Accepted by the 48th IEEE Symposium on Security and Privacy (IEEE S&P 2027)

ScrambleMark: Zero-Query Watermark-Agnostic Black-Box Evasion of Modern Speech Watermarking

Binhao Ma[§], Weikang Ding[§], Hanqing Guo[†], Wenwei Zhao[‡], Xiaowen Li[‡]
Leah Ding[*], Zhuo Lu[‡], Rui Duan[§]
[§]*University of Missouri–Kansas City* [†]*Indiana University Bloomington*
[‡]*University of South Florida* [*]*American University*

***Abstract*—Speech watermarking is becoming a key defense for digital copyright and content authentication. Existing attacks treat evasion as a removal problem: they erase the watermark, relying on detector access or on speech data that is hard to collect, such as paired clean–watermarked or clean target speaker speech. Commercial platforms are moving to close off these assumptions. TrustedAudio, for instance, restricts detection to authenticated users and denies meaningful detection feedback to unauthorized parties. Adversaries thus cannot tell whether a sample is watermarked or which watermark it carries, nor reliably remove it without degrading speech quality.**

In this work, we show that evading a speech watermark does not require erasing it: detection relies on the structural consistency of the embedded pattern, and disrupting this consistency is enough to defeat it. Building on this observation, we identify harmonic structure as an effective attack surface, as perturbing it breaks watermark consistency while preserving perceptual quality. We therefore propose ScrambleMark, a zero-query, watermark-agnostic, black-box attack that evades detection through harmonic scrambling. ScrambleMark combines a deterministic engine for discovering transferable harmonic perturbations with a generative model that learns randomized scrambling patterns, avoiding reliance on fixed attack rules. Across three commercial platforms (Google SynthID, Resemble AI, and TrustedAudio), ScrambleMark is the only attack to both evade detection and keep speech natural, achieving nearly 97% evasion and a 4.2/5 human rating, far above the 1.1 to 1.7 of prior attacks. Beyond evasion, ScrambleMark withstands adaptive recovery designed to reverse it, evidence that structural evasion is hard to mitigate in practice.

1. Introduction

Driven by recent advances in deep neural networks (DNNs) [1], [2], [3], [4], [5], audio watermarking has become a widely adopted technique for establishing provenance and protecting copyright in digital media [6], [7], [8]. Among its applications, speech has emerged as a particularly important domain due to the rapid progress of voice cloning and speech synthesis systems, which enable adversaries to impersonate individuals [9], commit fraud [10], and spread disinformation [11]. To address these threats, modern speech watermarking systems embed imperceptible marks into audio that can later be verified to establish provenance and detect misuse [3], [4], [5], [12], [13]. However, despite these advances, watermarking systems remain vulnerable to evasion attacks [14], [15], in which an adversary manipulates a watermarked speech signal to render the embedded watermark undetectable. Such attacks undermine the reliability of watermark-based provenance and misuse detection.

The first wave of studies on watermark evasion attacks has shown that audio watermarking fails under a range of adversarial capabilities, each assuming a different form of access. With query access to the detector, an adversary can iteratively optimize noise perturbations that suppress the watermark signal until detection fails [14]. When the embedding process is exposed, the adversary can instead inject a secondary watermark that overwrites the original [16]. Given knowledge of the watermark pattern, the watermark can be treated as structured noise and removed by a trained enhancement model [15].

However, existing attacks are impractical in realistic deployment settings. Most rely on one or more strong assumptions: access to detector feedback [14], [16], paired clean–watermarked data [15], or clean target-speaker speech [17]. In practice, an adversary typically cannot tell whether a speech sample is watermarked, which scheme was used, or which detector verifies it, and so lacks the feedback and side information these attacks require. While an attacker could, theoretically, brute-force query a collection of public watermark detectors to guess which one applies, the commercial watermarking platform, TrustedAudio [18], is specifically designed to prevent such probing. Operationally, detection interfaces are account-scoped: only authenticated users can retrieve detection outcomes, whereas adversaries are denied informative feedback (e.g., receiving "not detected" responses) even when submitting speech that contains a legitimate user's watermark. Therefore, access-dependent strategies become largely ineffective in the real world.

These constraints lead us to challenge an assumption shared by prior attacks: that successful evasion must erase the watermark. We therefore examine what modern water-

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

mark detectors actually rely on. To remain detectable under common distortions such as compression and noise, watermarking systems do not embed information at a single location; they spread it into a structured pattern across the time-frequency representation, and detection commonly works by matching this pattern. This exposes a new attack surface: because the watermark's robustness lies in this structural consistency, an adversary can defeat detection by disrupting that consistency rather than removing the watermark, so that the detector fails to match the pattern.

To bypass these real-world constraints, we propose ScrambleMark, a novel zero-query, watermark-agnostic, black-box attack that scrambles the harmonic structure of speech to disrupt the consistency required for watermark verification. Building such an attack under a strict black-box setting raises two challenges. (i) The attacker has no knowledge of where the watermark resides, and modern watermarking schemes often distribute watermark information across broad regions of the spectrum. Effective evasion therefore requires perturbations that generalize across the signal while preserving perceptual quality. (ii) The most straightforward way to disrupt watermark structure is through deterministic signal-processing transforms. However, fixed scrambling rules leave predictable perturbation patterns that an adaptive defender can estimate and partially reverse. Robust evasion therefore also requires randomized scrambling that avoids reliance on any fixed transformation.

To systematically resolve these challenges, ScrambleMark introduces a multi-stage synthesis framework. We first formalize the Evasion-Quality Trade-off Ratio (EQTR) across diverse perceptual feature domains, finding that harmonic-axis warping provides the best balance between disrupting watermark structure and preserving speech intelligibility. Building on this insight, we construct a surrogate-guided deterministic partition-shift engine that generates transferable, recovery-aware scrambling examples for training. Finally, to eliminate mathematical invertibility, we lift these deterministic operators into a conditional diffusion model governed by randomized target supervision and dynamic temporal scrambling. Through a pretraining and fine-tuning alignment scheme, ScrambleMark learns to confine its randomized perturbations exclusively to the harmonic dimension. This design lets the model generate natural speech on unseen watermarked speech while executing a recovery-resistant structural evasion attack.

We comprehensively evaluate ScrambleMark against seven state-of-the-art academic watermarking systems and three commercial platforms (Google SynthID, TrustedAudio, and Resemble AI). On commercial platforms, ScrambleMark achieves an average attack success rate (ASR) of nearly 97% while maintaining a mean human rating of 4.2/5, substantially outperforming existing watermark removal attacks. Notably, even when the defender knows the attack algorithm, recovery reduces ScrambleMark's ASR by only 0.02. Audio samples for ScrambleMark and the baseline attacks are available on the project website.

Our contributions are summarized as follows:

- **Structural Evasion.** We introduce *structural evasion*, a new attack paradigm that defeats robust watermarking schemes by disrupting watermark structural consistency under a strict zero-query, black-box setting. We design ScrambleMark, a conditional diffusion framework that implements this paradigm through randomized target supervision and dynamic temporal scrambling, avoiding fixed recovery paths for defenders.
- **Real-world Impact.** We present the first evasion study of commercial speech watermarking platforms, including Google SynthID, TrustedAudio, and Resemble AI. We further conduct human listening studies to evaluate perceptual quality and demonstrate the practical impact of ScrambleMark in real-world settings.
- **Defense Resilience.** We further evaluate ScrambleMark against two adaptive recovery defenses and show that it remains effective even when the defender knows the attack algorithm and has access to the original speech.

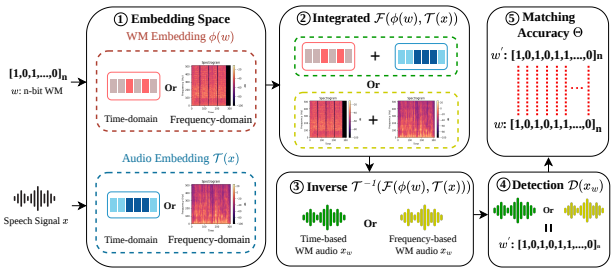


Figure 1: Overview of the watermarking pipeline.

2. Background and Design Intuition

We first introduce speech watermarking background, then present our threat model, review existing evasion attacks, and describe the design intuition and system overview.

2.1. Speech Watermarking

Speech watermarking is a key mechanism for copyright tracing, content attribution, and identity protection [6], [8]. While prior work has proposed academic systems such as AudioSeal [5] and Timbre [4], commercial adoption is rapidly accelerating. Google SynthID [19], Resemble AI [20], and TrustedAudio [18] have already deployed speech watermarking in real-world services.

Watermarking Pipeline. As illustrated in Fig. 1, the watermarking framework is composed of two primary stages: watermark embedding (steps ① to ③) and detection (steps ④ to ⑤). The details of each step are described as follows:

Watermarking Embedding. The speech watermarking embedding takes two inputs: a speech signal x and an n-bit watermark $w = \{w_i\}_{i=1}^{n}$, $w_i \in \{-1, 1\}$ [4]. The embedding function $\mathcal{E}$ maps (x, w) to the watermarked signal x_w as $x_w = \mathcal{E}(x, w) = \mathcal{T}^{-1}(\mathcal{F}(\mathcal{T}(x), \phi(w)))$.

① Input transformation. The speech signal x is mapped to an embedding space via $\mathcal{T}(x)$, chosen as either the identity

function for time-domain embedding [21], [22] or the Short-Time Fourier Transform (STFT) as a frequency-domain embedding [23], [24]. In parallel, the watermark vector w is mapped by an embedding function $\phi(\cdot)$, such as a neural encoder [5], into a structured pattern $\phi(w)$ that lies in the same domain as $\mathcal{T}(x)$ for subsequent integration [4].

② Integration. The transformed audio $\mathcal{T}(x)$ and the embedded watermark pattern $\phi(w)$ are integrated via a fusion function $\mathcal{F}(\cdot)$ to form a unified latent representation [3], [5].

