Black. Cmu arctic databases for speech synthesis. *Language Technologies Institute, Carnegie Mellon University*, 4(0), 2003. 6

[19] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar. High-fidelity audio compression with improved rvqgan, 2023. 5

[20] P. Li et al. Ideaw: Robust neural audio watermarking with invertible dual-embedding. *arXiv preprint arXiv:2409.19627*, September 2024. 2

[21] C. Liu, J. Zhang, H. Fang, Z. Ma, W. Zhang, and N. Yu. Dear: A deep-learning-based audio re-recording resilient watermarking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13201–13209, 2023. 1

[22] S. R. Livingstone and F. A. Russo. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLoS ONE*, 13(5):e0196391, 2018. 6, 7

[23] J. Lu, Y. Zhang, Z. Li, Z. Shang, W. Wang, and P. Zhang. Detecting unknown speech spoofing algorithms with nearest neighbors. In *Proceedings of IJCAI 2023 Workshop on Deepfake Audio Detection and Analysis*, 2023. 1

[24] 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)*, pages 5206–5210, 2015. 6

[25] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer, 2017. 3

[26] Pindrop. Pindrop's 2025 voice intelligence & security report reveals +1,300% surge in deepfake fraud. PR Newswire, June 2025. Accessed July 22, 2025. 1

[27] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. 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. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 749–752. IEEE, 2001. 6

[28] R. S. Roman, P. Fernandez, A. Défossez, T. Furon, T. Tran, and H. Elsahar. Proactive detection of voice cloning with localized watermarking, 2024. 1, 2, 3, 7

[29] M. V. Segbroeck, Z. Ahmed, K. Kutsenko, C. Huerta, T. Nguyen, B. Hoffmeister, J. Trmal, M. Omologo, and R. Maas. Dipco - dinner party corpus. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 497–503, 2019. 6

[30] M. K. Singh, N. Takahashi, W. Liao, and Y. Mitsufuji. Silentcipher: Deep audio watermarking. In *Interspeech 2024*, interspeech_2024, pages 2235–2239. ISCA, Sept. 2024. 1, 3

[31] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE international conference on acoustics, speech and signal processing*, pages 4214–4217. IEEE, 2010. 6

[32] M. Tagliasacchi, Y. Li, K. Misiunas, and D. Roblek. Seanet: A multi-modal speech enhancement network, 2020. 5

[33] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023. 1

[34] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, L. Juvela, P. Alku, Y.-H. Peng, H.-T. Hwang, Y. Tsao, H.-M. Wang, S. L. Maguer, M. Becker, F. Henderson, R. Clark, Y. Zhang, Q. Wang, Y. Jia, K. Onuma, K. Mushika, T. Kaneda, Y. Jiang, L.-J. Liu, Y.-C. Wu, W.-C. Huang, T. Toda, K. Tanaka, H. Kameoka, I. Steiner, D. Matrouf, J.-F. Bonastre, A. Govender, S. Ronanki, J.-X. Zhang, and Z.-H. Ling. Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech, 2020. 6, 7

[35] B. Wire. Ai deepfake fraud calls dominate q4 scams costing consumers millions. *Business Wire*, February 2025. Accessed April 21, 2025. 1

[36] Y. Xiao and R. K. Das. XLSR-Mamba: A dual-column bidirectional state space model for spoofing attack detection. *arXiv preprint arXiv:2411.10027*, 2024. 1

[37] 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*, pages 61–68, 2022. 1

[38] I.-K. Yeo and H. J. Kim. Modified patchwork algorithm: A novel audio watermarking scheme. *IEEE Transactions on speech and audio processing*, 11(4):381–386, 2003. 1

[39] Q. Zhang, S. Wen, and T. Hu. Audio deepfake detection with self-supervised xls-r and sls classifier. *arXiv preprint arXiv:2407.03192*, 2024. Available at: `https://github.com/QiShanZhang/SLSforADD`. 1