③ Inverse transformation. Finally, the watermarked signal x_w is reconstructed by applying the inverse transformation $\mathcal{T}^{-1}(\cdot)$, e.g., inverse STFT, recovering a time-domain signal.

Watermarking Detection. The detector [4] decides whether the received speech signal is watermarked by estimating the embedded pattern in two steps (Fig. 1):

④ Watermark extraction. Given a potentially watermarked signal x_w, the detection function $\mathcal{D}(\cdot)$ outputs an estimated bit sequence $w' = \mathcal{D}(x_w) = \{w'_i\}_{i=1}^n$, where $w'_i \in \{-1, 1\}$ denotes the i-th predicted bit.

⑤ Matching score. The predicted sequence w' is evaluated against the watermark w using the bitwise matching score $\Theta(w, w') = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(w_i = w'_i)$, where $\mathbb{I}(\cdot)$ denotes the indicator function, returning 1 if the condition is true and 0 otherwise.

Watermarking decision. The detector compares the bitwise matching accuracy with a threshold κ and outputs $y \in \{0, 1\}$ as $y = 1$ if $\Theta(w, w') > \kappa$, else 0, where $y = 1$ denotes watermark presence and $y = 0$ absence. The threshold κ controls the robustness–reliability trade-off [5], [25], [26].

2.2. Threat Model

Attack Scenario. We consider a scenario in which an adversary attempts to evade detection on watermarked speech to enable downstream misuse, such as unauthorized voice cloning [4], [13] or illegal redistribution. The watermark is imperceptible, meaning the attacker is unaware of whether a given audio sample is watermarked, and therefore attempts to disrupt any potential watermark before misuse.

Attack Goal. To enable downstream misuse of the audio, the adversary is given a watermarked audio x_{w_t} carrying an unknown target watermark message w_t, and produces a perturbed audio x' such that (i) x' evades watermark detection, and (ii) x' remains perceptually close to x_{w_t}.

Adversary Capabilities. We adopt a strict zero-query black-box model in which the adversary is constrained along three dimensions: (i) No watermarking system knowledge: the adversary has no prior knowledge of the target watermarking system, including its embedding and detection mechanisms. (ii) No query access: the adversary cannot interact with the target watermarking system and receives no feedback during the attack process. Specifically, the target watermarking system is accessed only beforehand to obtain the watermarked audio x_{w_t} as the attack input; once x_{w_t} is obtained, zero queries are made to the target detector throughout the attack. (iii) No access to clean or watermarked dataset: the adversary only has access to the single target watermarked speech x_{w_t} to be attacked, and no access to any additional clean speech x or watermarked data from the target speaker. As a result, the attacker cannot exploit speaker-specific features such as timbre. Instead, the adversary relies on a publicly available surrogate watermarking model, which is not assumed to match the target system and is used only to learn transferable perturbations δ. Therefore, our zero-query constraint applies specifically to the target detector during the attack, while allowing access to the surrogate model.

TABLE 1: Adversary's knowledge and evasion scope.

Method	System Access		Data Dependencies		Evasion Scope	
	WM Pipeline	Query	Paired Data	Target Data	Academic	Commercial
White-box Noise [14]	✓	✓	✗	✗	✓	✗
Black-box Noise [14]	✗	✓	✗	✗	✓	✗
Overwrite (query) [16]	✗	✓	✗	✗	✓	✗
Overwrite$_{NQ}$ [16]	✗	✗	✗	✗	✓	✗
SE Attack [15]	✗	✗	✓	✓	✓	✗
HarmonicAttack [30]	✗	✗	✓	✓	✓	✗
DiffErase [17]	✗	✗	✗	△	✓	✗
Encodec [31]	✗	✗	✗	△	✓	✗
ScrambleMark	✗	✗	✗	✗	✓	✓

(i) **WM Pipeline**: knowledge of the watermark embedding and detection pipeline [14]; (ii) **Query**: query access to the watermark detector [32], [33], and Overwrite$_{NQ}$ denotes the non-query variants of [16]; (iii) **Paired Data**: paired clean and watermarked speech [15], [30]; (iv) **Target Data**: clean target-speaker speech samples. DiffErase [17] and recent works [17], [31], [34], [35] do not clarify whether the pretrained generative model has been exposed to the clean target speaker's speech. We denote this assumption as △ to reflect the ambiguity, and revisit it empirically in Section 6; (v) **Evasion Scope**: evaluated on **Academic** schemes and/or **Commercial** platforms.

Defender's Knowledge and Capabilities. The defender is the party that embeds watermarks into speech to support provenance verification and misuse detection. The defender has full control over the watermarking system and can deploy any watermarking model unknown to the adversary. Following the principle of adaptive threat modeling [27], [28], [29], we further assume that the defender is aware of the adversary's algorithm, including the optimization strategy used to generate δ. The defender leverages this knowledge to develop adaptive countermeasures, such as recovering watermark evidence on suspicious speech samples.

2.3. Limitation of Existing Attacks

Existing watermark evasion attacks can be broadly categorized into three types: noise-based attacks, overwriting attacks, and generative removal attacks. (i) Noise-based attacks [14], [32], [33] perturb the watermarked audio to suppress detection, either through white-box optimization with full knowledge of the embedding and detection pipeline, or through black-box queries to the detector. (ii) Overwriting attack [16] instead embeds surrogate watermarks into the target audio to overwrite the original, relying on either detector feedback or brute-force surrogate ensembles. (iii) Generative removal attacks [15], [17], [30], [34], [35] leverage generative models to reconstruct unwatermarked audio, and split into supervised approaches that require paired clean–watermarked data [15], [30] and unsupervised approaches that rely on pretrained generative priors [17]. Table 1 summarizes the knowledge each attack assumes about the watermarking system.

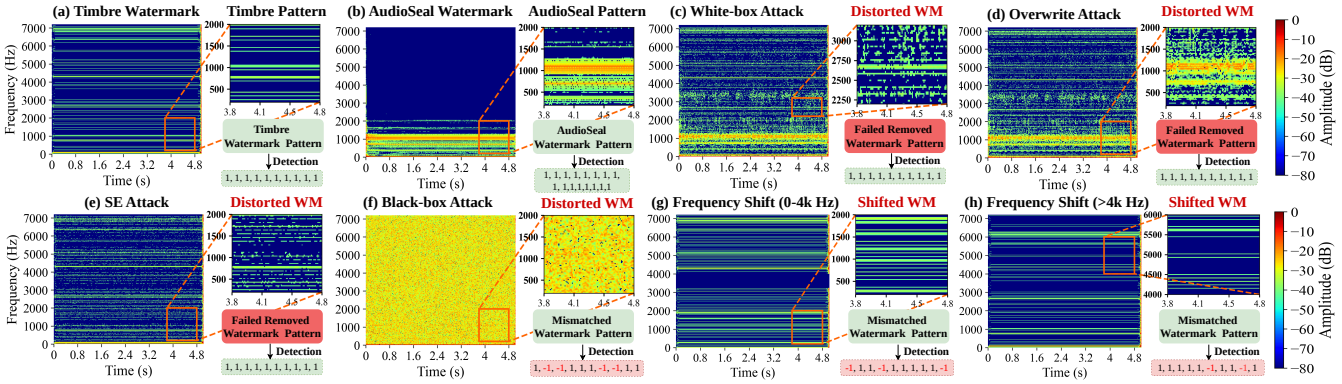


Figure 2: Comparison of watermark patterns under different attack strategies. The Timbre watermark [4] is used as the target, while AudioSeal [5] serves as a surrogate watermarking model to which noise-based [14], overwrite [16], and SE [15] attacks are applied. (a) Timbre watermark and (b) AudioSeal pattern under the same all-ones message; (c)–(f) patterns after existing attacks; (g)–(h) Simple scrambling: (g) 0–4 kHz shifted down by 753 Hz; (h) above 4 kHz shifted down by 753 Hz.

However, existing attacks make assumptions that may not hold under realistic threat models. Noise-based and overwriting attacks typically require white-box access or detector queries, while supervised generative attacks rely on paired clean–watermarked data that is impractical to obtain at scale. Unsupervised generative attacks such as DiffErase [17] avoid these requirements, but their reliance on pretrained generative models (e.g., BigVGAN vocoder [36]) introduces an implicit dependence on the pretraining distribution: On audio from unseen target speakers, the generative prior fails to generalize and produces audible distortions, as we demonstrate in Section 6.3.

Limited Transferability. To understand why existing attacks transfer poorly across watermarking schemes, we compare their perturbation spectra in Fig. 2, using AudioSeal as the surrogate and Timbre as the target. Three observations emerge from the comparison: (i) The white-box attack (Fig. 2(c)) injects localized low-frequency perturbations tailored to AudioSeal's watermark pattern (Fig. 2(b)). Because Timbre exhibits a different watermark structure (Fig. 2(a)), the perturbation only partially suppresses detection; (ii) The overwrite attack (Fig. 2(d)) concentrates its signal below 1200 Hz, leaving most of Timbre's watermark structure unaffected; (iii) The SE attack (Fig. 2(e)) produces sparse, dot-like perturbations that affect only isolated portions of Timbre's stripe pattern (Fig. 2(a)), leaving its overall structure largely intact. Despite their different mechanisms, all three attacks target localized watermark artifacts rather than the broader structure used for verification, thus transfer poorly to unseen watermarking systems.

Perceptible Distortion. A simple way to improve transferability is to enlarge the perturbation budget to further disrupt the watermark pattern. With a high budget, the black-box attack in Fig. 2(f) bypasses detection by flipping 40% of the watermark bits, but at a clear cost: its perturbation energy exceeds the white-box attack by 20–30 dB, producing a dense, noisy spectrogram and rendering the speech unusable. In practice, the resulting audio is no longer usable (e.g., voice cloning) due to severe perceptual degradation.

2.4. Design Intuition

A natural evasion strategy is to locate and remove watermark artifacts, but this is rarely feasible: watermarks are designed to be imperceptible, dispersed across time [5] or confined to narrow frequency bands [4], and masked by the host speech itself. Without clean reference speech, their precise locations cannot be reliably inferred.

The difficulty of locating watermark artifacts raises a fundamental question: *is removal even necessary for evasion?* The answer lies in how watermarks are constructed. Modern watermarking systems [4], [5], [13] encode messages as structured patterns designed to survive common distortions such as compression and noise. To achieve both robustness and capacity, watermark information is typically distributed across the time-frequency representation of the signal rather than embedded at a single location. The form of this distributed watermark pattern commonly differs across methods; e.g., Timbre and AudioSeal produce markedly different spectro-temporal layouts (Fig. 2(a)–(b)), and detection likely depends on matching these watermark-dependent patterns. This observation suggests a different strategy for watermark evasion: rather than removing watermark artifacts, an adversary can perturb the time-frequency structure of speech to disrupt this consistency and cause detection to fail. Since this strategy operates directly on the watermarked signal without requiring access to or modeling of the watermark model, it is particularly suitable for real-world black-box settings where the target system is inaccessible.

Scrambling Observation. To assess whether such direct perturbation is feasible, we probe with a simple and interpretable scrambling operation: band-wise frequency shifting, which warps spectral patterns within fixed frequency bands. We find that frequency shifts corrupt 30% of watermark bits in low-frequency bands and 20% in high-frequency bands (Fig. 2(g)–(h)). Although the watermark artifact is not explicitly erased, detection accuracy drops substantially, establishing structural disruption as a viable attack strategy.

Next, we build on this insight to develop a new framework for structural watermark evasion.

2.5. The ScrambleMark Framework

We propose ScrambleMark, a zero-query, watermark-agnostic evasion attack that defeats detection by disrupting structural consistency. Realizing this idea in a strict black-box setting raises two challenges. *(i) Full-spectrum coverage under perceptual constraints.* A watermark may reside anywhere in the spectrum, as watermark footprints vary substantially across schemes [4], [5]. Without knowing the watermark's location, an effective attack must provide full-spectrum coverage, remaining capable of disrupting watermark structures regardless of where they are embedded. At the same time, indiscriminate perturbations can distort perceptually important speech characteristics such as pitch and timbre. Effective evasion therefore requires achieving two competing objectives: full-spectrum coverage and perceptual quality preservation. *(ii) Recovery resistance.* Deterministic signal-processing transforms such as frequency shifting can effectively disrupt watermark structure, but their fixed transformation rules leave predictable perturbation patterns. An adaptive defender may estimate these patterns and reverse the attack to restore watermark evidence. Robust evasion therefore requires achieving resistance to recovery.

ScrambleMark addresses these challenges through a three-stage pipeline (Fig. 3). (i) The *search* stage addresses the first challenge by systematically exploring the acoustic feature space to identify a scrambling primitive that balances watermark disruption and perceptual quality (Section 3). (ii) The *optimization* stage optimizes a deterministic scrambler against a surrogate watermark to apply this primitive, validating its transferability across unseen schemes and generating the training data for generative scrambling (Section 4). (iii) The *synthesis* stage addresses the second challenge by lifting these deterministic rules into a learned randomized distribution, producing recovery-resistant scrambling while retaining evasion effectiveness (Section 5). We then evaluate ScrambleMark: Section 6 measures evasion effectiveness on commercial platforms and reports human perceptual ratings, and Section 7 tests its resilience to adaptive recovery.

3. Searching for a Feasible Scrambling Space

We first formulate watermark evasion and systematically evaluate perceptual feature perturbations. The resulting analysis identifies harmonic disruption as the key evasion mechanism, which motivates the design of ScrambleMark.

3.1. Problem Formulation

Given a watermarked audio x_{w_t} carrying watermark w_t, the attacker seeks a perturbation δ that evades detection while remaining perceptually close to x_{w_t}:

$$\min_{\delta} \ \mathcal{L}_{\text{perc}}(x_{w_t} + \delta,\, x_{w_t}) \quad \text{s.t.} \quad y_t = 0, \tag{1}$$

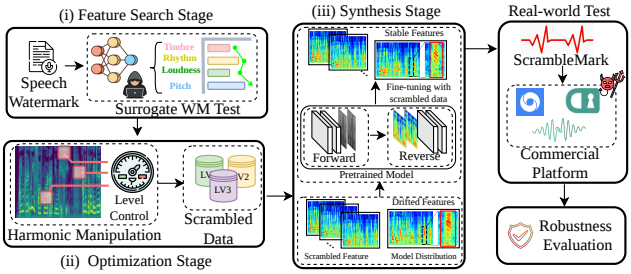


Figure 3: Overview of ScrambleMark.

where δ is the perturbation, $\mathcal{L}_{\text{perc}}$ measures perceptual distance from x_{w_t}, and $y_t = 0$ denotes that the target detector fails to recover the watermark from $x_{w_t} + \delta$. Motivated by the structural nature of modern speech watermarks (Section 2.4), we focus on perturbations that can effectively disrupt the speech structure. However, the space of possible scrambling operations remains vast, and different perturbations may affect watermark detection and perceptual quality in fundamentally different ways. We next seek effective scrambling directions that balance evasion and fidelity.

3.2. Perceptual Features as the Scrambling Space

We explore scrambling directions in a perceptual feature space [37], [38], characterized by *pitch*, *rhythm*, *timbre*, and *loudness*. This choice is motivated by two considerations. (i) The perceptual feature space is compact and interpretable, allowing us to pinpoint the perceptual factors responsible for watermark evasion and to explicitly control the evasion–quality trade-off by tuning feature-level scrambling strength. (ii) Compared to localized noise injection, feature-level modifications naturally tend to propagate across broad time-frequency regions. Such full-spectrum coverage directly addresses the challenge that watermark may reside anywhere in the spectrum, enabling comprehensive disruption of watermark structures while preserving the fidelity.

3.2.1. Feature-level Perturbation Dataset Setup. To systematically probe watermark behavior under feature-level perturbations, we construct an evaluation dataset along two functional axes: (i) perturbation type, covering a set of perceptually meaningful features whose effects jointly span complementary regions of the spectrogram, and (ii) perturbation strength, sampled across a wide range of intensities to characterize the evasion–quality trade-off. The corresponding features are introduced below.

Pitch determines harmonic structure [39], [40]. We consider (i) *Pitch shifting*, which changes pitch by scaling harmonic structure; (ii) *Frequency shifting*, which shifts spectrogram energy along the frequency axis rather than scaling it.

Rhythm governs temporal structure [41], [42]. We consider (i) *Time compression/expansion*, which alters tempo; (ii) *Time cropping*, which removes a signal segment.

Timbre characterizes the spectral envelope of a sound [43]. We consider (i) *Quantization*, which introduces quantization noise; (ii) *Median filtering*, which smooths spectral details.

TABLE 2: Parameters for each perturbation method.

Method	Perturbation Strengths				
	Level 1	Level 2	Level 3	Level 4	Level 5
Median Filtering (samples)	5	15	25	35	45
Quantization (bps)	9	8	7	5	3
Gaussian Noise (dB)	25	20	15	10	-10
Downsampling (HZ)	16000	14000	6000	4000	2500
Background Noise (dB)	25	20	15	10	-10
Amplitude Scaling (%)	60	40	10	5	0.01
Pitch Shift Up (Cents)	15	35	200	300	400
Pitch Shift Down (Cents)	15	35	200	300	400
Frequency Shift (HZ)	10	15	20	25	30
Time Crop(%)	20	40	80	95	99
Time Compression	0.8	0.6	0.4	0.2	0.1
Time Expansion	1.25	1.67	2.5	5	10

Loudness controls overall signal intensity [44], [45]. We consider *amplitude scaling* to vary signal energy.

Baselines. We compare feature-level scrambling against three baselines: (i) additive Gaussian noise; (ii) ambient sound injection [46] (e.g., car); and (iii) speech enhancement [47] and denoising [48], applied as a preprocessing step that aims to suppress watermark-related components.

3.2.2. Experimental Setup. (i) Perturbation Configuration: To cover a broad range of perturbation strengths, we configure each method at five levels (Level 1–5), with parameters summarized in Table 2. Existing speech enhancement and denoising methods are applied as fixed preprocessing steps and therefore have no level settings. (ii) Dataset: We randomly sample 500 speech segments each from Valentini [49] and VoxCeleb1 [50] (1,000 clips total), and apply every method at all five levels to construct the evaluation set.

Target Watermarking Model. For the experiments in this section, we adopt Timbre [4], a robust watermark designed to stay detectable under common audio processing. We use it as a demanding target because it spreads its watermark across nearly the entire spectrum, so evading it requires a perturbation that spans the full spectrum, matching the requirement we identified for effective evasion.

Evaluation Metric. To identify which feature manipulation achieves strong evasion with minimal speech degradation, we define the *Evasion–Quality Trade-off Ratio* (EQTR), which measures the drop in watermark matching score per unit of perceptual quality loss [51], [52]: $\mathrm{EQTR} = \Delta\Theta\,(2-\mathrm{STOI})^{-1}$, where $\Delta\Theta$ is the drop in watermark matching score, computed as the score on the original watermarked speech minus the score on the perturbed speech (both normalized to $[0,1]$). Short-Time Objective Intelligibility (STOI) [53] measures speech intelligibility, with higher values indicating better perceptual quality. The denominator $(2-\mathrm{STOI})$ normalizes the ratio: since practical STOI values typically lie in $[0,1]$, $(2-\mathrm{STOI})$ remains bounded away from zero, and confines EQTR to $[0,1]$. A larger EQTR indicates stronger evasion at lower perceptual cost.