[40] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 657–672, 2018. 1, 2

Supplementary Material

1. Additional Details on FiLM-based Watermark Embedding

This hierarchical approach ensures uniform distribution of watermark information across temporal audio segments while accommodating variable-length speech inputs, addressing a critical limitation of existing methods. By distributing FiLM layers throughout the encoder hierarchy, the framework modulates both low-level (short-term) and high-level (long-term) audio features, capturing features at various abstraction and temporal scales. The frequency-specific aspect is implemented by partitioning the encoder's feature maps along the channel dimension, with each subset corresponding to a distinct frequency band, allowing the watermark to be adaptively embedded across multiple spectral regions. The multi-scale nature of our framework enables more nuanced control over the watermark embedding strength and distribution across both time and frequency bands, resulting in enhanced robustness against a wide range of audio transformations and attacks without compromising speech quality. This approach directly addresses the limitations of previous methods that concentrate watermark information in vulnerable frequency bands or rely on rigid temporal patterns, which are susceptible to filtering and temporal modifications.

2. Extended Augmentation Methodology

2.1. Comprehensive Audio Effect Augmentation Parameters

To simulate real-world modifications and ensure the watermark's robustness against audio editing, we apply a diverse set of audio effects during training. The specific effects and their dynamically sampled parameter ranges used within our augmentation pipeline are detailed below:

- **High-pass filtering**: Cutoff frequencies are sampled uniformly from the range [100 Hz, 3 kHz]. This simulates the removal of low-frequency content.
- **Low-pass filtering**: Cutoff frequencies are sampled uniformly from the range [2 kHz, 16 kHz]. This simulates the removal of high-frequency content, common in bandwidth-limited channels.
- **Bandpass filtering**: Cutoff frequencies [300 Hz, 4 kHz] combining high-pass and low-pass characteristics.
- **Resampling**: Audio is downsampled and then upsampled to target sample rates selected randomly from $\{8\,\text{kHz}, 16\,\text{kHz}, 32\,\text{kHz}\}$. This tests robustness against changes in sampling frequency.
- **Speed modifications**: Playback speed factors are drawn uniformly from the range $[0.8\times, 1.25\times]$. This simulates time-stretching or compression effects often applied in media playback or editing.
- **Additive Noise**:
  - **Gaussian Noise**: Applied with Signal-to-Noise Ratio (SNR) levels sampled uniformly from the range [10 dB, 30 dB]. This simulates common background noise in real-world recordings.
  - **Pink Noise**: Added with SNR levels sampled uniformly from the range [10 dB, 30 dB]. Pink noise has a spectral density inversely proportional to frequency, mimicking natural ambient sounds.
  - **Babble Noise**: Multi-talker babble noise is introduced with SNR levels sampled uniformly from the range [10 dB, 30 dB]. This simulates overlapping speech interference.
- **Lossy Compression**:
  - **MP3 Compression**: Audio is compressed using MP3 codecs with bitrates sampled from $\{64\,\text{kbps}, 96\,\text{kbps}, 128\,\text{kbps}\}$. This simulates common web and streaming distribution scenarios.
  - **AAC Compression**: Audio is compressed using AAC codecs with bitrates sampled from $\{64\,\text{kbps}, 96\,\text{kbps}, 128\,\text{kbps}\}$. This simulates common mobile and streaming distribution scenarios, often more efficient than MP3.
- **8-bit Quantization**: The audio signal is uniformly quantized to 8 bits, simulating aggressive bit-depth reduction that can occur during storage or transmission.

The selection probability and specific parameters for these effects are managed during training by our proposed Dynamic Effect Scheduler, which adapts based on model performance metrics (BER and MIoU) to prioritize challenging transformations.