3.2.3. Results Analysis. Fig. 4 reveals a clear advantage of harmonic-axis perturbations. Pitch shifting up & down, and frequency shifting achieve the highest EQTR scores (0.31, 0.28, and 0.25), substantially outperforming Gaussian noise (0.16), rhythm-based perturbations, and denoising methods (both near 0.01). The common property shared by these strongest attacks is harmonic disruption. Pitch shifting rescales every harmonic component by a shared ratio;

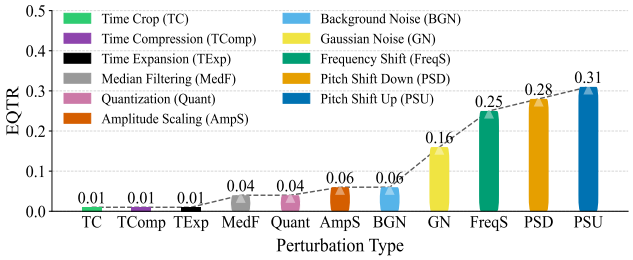


Figure 4: EQTR across feature-level perturbations.

frequency shifting moves them by a shared offset. Both transform the harmonic structure as a coherent whole. The weaker methods touch other axes: timbre-based operations (e.g., quantization) perturb local spectral texture, rhythm-based methods alter temporal structure, and speech enhancement/denoising removes noise-like artifacts, all leaving the harmonic structure intact.

Core Mechanism of Structural Evasion. *Watermark evasion is primarily driven by the targeted disruption of harmonic structures.* This finding motivates a scrambling strategy based on controlled frequency-axis deformation. Instead of injecting generic noise or altering temporal structures, the scrambling mechanism should stretch, compress, or warp harmonic components to break watermark-specific structural consistency while preserving perceptual quality.

3.3. From Harmonic Scrambling to ScrambleMark

While the above strategy is promising, turning it into a practical generative ScrambleMark framework faces an immediate difficulty: training the generator requires a dataset of harmonically scrambled speech, and no public dataset is tailored to this purpose. To close this gap, we need a scalable data generation pipeline that produces scrambled samples achieving both high evasion and good perceptual quality. Accordingly, we structure our approach as follows. *(i) A Deterministic Data Engine.* We first build a deterministic harmonic scrambler that serves as a data engine, supplying the training datasets the generator needs. We study and optimize a deterministic scrambling operation that generalizes across multiple watermarking models. *(ii) Generative ScrambleMark Training.* We then use this optimized scrambler to construct the training dataset for a generative ScrambleMark model. The resulting model preserves the evasion effectiveness, while introducing randomness that improves robustness against adaptive recovery and defenses.

4. Building the Deterministic Data Engine

This section develops a deterministic scrambler for generating ScrambleMark training data. We first construct a band-wise harmonic scrambler, then optimize it against a surrogate watermark detector and evaluate its effectiveness across diverse watermarking models.

Band-wise Harmonic Scrambling Formulation. Since watermark signatures and perceptual sensitivity are distributed

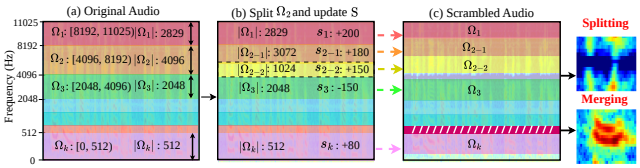


Figure 5: Harmonic partition-shift operation.

unevenly across the frequency spectrum, global transformations (e.g., uniform pitch shifting) inevitably force a suboptimal trade-off between evasion and audio fidelity. To address this, our deterministic scrambler partitions the entire frequency axis into multiple bands, applying independent frequency shifts to each. Specifically, given a watermarked signal x_{w_t}, the scrambler leverages a band-wise signal-processing operator $\mathcal{H}$ to generate the scrambled output:

$$x'_{w_t} = \mathcal{H}(x_{w_t} \mid (\mathbf{\Omega}, \mathbf{S}))\,,$$

where the operator $\mathcal{H}$ is implemented using a modified open-source framework [54] that extends conventional global spectral transformations to flexible band-wise manipulations. Formally, the deterministic scrambler is parameterized by two components: a frequency partition $\mathbf{\Omega}$ and a shift vector $\mathbf{S}$. The partition splits the frequency axis into K contiguous bands $\mathbf{\Omega} = \{\Omega_k\}_{k=1}^{K}$, whose widths $|\Omega_k|$ need not be equal. Each band Ω_k then receives a signed shift $s_k \in \mathbb{R}$ in cents, positive for upward and negative for downward, collected into $\mathbf{S} = \{s_k\}_{k=1}^{K}$. Together, $(\mathbf{\Omega}, \mathbf{S})$ determine how the spectrum is scrambled in frequency-axis.

4.1. Surrogate-Guided Optimization

In practice, the target detector is unknown, so we cannot optimize the scrambler against it directly. We instead optimize against a surrogate: we embed a surrogate watermark w_s into the audio via $x_s = \mathcal{E}(x_{w_t}, w_s)$, then optimize the scrambling parameters to suppress w_s. Because embedded watermarks are imperceptible by design, this surrogate embedding leaves the audio essentially unchanged. We optimize the scrambling parameters $(\mathbf{\Omega}, \mathbf{S})$ by replacing the hard detection constraint in (1) with a soft surrogate penalty:

$$\begin{aligned} \arg\min_{\mathbf{\Omega},\mathbf{S}} \quad & \Theta(\mathcal{D}(x'_s),\, w_s) + \lambda\, \mathcal{L}_{\text{Mel}}(x'_s,\, x_{w_t}) \\ \text{s.t.} \quad & x'_s = \mathcal{H}(x_s \mid (\mathbf{\Omega}, \mathbf{S}))\,, \end{aligned} \tag{2}$$

where the first term, $\Theta(\cdot,\cdot)$, drives the surrogate detector to fail on the scrambled audio, and the second realizes $\mathcal{L}_{\text{perc}}$ as a Mel-spectrogram distance $\mathcal{L}_{\text{Mel}}(\cdot,\cdot)$ that keeps the audio perceptually close to the original [55], [56]. The weight λ balances evasion against fidelity.

Alternating Partition and Shift Optimization. The optimization problem in (2) jointly searches for the frequency partition $\mathbf{\Omega}$ and the corresponding shift vector $\mathbf{S}$. However, this optimization is challenging because $\mathbf{\Omega}$ and $\mathbf{S}$ are coupled: modifying $\mathbf{\Omega}$ changes the number and boundaries of frequency bands, thereby altering the dimensionality of $\mathbf{S}$. Our idea is to break this coupling by alternating (Fig. 5):

Algorithm 1: Partition-Shift Optimization

Input : surrogate audio x_s; initial partition $\mathbf{\Omega}^0$ and shifts $\mathbf{S}^0$; outer iterations T; CMA-ES iterations M; candidate count z
Output: scrambled audio x'_s

1 $\mathbf{\Omega}, \mathbf{S} \leftarrow \mathbf{\Omega}^0, \mathbf{S}^0$; $\quad \mathcal{J}^* \leftarrow \infty$;
2 $\mathcal{J} \leftarrow \Theta(\mathcal{D}(x'_s),\, w_s) + \lambda\, \mathcal{L}_{\text{Mel}}(x'_s,\, x_{w_t})$;
3 **for** $t \leftarrow 1$ **to** T **do**
  // Refine the partition: split a few of the widest bands
4 $\mathcal{K} \leftarrow$ selecting the z widest bands in $\mathbf{\Omega}$;
5 **foreach** $\Omega_k \in \mathcal{K}$ **do**
6 **with probability** $P_k = |\Omega_k| / \max_j |\Omega_j|$:
7 randomly split Ω_k into two sub-bands;
8 each sub-band inherits the shift s_k of Ω_k;
  // Optimize the shifts under the refined partition
9 $\mathbf{S} \leftarrow \texttt{CMA-ES}(\mathbf{\Omega}, \mathbf{S}, M)$; // minimize $\mathcal{J}$, $\mathbf{\Omega}$ fixed
10 $\mathcal{J} \leftarrow \mathcal{J}(\mathbf{\Omega}, \mathbf{S})$
11 **if** $\mathcal{J} < \mathcal{J}^*$ **then** $\mathbf{\Omega}^*, \mathbf{S}^*, \mathcal{J}^* \leftarrow \mathbf{\Omega}, \mathbf{S}, \mathcal{J}$;
12 **return** $x'_s \leftarrow \mathcal{H}(x_s \mid (\mathbf{\Omega}^*, \mathbf{S}^*))$;

we fix the partition $\mathbf{\Omega}$ and optimize the shifts $\mathbf{S}$ under it, then split the wider bands into narrower ones and repeat. Each split gives the next round of shift optimization a finer partition to work on. As the bands grow narrower, each shift acts on a smaller slice of the spectrum, letting the scrambler balance evasion and quality more finely.

Algorithm 1 realizes this idea. We initialize the partition and shifts (Line 1) and the objective function $\mathcal{J}(\cdot,\cdot)$ (Line 2). Each iteration first refines the partition, splitting bands with probability proportional to their width so wider bands divide more frequently (Lines 3–7); each sub-band inherits the shift of the band it split from, keeping the scrambling continuous (Line 8). The shift vector is then optimized under the refined partition using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [57] (Line 9). The resulting configuration is retained only if it reduces the objective (Line 11). The best configuration produces the scrambled audio (Line 12).

4.2. Evasion and Quality Evaluation

We next evaluate whether the optimized scrambler achieves strong cross-model transferability to unseen watermarking models while maintaining perceptual quality.

Datasets. We randomly select 500 test samples from VoxCeleb1 [50] and VOiCES [62], and embed a random watermark message into each sample.

Watermarking methods. We evaluate against three state-of-the-art watermarking methods that span the time and frequency domains: AudioSeal, WavMark, and SilentCipher, each run with its official implementation.

Metrics. We measure evasion effectiveness by the attack success rate (ASR), the fraction of perturbed samples that evade the target detector, and perceptual quality by STOI.

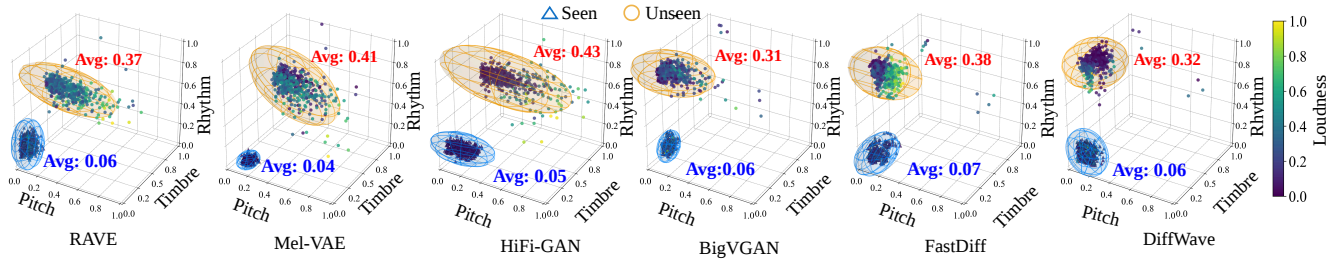


Figure 6: Feature-variation distributions on seen vs. unseen data. Over 500 original clips, we measure variations in loudness (Root Mean Square difference [58]), pitch (F_0 difference [59]), timbre (Mel-Cepstral Distortion [60]), and rhythm (Energy-Envelope Dissimilarity [61]), all normalized to $[0, 1]$ (lower indicates less variation). Details are provided in Appendix A.

Implementation. We use Timbre as the surrogate watermark model for optimization. The partition $\mathbf{\Omega}^0$ is initialized as a single band spanning the full spectrum, with initial shift $\mathbf{S}^0 = 80$ cents and trade-off weight $\lambda = 1$. We run $T = 50$ outer iterations and $M = 50$ CMA-ES iterations per round. For partitioning, at most $z = 3$ bands are probabilistically selected for splitting in each round.

TABLE 3: Evasion Effectiveness and Quality Results.

Metric	AudioSeal		WavMark		SilentCipher		Average
	VoxCeleb1	VOiCES	VoxCeleb1	VOiCES	VoxCeleb1	VOiCES	
ASR	0.98	1.00	1.00	0.99	0.97	0.98	0.99
STOI	0.91	0.92	0.90	0.90	0.91	0.90	0.91

Evaluation Results. Table 3 shows that the surrogate-tuned scrambler transfers effectively across watermarking methods, achieving nearly 99% average ASR while maintaining a STOI of 0.91. These results suggest that the learned scrambling strategy captures weaknesses shared across watermarking schemes rather than overfitting to the surrogate model. The deterministic scrambler therefore serves as an effective data engine, providing the training data needed for generative ScrambleMark, which we develop next.

5. Building Generative ScrambleMark

We now lift the deterministic scrambler into a generative model, ScrambleMark, that introduces the randomness needed to resist recovery. This transition raises two challenges: (i) how to efficiently learn from deterministic scrambling behaviors, and (ii) how to make the generative model preserve perceptual quality on previously unseen audio. This section presents our solution and integrates its components into the full ScrambleMark attack.

5.1. Challenges in Generative Scrambling

We model ScrambleMark as a conditional generator that maps the surrogate-watermarked audio x_s to a scrambled output under a scrambling condition C: $x' = \mathbb{G}_\theta(x_s \mid C)$.

A Strawman Solution. One possibility is to let the deterministic scrambler produce an optimal condition C and train the generator to reproduce it. However, the generator produces randomized outputs, so a condition that is optimal for the deterministic scrambler may no longer be optimal once reproduced by the generator. Moreover, learning a single deterministic scrambling configuration risks exposing a predictable perturbation pattern that a defender could track and reverse. Instead, the generator should learn from multiple valid conditions that achieve similar effectiveness. Realizing such a generator introduces two challenges.

The Conditional-Space Explosion. The space of possible band-wise operations is enormous. Each band carries its own shift, set independently of the others, so the number of full configurations grows exponentially with the number of bands. The solutions in Table 3 already use around 100 bands, each shifted within roughly ± 40 to ± 80 cents. Even with only tens of shift values per band, this gives well over 10^{100} distinct conditions, far beyond what any training set could enumerate. Finer partitions only push the count higher.

Limited Generalization to Unseen Inputs. Our threat model requires the generator to operate on audio it has never seen. This is where unsupervised generative removal attacks tend to break down [17], and ours is no exception: on unseen inputs, the synthesized audio carries audible distortion. To locate its source, we measure how four acoustic features shift from seen to unseen data across generative adversarial networks (GANs), variational autoencoders (VAEs), and diffusion models [63], [64]. The shift is indiscriminate. As Fig. 6 shows, all four features drift on unseen inputs, with average changes rising from below 0.1 to 0.31–0.43, roughly five to eight times higher [65], [66], [67], [68]. For an attack that should touch only the harmonic features, every other deviation, in rhythm or loudness, is wasted perceptual budget. Making the generator change only the target features on unseen audio is the second challenge.

Overview. Fig. 7 summarizes the two-phase ScrambleMark pipeline to solve these challenges. *(i) Scramble-domain dataset generation* tackles the conditional-space explosion by compressing band-wise scrambling operations into a small set of intensity-level conditions. During construction, a recovery-aware objective biases the search toward scrambling configurations that are difficult to invert. *(ii) Generative scrambler training* tackles the generalization challenge by training a diffusion-based generator on the resulting dataset. Pretraining provides general synthesis ability, while dynamic fine-tuning enforces harmonic scrambling and output diversity. Finally, ScrambleMark generates mul-

tiple intensity-level candidates and retains the minimum-loss one. We next present the design of each component.

5.2. Scramble-Domain Dataset Generation

Intensity-Level Condition. To address the conditional-space explosion, instead of mapping each scrambled sample to its own condition, we map a set of samples of similar shift magnitude, differing in their band partitions, to a single intensity-level condition; the collection of these conditions forms the scramble dataset $\mathcal{C}_{\text{scr}}$. This shrinks the conditioning space from per-sample to intensity-level, making training feasible. Because each level pools samples with different band partitions, the generator varies its band structure across outputs rather than reusing a fixed one, making its perturbations harder for a defender to track and reverse. On top of this design, we further harden each sample against recovery and define how each intensity level is built, as detailed below.

5.2.1. Recovery-Aware Construction. To make each sample in $\mathcal{C}_{\text{scr}}$ robust against recovery attempts, we exploit a side effect of the partition–shift operation: shifting adjacent bands often causes their spectral energies to overlap and merge (Fig. 5(c)). Once merged, the original band boundaries are lost, and an inverse shift struggles to separate them cleanly. Merging therefore acts as a natural barrier to recovery, though at the cost of some perceptual quality, since merged harmonic energies cannot be cleanly preserved.

We turn this observation into a recovery penalty during the partition-shift search. A defender does not know the per-band shifts $\mathbf{S}$ and can only attempt recovery with a coarse estimate of the overall shift, so we approximate this by averaging $\mathbf{S}$ into a single scalar $\bar{s} = \frac{1}{K}\sum_{k=1}^{K} s_k$ and applying $-\bar{s}$ uniformly across all bands: $\tilde{x}_{wt} = \mathcal{H}(x'_s \mid (\mathbf{\Omega}, -\bar{s}))$. Detection on $\tilde{x}_{wt}$ then measures how much of the watermark this coarse inversion can recover. We add it as a penalty to the search objective:

$$\mathcal{J} = \Theta(\mathcal{D}(x'_s),\, w_s) + \lambda \mathcal{L}_{\text{Mel}}(x'_s, x_{wt}) + \beta\, \Theta(\mathcal{D}(\tilde{x}_{wt}),\, w_s)\,, \tag{3}$$

where the third term, weighted by β, penalizes configurations whose inversion still recovers the watermark. Configurations with more band merging resist the inversion and incur a smaller penalty, so the search is biased toward them; larger β strengthens this bias at the cost of perceptual quality. We update Line 2 of Algorithm 1 with (3).

5.2.2. Dataset Generation. With the recovery-aware objective in place, we run the search under a range of intensity constraints to build $\mathcal{C}_{\text{scr}}$. We index intensity by the magnitude of the shift vector $\mathbf{S} = \{s_k\}_{k=1}^{K}$: for the i-th level, we constrain each shift to $|s_k| \in [\gamma_i, \gamma_i + \alpha)$, where γ_i is the level's starting magnitude and α a fixed interval width. We run Algorithm 1 under each constraint, repeating with different random initializations to diversify the band partitions within a level. As illustrated in Fig. 7, each speech has a subset $\mathcal{C}_i$ of N_s scrambled samples at each intensity level,

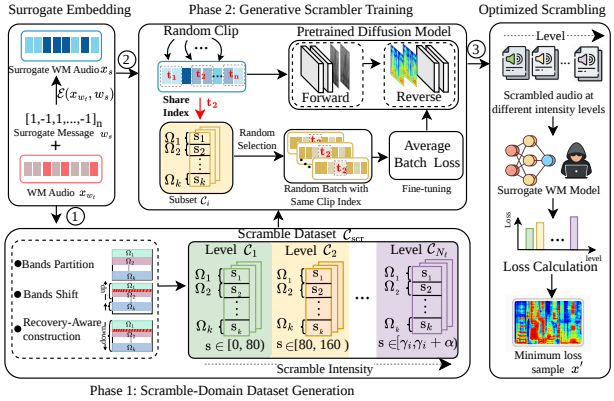


Figure 7: ScrambleMark: dataset generation and training.

generated under the same shift range. With N_ℓ levels, the subsets $\{\mathcal{C}_i\}_{i=1}^{N_\ell}$ form the scramble dataset $\mathcal{C}_{\text{scr}}$ with $N_\ell \times N_s$ samples per speech.

5.3. Generative ScrambleMark Training

We next address the challenge of generalizing harmonic scrambling to unseen audio without introducing unnecessary distortions. Our solution is a pretraining then fine-tuning strategy: pretraining provides general speech synthesis ability, while fine-tuning on $\mathcal{C}_{\text{scr}}$ aligns each scrambling condition with harmonic modifications. Thus, perturbations remain concentrated on harmonic structure for unseen inputs.