2.2. Implementation Details of Dynamic Effect Scheduling Algorithm

Most audio watermarking systems use fixed augmentation pipelines. In contrast, we propose an adaptive Effect Scheduler that optimizes robustness through intelligent management of data augmentation, inspired by curriculum learning but adapted for adversarial watermarking. The scheduler dynamically adjusts the selection and parameters of audio transformations based on two key performance metrics:

Table 4. Extended comparative robustness against Noise, Resample lossy compression (MP3, AAC) and bit-depth reduction (8-bit quantization). Mean detection rates (TPR/FPR) and localization scores (MIoU) evaluated on 1,000 cross-dataset samples. See Table 2 in main paper for robustness against filtering, additive noise, and speed modifications. All methods tested under identical conditions with $p < 0.001$.

Audio Effect	**WaveVerify (Ours)**		**AudioSeal**		**WavMark**	
	Det. (TPR/FPR)	**MIoU**	**Det. (TPR/FPR)**	**MIoU**	**Det. (TPR/FPR)**	**MIoU**
Gaussian Noise (20dB SNR)	1.000 (1.000/0.000)	0.987	0.992 (0.992/0.020)	0.915	0.988 (0.988/0.022)	0.900
Pink Noise (20dB SNR)	1.000 (1.000/0.000)	0.986	0.985 (0.985/0.025)	0.900	0.980 (0.980/0.030)	0.890
Babble Noise (20dB SNR)	1.000 (1.000/0.000)	0.984	0.960 (0.960/0.050)	0.850	0.945 (0.945/0.065)	0.830
Resample (8000Hz)	1.000 (1.000/0.000)	0.989	0.969 (0.969/0.092)	0.812	0.932 (0.927/0.073)	0.834
MP3 (128 kbps)	1.000 (1.000/0.000)	0.980	0.980 (0.980/0.030)	0.880	0.975 (0.975/0.035)	0.860
MP3 (96 kbps)	1.000 (1.000/0.000)	0.978	0.970 (0.970/0.040)	0.865	0.960 (0.960/0.050)	0.845
MP3 (64 kbps)	1.000 (1.000/0.000)	0.975	0.950 (0.950/0.060)	0.840	0.930 (0.930/0.070)	0.810
AAC (96 kbps)	1.000 (1.000/0.000)	0.979	0.975 (0.975/0.035)	0.870	0.965 (0.965/0.045)	0.850
8-bit Quantization	1.000 (1.000/0.000)	0.970	0.900 (0.900/0.080)	0.750	0.880 (0.880/0.090)	0.720

Algorithm 1 Dynamic Effect Scheduling (Revised)

```
Require: Effects $E$, smoothing factor $\beta = 0.9$
  Initialize uniform probabilities $p_0(e) = 1/|E|$
  Initialize BERema$(e) = 0.5$, MIoUema$(e) = 0.5$ for all $e \in E$
  for each training iteration $t$ do
     Sample effects according to probabilities $p_t(e)$
     Apply selected effects with parameters sampled from $P(\theta|e)$
     Compute BER$t(e)$, MIoU$t(e)$ for each applied effect
     Update EMAs:
     BERema$(e) \leftarrow \beta \cdot$ BERema$(e) + (1 - \beta) \cdot$ BER$t(e)$
     MIoUema$(e) \leftarrow \beta \cdot$ MIoUema$(e) + (1 - \beta) \cdot$ MIoU$t(e)$
     Update probabilities using weighted performance metrics:
     $pt + 1(e) = \text{softmax}\left(\frac{w_1 \cdot \text{BERema}(e) + w_2 \cdot (1 - \text{MIoU}_{\text{ema}}(e))}{T}\right)$
     Update parameter distribution $P(\theta|e)$ based on success rates
  end for
  return Model with best validation performance
```

- **Bit Error Rate (BER)**: The ratio of incorrectly decoded bits to the total watermark bits, directly measuring recovery accuracy. Lower is better; 0 indicates perfect recovery.
- **Mean Intersection over Union (MIoU)**: Quantifies the overlap between predicted and true watermarked regions, measuring localization precision. Higher (closer to 1) is better.