5.3.1. Model and Pretraining. We seek a generator with non-deterministic outputs, both for perceptual diversity and for the randomized generation that resists recovery. Diffusion models fit on two counts. First, different noise initializations give diverse outputs for the same input, so the scrambling cannot be pinned to a fixed pattern and reversed. Second, the step-by-step denoising operates on fine-grained spectral detail and trains stably on limited data, reconstructing audio at high fidelity. We adopt DiffWave [69], a diffusion vocoder that generates a waveform from a conditioning spectrogram and requires only six inference steps under fast sampling while maintaining high-fidelity speech quality; we drive it with the spectrogram of the input audio so that ScrambleMark maps the surrogate-watermarked audio to its scrambled waveform. To support different intensity levels, a learnable linear layer encodes the input condition, and the resulting embedding is added to the DiffWave spectrogram features as the conditioning input for fine-tuning.

5.3.2. Dynamic Fine-Tuning. We now fine-tune the model on $\mathcal{C}_{\text{scr}}$. The generator should avoid converging to a fixed scrambling pattern that can be reversed. To this end, we introduce two complementary sources of randomness, shown in Fig. 7: randomized supervision across scrambling targets and dynamic temporal scrambling across audio segments.

Randomized supervision adds randomness at the sample level. For any input x_s, the subset $\mathcal{C}_i$ holds many scrambled

samples at the same-level intensity, each one a valid scrambled output the generator may learn to produce. Rather than fix the generator to all of them, each mini-batch randomly draws a small set of 16 and uses them as the generation targets for this update, resampling at the next. The generator learns to produce these samples in the standard diffusion manner: for each one, we add Gaussian noise at a randomly drawn step r of a 50-step schedule and train the model to predict that noise from the noisy sample $x'_{s,r}$, the step r, and the input x_s, so that it learns to turn x_s into a scrambled output. Averaging the loss over the 16 samples and resampling them each update helps the generator avoid locking onto a fixed scramble pattern.

Dynamic temporal scrambling applies this random selection within time. The randomized supervision mechanism draws random targets for the input as a whole; here we instead crop a random temporal segment of the input and draw its targets from the same time range of the scrambled samples from $\mathcal{C}_i$. Because each segment is paired with its own random selection, the scrambling varies across the audio, making it harder to track or reverse, and the generator learns segment-level scrambling rather than one static pattern over the full duration.

Fine-tuning Objective. Given source speech x_s and its scrambled counterpart $x'_s \sim \mathcal{C}_{\text{scr}}$, where $x'_{s,r}$ denotes the noisy version of x'_s at diffusion step r, we fine-tune the generator to predict the injected diffusion noise:

$$\theta^\star = \arg\min_\theta \mathbb{E}_{x_s,\, x'_s \sim \mathcal{C}_{\text{scr}},\, \epsilon,\, r} \left[\left\| \epsilon - \mathbb{G}_\theta(x'_{s,r},\, r \mid x_s) \right\|_1 \right]. \tag{4}$$

Here, each x'_s is a scrambled sample generated by the deterministic scrambler and stored in $\mathcal{C}_{\text{scr}}$, serving as the supervision target during training. The resulting generator $\mathbb{G}_{\theta^\star}$ is then used to perform the attack.

5.4. Attack Formulation

We now assemble the components built in this section into the final attack for (1). Since the target detector is inaccessible, we adopt the surrogate-guided formulation (2). The attack is formulated as finding the optimal scrambling condition $\mathcal{C}_i$ from the candidate set $\mathcal{C}_{\text{scr}}$:

$$\begin{aligned} \mathcal{C}_i^\star = \arg\min_{\mathcal{C}_i \in \mathcal{C}_{\text{scr}}} \Theta(\mathcal{D}(x'),\, w_s) + \lambda\, \mathcal{L}_{\text{Mel}}(x',\, x_{wt}) \\ \text{s.t. } x' = \mathbb{G}_{\theta^\star}(x_s \mid \mathcal{C}_i), \end{aligned} \tag{5}$$

where $\mathcal{C}_i$ is a scrambling condition drawn from the candidate set $\mathcal{C}_{\text{scr}}$, Θ penalizes detection of the surrogate watermark w_s from x', $\mathcal{L}_{\text{Mel}}$ quantifies perceptual degradation, and λ balances evasion and quality. Since $\mathcal{C}_{\text{scr}}$ is a finite set, we solve (5) via exhaustive search over $\mathcal{C}$, generating x' under each candidate $\mathcal{C}_i$ and selecting the minimizer $\mathcal{C}_i^\star$.

6. Evaluation

Our evaluation proceeds in two parts. (i) Academic schemes: we measure the evasion success and audio quality of ScrambleMark across various academic schemes. (ii) Real-world impact: we attack three commercial platforms and conduct a human study on perceptual quality.

6.1. Experimental Setup

Target Watermarks. We evaluate ScrambleMark against seven academic watermarking schemes spanning the time and frequency domains: Timbre [4], AudioSeal [5], WavMark [3], Collaborative [70], AudioMarkNet [13], RobustDNN [25], and SilentCipher [71], all run with their official checkpoints. Timbre serves as our surrogate by default; we also attack it as a target (marked *) using AudioSeal as the surrogate. We keep each scheme's default detection threshold κ; Table 8 in Appendix B reports each scheme's detection performance under this threshold.

6.1.1. Comparison Attacks. The baselines in Table 4 include (i) noise-based attacks [14], [32], [33], [72], [73], using the default 20-dB SNR bound for white-box attacks and a 10,000-query budget for black-box attacks; (ii) overwriting attacks [16], using 3 embeddings from the same surrogate with different messages; (iii) supervised generative removal [15], [30], [74]; and (iv) unsupervised generative removal [17], [31], both using official default settings. To match our threat model, methods requiring watermark knowledge use Timbre as the surrogate watermarking model, switching to AudioSeal when Timbre is the target.

Dataset. To eliminate training-data ambiguity mentioned in Table 1 and enable a fair comparison of feature variation on unseen inputs, all generative models, including ScrambleMark, are trained on LJ Speech [75]. We evaluate them on a disjoint multilingual benchmark spanning English [50], Chinese [76], Korean [77], French [78], and German [78], without speaker overlap. For each language, we randomly sample 500 speech clips.

6.1.2. ScrambleMark Configuration. *(i) Scramble-domain dataset generation.* We construct the scramble dataset using the deterministic scrambler. The evasion–quality weight is set to $\lambda = 1$ and the recovery weight to $\beta = 1$. Alternating Partition–Shift optimization runs for $T = 100$ outer iterations with $M = 50$ CMA-ES iterations per round (out-of-range values are clipped to the corresponding bounds during CMA-ES optimization). We initialize $\mathbf{\Omega}^0$ as a single band spanning the full spectrum, set the initial shift to $\mathbf{S}^0 = 80$ cents, and allow at most $z = 3$ bands to be selected for splitting in each round. The resulting dataset contains $N_\ell = 10$ intensity levels with $N_s = 100$ samples per level and a shift-range width of $\alpha = 60$ cents.
(ii) Generative training. We use DiffWave [69] with 30 residual layers and 64 channels. It is pretrained on LJ Speech [75] for 10^5 steps and fine-tuned on the scramble dataset for 5×10^4 steps using Adam (lr $= 2 \times 10^{-4}$). Fine-tuning uses batch size 16 and a 50-step diffusion schedule.
(iii) Attack generation. At inference time, we enumerate all $N_\ell = 10$ intensity levels and generate one scrambled candidate per level. The candidate minimizing the attack

TABLE 4: ScrambleMark compared with existing attacks on academic watermarking schemes.

Method	AudioSeal		WavMark		AudioMarkNet		SilentCipher		Collaborative		RobustDNN		Timbre*		**Average**	
	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)
White-Box-PGD [14]	0.00	1.00	0.00	1.00	0.15	1.00	0.00	1.00	0.02	1.00	0.00	1.00	0.00	1.00	0.02	1.00
White-Box-I-FGSM [14]	0.00	1.00	0.00	1.00	0.13	1.00	0.00	1.00	0.01	1.00	0.00	1.00	0.00	1.00	0.02	1.00
Black-Box-HSJA [14]	0.55	0.85	1.00	0.83	0.46	0.83	1.00	0.86	1.00	0.84	0.03	0.87	0.56	0.81	0.66	0.84
Black-Box-S-HSJA [14]	0.82	0.84	1.00	0.85	0.58	0.81	1.00	0.84	1.00	0.84	0.03	0.86	0.20	0.83	0.66	0.84
Overwriting Attack [16]	0.00	0.98	0.03	0.97	0.14	0.99	0.27	0.98	0.37	0.98	0.02	0.97	0.00	0.98	0.12	0.98
SE Attack [15]	0.63	0.98	0.18	0.99	0.05	0.99	0.26	0.99	0.21	0.99	1.00	0.98	0.00	0.98	0.33	0.99
Harmonic Attack [30]	0.57	0.96	0.10	0.99	0.00	0.97	0.17	0.96	0.13	0.98	0.91	0.98	0.00	0.97	0.27	0.97
DiffErase-L [17]	1.00	0.70	1.00	0.72	0.93	0.70	1.00	0.70	1.00	0.67	0.93	0.68	1.00	0.73	0.98	0.70
DiffErase-M [17]	1.00	0.72	1.00	0.71	0.95	0.71	1.00	0.73	1.00	0.69	0.95	0.70	1.00	0.75	0.99	0.72
Encodec [31]	0.96	0.75	1.00	0.73	0.81	0.72	1.00	0.71	1.00	0.70	0.93	0.69	0.94	0.72	0.95	0.72
ScrambleMark	1.00	0.91	1.00	0.90	0.98	0.88	1.00	0.92	0.96	0.90	0.86	0.93	1.00	0.91	**0.97**	**0.91**

objective is selected as the final output. Detailed inference-time computational is reported in Appendix B.