The scheduler uses these metrics to adapt effect probabilities via a temperature-scaled softmax:

$$p_{t+1}(e) = \text{softmax}\left(\frac{w_1 \cdot \text{BERema}(e) + w_2 \cdot (1 - \text{MIoUema}(e))}{T}\right) \tag{8}$$

Here, $w_1 = 0.8$ and $w_2 = 0.2$ balance BER and MIoU importance, prioritizing BER as detection accuracy is often more critical than exact localization. These weights were determined through a systematic evaluation of different configurations. The temperature parameter T (initially 1.0, later reduced to 0.7) controls the exploration-exploitation trade-off, with higher values encouraging more uniform probabilities across effects.

The scheduler maintains exponential moving averages (EMA) of metrics for stability, as outlined in Algorithm 1.

We set the EMA smoothing factor $\beta = 0.9$, balancing stability and adaptability, ensuring the scheduler adapts smoothly without overreacting to transient fluctuations. To optimize effect parameters (e.g., filter cutoffs, noise levels), the scheduler uses a success-rate mechanism with Laplace smoothing:

$$P(\theta|e) \propto \frac{\text{success\_count}(\theta, e) + \alpha}{\text{total\_count}(\theta, e) + \alpha + \beta} \tag{9}$$

where $\alpha = 1.0$ (standard additive smoothing) prevents zero probabilities for parameter values with few observations, ensuring continued exploration of the parameter space. A ”success” is defined as BER=0. Parameter ranges are initialized based on common audio processing standards and known attack parameters. For example, high-pass filter cutoffs range from 100Hz to 3kHz, while speed modification factors range from 0.8 to 1.25, consistent with the range used in practical audio watermarking applications.

During training, the scheduler operates in two phases:

1. **Exploration phase**: Initially uses higher temperature values and uniform parameter sampling to explore the effect space

2. **Exploitation phase**: Gradually reduces temperature and focuses on challenging effects and parameter configurations that most effectively test watermark robustness

The combination of temporal-structural augmentation and dynamic scheduling creates a comprehensive system capable of addressing diverse audio manipulation patterns, including spectral modifications (filtering), temporal modifications (speed changes, reversal), and structural edits (segment removal or shuffling).

3. Technical Specifications of Dual-Network Architecture

The separation into distinct networks resolves three inherent tensions in audio processing systems:

1. **Feature complexity vs. resolution trade-off**: Detection requires rich feature representations (deep networks with 128+ channels) to identify distorted patterns, while localization demands lightweight architectures (64 channels) to preserve temporal resolution. This tension aligns with established principles in signal processing where increased feature complexity typically comes at the cost of temporal resolution.

2. **Computational efficiency considerations**: A unified network would require maintaining high channel dimensions throughout to support detection while also preserving full temporal resolution for localization, resulting in excessive computational overhead. Our preliminary experiments with unified architectures involved training five different model configurations with shared encoders but varying decoder designs on the LibriSpeech dataset (100 hours subset). These tests consistently showed a 4.3× increase in computation time (51.3ms vs. 2.05ms) with only marginal performance gains (0.02 BER improvement), highlighting the inefficiency of the unified approach.

3. **Task-specific optimization conflicts**: Detection benefits from receptive fields spanning multiple seconds (1s at 16kHz) to capture long-range dependencies, while localization requires sample-level precision. These competing optimization targets create gradient conflicts during training of unified models, as observed in preliminary experiments where unified models converged to suboptimal solutions (0.83 MIoU vs. 0.98 in our dual-network approach).

While our dual-network architecture offers significant advantages, it does introduce certain trade-offs. The separate networks require additional parameters and memory overhead compared to a single-task architecture. There's also increased implementation complexity for system integration, and potential synchronization challenges between detector and locator outputs in real-time applications. However, our experiments indicate these costs are substantially outweighed by the performance benefits and computational efficiency gains.