(iv) Surrogate watermarking model. Same as the comparison attacks, we use Timbre as the surrogate watermarking model and embed a random 10-bit message to produce x_s. When Timbre is the target, AudioSeal serves as the surrogate.

6.1.3. Evaluation Metrics. We measure attack effectiveness using ASR, defined as the fraction of perturbed watermarked speeches that fall below the target model's detection threshold, and assess speech quality using STOI.

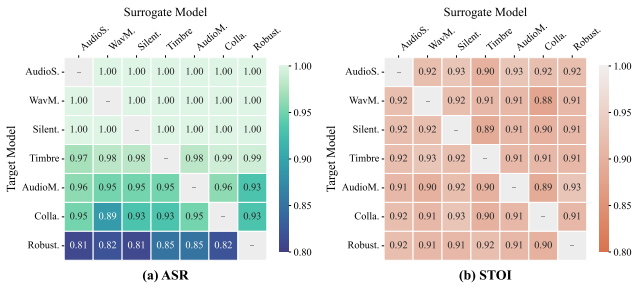


Figure 8: Transferability and fidelity across surrogates.

6.2. Effectiveness on Academic Schemes

We first evaluate whether ScrambleMark transfers across seven academic watermarking models and at what cost to audio quality. Results can be found in Table 4.

Observation 1: ScrambleMark achieves high evasion effectiveness while preserving speech quality. Averaged across all seven watermarking models, ScrambleMark attains an ASR of 0.97 and a STOI of 0.91. Its performance is also consistent across targets, maintaining ASR above 0.86 and STOI above 0.88 on every watermarking model. In contrast, existing attacks achieve at most one objective. White-box attacks, Overwriting, SE, and HarmonicAttack preserve audio quality but attain average ASRs of at most 0.33. Conversely, attacks that evade watermark detection often do so at the expense of perceptual quality. Generative removal methods such as Encodec and DiffErase achieve high ASR but reduce STOI to 0.70–0.75. Together, these results show that ScrambleMark is the only method that simultaneously achieves strong transferability and high perceptual quality.

Detector Threshold Sensitivity. We further examine ScrambleMark under relaxed detector thresholds with higher FPRs. As shown in Table 11, ScrambleMark maintains high ASR across increasingly relaxed thresholds. Detailed setup and results are provided in Appendix D.2.

6.2.1. Impact of Surrogate Choice. We evaluate whether ScrambleMark remains effective under different surrogate watermarking models. To this end, we rotate the surrogate across all academic watermarking models considered in our evaluation. We use the same multilingual benchmark as before, randomly sampling 500 speech clips per language.

Observation 2: ScrambleMark transfers consistently regardless of the surrogate choice. Fig. 8 shows that for most target watermarking models, ASR remains between 0.93 and 1.00 regardless of which surrogate is used, while STOI stays tightly concentrated around 0.90–0.93. This indicates that the attack does not rely on a particular surrogate watermarking model. Surrogate robustness nevertheless has a mild effect. RobustDNN remains the most difficult target across surrogate choices, likely due to its robustness-oriented training strategy. Weaker surrogates such as WavMark [3] also transfer slightly worse to some targets (e.g., 0.89 ASR on Collaborative and 0.82 ASR on RobustDNN), suggesting a greater tendency to capture model-specific artifacts. Overall, these results suggest that ScrambleMark targets shared harmonic structures rather than detector-specific weaknesses.

6.2.2. Impact of Fine-tuning. We examine the impact of fine-tuning. Fig. 10 shows that fine-tuning progressively reduces feature drift in unseen speech, while improving STOI. Details are provided in Appendix B.

6.3. Evaluations on Commercial Platforms

We evaluate ScrambleMark against three commercial systems. We measure evasion effectiveness and assess audio quality using (i) STOI and (ii) human-subject study.

6.3.1. Commercial Watermarks Setup. Commercial speech watermarking comes in two forms, and each dictates how we obtain its watermarked test data.

(i) Watermarking-as-a-service (WaaS) lets a user submit their own audio to a service that embeds an imperceptible watermark for ownership protection and leak tracing; Resemble AI [20] and TrustedAudio [18] offer this model, we build their test sets from English utterances in Section 6.1: for Resemble AI we use a balanced 3 female / 3 male split and clips of 3 to 12 s (42 utterances total if matched to the

TABLE 5: Commercial platforms: ASR & objective quality

Method	SynthID		Resemble AI		TrustedAudio		**Average**	
	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)
White-Box-PGD [14]	0.00	1.00	0.00	1.00	0.15	1.00	0.05	1.00
White-Box-I-FGSM [14]	0.00	1.00	0.00	1.00	0.13	1.00	0.04	1.00
Black-Box-HSJA [14]	0.79	0.87	1.00	0.83	1.00	0.77	0.93	0.82
Black-Box-S-HSJA [14]	0.38	0.83	1.00	0.85	1.00	0.79	0.79	0.82
SE Attack [15]	0.00	1.00	0.00	0.99	0.00	0.99	0.00	0.99
Overwriting Attack [16]	0.00	0.98	0.00	0.98	0.00	0.99	0.00	0.98
HarmonicAttack [30]	0.00	0.95	0.00	0.98	0.00	0.97	0.00	0.97
DiffErase-L [17]	0.90	0.75	1.00	0.73	1.00	0.71	0.97	0.73
DiffErase-M [17]	0.93	0.79	1.00	0.72	1.00	0.72	0.98	0.74
Encodec [31]	0.90	0.78	0.98	0.73	1.00	0.69	0.96	0.73
ScrambleMark	0.93	0.92	1.00	0.91	1.00	0.90	**0.98**	**0.91**

SynthID setup), and for TrustedAudio we use longer clips of around 20 s to suit its tracing use case (42 utterances total).

(ii) Generation-native watermarking embeds the mark into the waveform as a text-to-speech (TTS) model generates speech, certifying content as machine-generated rather than protecting user audio; Google SynthID [19] is the representative example. Since its watermark exists only in platform-generated speech, we synthesize the test set with Google's text-to-speech. We use six speakers (3 female and 3 male) and have each read distinct transcripts drawn from LibriSpeech [79], producing 42 watermarked utterances of 3 to 12 seconds, with no transcript reused across speakers.

6.3.2. Objective Speech Quality Evaluation. We first evaluate adversarial audio quality using STOI, enabling large-scale assessment across all attack methods.

Setup. For WaaS watermarking systems, we adopt the attack configuration described in Section 6.1. For generation-native watermarking systems, we expand the initial configuration $\mathbf{\Omega}^0$ to comprise 200 randomly initialized sub-bands. This warm-start strategy introduces greater diversity into the initial perturbation pattern and enables more effective spectrogram-level modifications. We further evaluate the impact of initial band through a sensitivity analysis in Table 12.

Observation 3: ScrambleMark remains effective against deployed commercial watermarking systems. As shown in Table 5, ScrambleMark achieves an average ASR of 0.98 and a STOI of 0.91 across all three platforms. In comparison, the strongest baselines, DiffErase-M and Encodec, achieve similar ASRs (0.98 and 0.96) but substantially lower STOI scores (0.74 and 0.73). As a result, ScrambleMark improves speech intelligibility by approximately 23% without sacrificing evasion effectiveness. This trend is consistent across all platforms, where ScrambleMark maintains ASRs between 0.93 and 1.00 while preserving high audio quality.

6.3.3. Cross-Surrogate Transferability. We further evaluate whether transferability to commercial platforms depends on the surrogate by testing each academic model.

TABLE 6: Cross-Surrogate Transferability and Quality.

Surrogate Model	SynthID		Resemble AI		TrustedAudio		**Average**	
	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)	ASR(↑)	STOI(↑)
AudioSeal	0.86	0.92	0.98	0.90	1.00	0.90	0.95	0.91
WavMark	0.83	0.92	0.93	0.91	1.00	0.91	0.92	0.91
AudioMarkNet	0.86	0.89	0.93	0.90	1.00	0.90	0.93	0.90
SilentCipher	0.83	0.92	0.90	0.89	1.00	0.89	0.91	0.90
Collaborative	0.86	0.90	0.98	0.91	1.00	0.90	0.95	0.90
RobustDNN	0.90	0.90	1.00	0.91	1.00	0.91	0.97	0.91
Timbre	0.93	0.92	1.00	0.91	1.00	0.91	0.98	0.91

TABLE 7: Commercial platforms: ASR & human ratings.

Method	SynthID		Resemble AI		TrustedAudio		Average	
	ASR↑	MOS↑	ASR↑	MOS↑	ASR↑	MOS↑	ASR↑	MOS↑
Black-Box-HSJA [14]	29/30	1.16	30/30	1.08	30/30	1.14	98.9%	1.13
DiffErase-L [17]	25/30	1.52	30/30	1.35	30/30	2.29	94.4%	1.72
DiffErase-M [17]	28/30	1.27	30/30	1.32	30/30	2.08	97.8%	1.56
ScrambleMark (Ours)	**27/30**	**3.95**	**30/30**	**4.36**	**30/30**	**4.20**	**96.7%**	**4.17**

Results. Table 6 confirms transferability to commercial platforms. Across surrogates, average ASR is 0.91–0.98 with STOI of 0.90–0.91, reaching 1.00 on TrustedAudio, showing that the transferability extends to commercial platforms.

6.3.4. Robustness and Sensitivity. We evaluate ScrambleMark's statistical robustness, parameter sensitivity, robustness to multiple watermarks, and impact on unwatermarked audio. Tables 10, 12, 13, and 15 show stable performance across independent runs, characterize parameter trade-offs, demonstrate effective multi-watermark evasion, and confirm preserved unwatermarked-audio utility, respectively. Results are provided in Appendices D.1, D.3, and D.4.

6.3.5. Human-Subject Evaluation. We conduct a human-subject study to assess the perceptual quality of perturbed speech. To limit participant burden, we consider only attacks that successfully evade commercial systems: DiffErase-L, DiffErase-M, and Black-Box-HSJA. We exclude White-box, SE, Overwriting, and HarmonicAttack due to their low ASR.

Dataset. To cover diverse speech content, we construct a new set comprising 30 samples per platform (5 different utterances from the same 6 speakers set used in Section 6.3.1) to increase utterance diversity and verify ASR stability on new utterances from the same speakers. We evaluate our attack and 3 baselines. Each method generates one adversarial version of every source sample, resulting in 360 adversarial clips across the 3 platforms.

Human Evaluation Setup. We recruited 24 normal-hearing university students (12 male, 12 female; aged 20–30) to evaluate the perceived quality of these clips. The study protocol was approved by our Institutional Review Board (IRB). The experiment was conducted in a quiet environment using headphones. Following the standard Mean Opinion Score (MOS) protocol, participants first listened to the original reference audio, followed by the corresponding perturbed speech. They rated the perceptual quality on a continuous 0–5 scale: 0–1 (Bad), 1–2 (Poor), 2–3 (Fair), 3–4 (Good), and 4–5 (Excellent). To prevent cognitive overload, participants were divided into three groups of eight, with each group evaluating 120 clips from a single commercial platform.

Observation 4: ScrambleMark achieves substantially higher human ratings than existing attacks. Table 7 reports the MOS of each attack method. ScrambleMark consistently achieves the highest perceptual quality across all platforms, with an average MOS of 4.17 while maintaining successful watermark evasion. In contrast, the strongest baseline, DiffErase-L, achieves an average MOS of only 1.72 despite attaining a comparable attack success rate. Overall, ScrambleMark improves perceptual quality by 2.4 times over DiffErase-L, indicating that effective watermark removal

need not come at the expense of audio quality. Appendix D.5 confirms that ScrambleMark achieves significantly higher MOS ratings than all baselines across three platforms.

7. Robustness to Adaptive Defenses

7.1. Adaptive Recovery Design

We consider a defender that suspects a speech has been scrambled and tries to recover it. To grant the defender a strong possible footing, we assume it holds the original watermarked audio x_{w_t} as a reference. The defender applies the same band-wise operator $\mathcal{H}$ used by the adversary, which performs a per-band frequency shift $\mathbf{S}$ over a band partition $\mathbf{\Omega}$, to map the scrambled speech x'_s to a recovered audio $x_{rec} = \mathcal{H}(x'_s \mid (\mathbf{\Omega}, \mathbf{S}))$. The defender optimizes $\mathbf{S}$ so that x_{rec} approaches the reference x_{w_t} under two objectives.
(i) Perceptual recovery drives x_{rec} back toward x_{w_t} in perceptual quality, under a Mel loss:

$$\min_{\mathbf{S}} \ \mathcal{L}_{\text{Mel}}(x_{rec}, x_{w_t}). \tag{6}$$

(ii) Bit recovery drives the bits extracted from x_{rec} back toward those of x_{w_t}, maximizing the matching score Θ:

$$\max_{\mathbf{S}} \ \Theta(\mathcal{D}(x_{rec}), \mathcal{D}(x_{w_t})). \tag{7}$$

Recovery Procedure. We assume an adaptive defender that knows the attack algorithm and attempts to recover the per-band shifts applied by the attacker. The spectrogram is partitioned into 500 equal-width frequency bands, and the shift vector $\mathbf{S}$ is optimized via greedy search. For each band, candidate shifts are evaluated at 15-cent intervals, and the best shift is retained. The procedure is repeated for five passes over all bands with early stopping upon convergence.

7.2. Robustness Evaluation

Setup. To assess recovery robustness, we compare ScrambleMark with the deterministic scrambler in two settings. *(i) Academic schemes*: we select 350 successful ScrambleMark samples per scheme from Section 6.1.1, totaling 2,100 samples across six schemes. *(ii) Commercial platforms*: we randomly select 5 successful samples per speaker from the six speakers in Sections 6.3.2 and 6.3.5, resulting in 90 samples across three platforms. Deterministic samples are generated using Algorithm 1 under the configuration of Section 4.2 and retained only if they evade the corresponding detector, matching ScrambleMark in sample count.

Since commercial platforms expose only binary detection outcomes and no matching scores, we evaluate perceptual recovery only. We further exclude RobustDNN from bit recovery because optimizing for bit recovery inflates its false positive rate (FPR), causing speech carrying non-target messages to be identified as the target watermark. Detailed FPR analysis for each model can be found in Appendix C.
Metric. We evaluate *pre-recovery* and *post-recovery ASR*, corresponding to attack success before and after recovery,

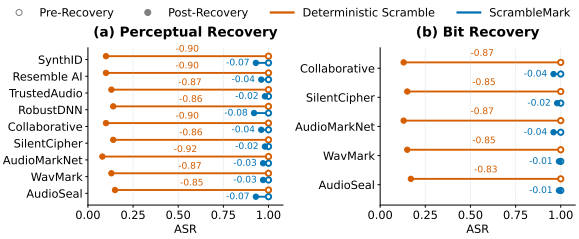


Figure 9: Attack robustness under recovery.

respectively. The drop between the two reflects the attack's robustness to recovery: smaller drops indicate that more attack effectiveness is retained after recovery.

Observation 5: ScrambleMark remains robust against adaptive recovery. Fig. 9 reports ASR before and after perceptual and bit recovery. On academic watermarking schemes, recovery reduces the deterministic scrambler's ASR by 0.88 on average, whereas ScrambleMark incurs only a 0.04 drop. On commercial platforms, the corresponding drops are 0.85 and 0.02. In both settings, recovery has only a marginal impact on ScrambleMark's effectiveness. This robustness stems from its learned scrambling distribution, which does not expose a deterministic recovery path.

Our adaptive recovery focuses on reversing the known band-wise shift operator. A stronger defender may train a detector-aware recovery model and search over intensity levels using detector feedback. While feasible, such recovery remains non-trivial because band shifting may merge or split features, causing information loss and ambiguous inversion. Reconstructing missing features may further remove watermark evidence or introduce watermark-like artifacts. These stronger recovery strategies are worth further study.
Ethical Considerations: Vulnerability disclosure and human-subject considerations are discussed in Appendix E.

8. Related Work

Image Watermarking Evasion Attacks. Watermark evasion has been studied more extensively for images, spanning white-box, black-box, and no-box settings. These attacks share a common recipe: they locate the watermark in a fixed representation, such as the image spectrum [80], the bit distribution [81], or a VAE latent space [82], [83], and then perturb that representation to remove the watermark until detection fails. Our ScrambleMark uses harmonic scrambling to disrupt watermark integrity without localization.
Adversarial Speech Attacks. Existing speech attacks add perturbations that target speech recognition [84], [85], [86], [87], [88], [89] or speaker recognition [27], [28], [52], [90], [91]. Because their perturbation only needs to flip the model, it stays small and localized. Full-spectrum watermarks require degrading full-spectrum perturbations, whereas ScrambleMark preserves naturalness by scrambling harmonics.
Generative Models for Speech. Generative speech models such as voice cloning and text-to-speech [92], [93], [94] are built to synthesize a target speaker's timbre, reproducing

who is speaking. None of them is designed to scramble the harmonic structure of audio, which is what watermark evasion requires. Repurposing these models for evasion is also not straightforward: trained to reconstruct or imitate seen voices, they distort audio drawn from unseen distributions [17], the same failure we observe for unsupervised removal attacks. ScrambleMark adapts a generative model to perform harmonic scrambling and, through its pretraining and fine-tuning design, holds this behavior on unseen inputs.

9. Conclusion

We introduce *structural evasion*, a new paradigm showing that speech watermarks can be evaded without being removed. Building on this insight, we propose ScrambleMark, a black-box harmonic-scrambling method that achieves high evasion across ten watermark models while preserving speech quality and resisting adaptive recovery.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation (NSF) under Grants CNS-2451168, CCF-2551963, DGE-2623286, and CCF-2551964.

References

[1] Z. Su, G. Zhang, F. Yue, L. Chang, J. Jiang, and X. Yao, "Snr-constrained heuristics for optimizing the scaling parameter of robust audio watermarking," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2631–2644, 2018.

[2] J. Zhao, T. Zong, Y. Xiang, L. Gao, G. Hua, K. Sood, and Y. Zhang, "Ssvs-ssvd based desynchronization attacks resilient watermarking method for stereo signals," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 448–461, 2022.

[3] G. Chen, Y. Wu, S. Liu, T. Liu, X. Du, and F. Wei, "Wavmark: Watermarking for audio generation," *arXiv*, 2023.

[4] C. Liu, J. Zhang, T. Zhang, X. Yang, W. Zhang, and N. Yu, "Detecting voice cloning attacks via timbre watermarking," in *NDSS*, 2024.

[5] R. San Roman, P. Fernandez, H. Elsahar, A. Défossez, T. Furon, and T. Tran, "Proactive detection of voice cloning with localized watermarking," in *ICML*. PMLR, 2024, pp. 43 180–43 196.

[6] M. S. Uddin, M. Hasan, T. Shimamura *et al.*, "Audio watermarking: A comprehensive review." *IJACSA*, vol. 15, no. 5, 2024.

[7] P. Kumsawat, K. Attakitmongcol, and A. Srikaew, "Digital audio watermarking for copyright protection based on multiwavelet transform," in *CCIS*. Springer, 2008, pp. 207–217.

[8] G. Hua, J. Huang, Y. Q. Shi, J. Goh, and V. L. Thing, "Twenty years of digital audio watermarking—a comprehensive review," *Signal processing*, vol. 128, pp. 222–242, 2016.

[9] CNN, "Arup's $25 million deepfake scam," 2024, https://www.cnn.com/2024/05/16/tech/arup-deepfake-scam-loss-hong-kong-intl-hnk.

[10] FBI, "Senior U.S. officials impersonated in malicious messaging campaign," 2025, https://www.fbi.gov/investigate/cyber/alerts/2025/senior-us-officials-impersonated-in-malicious-messaging-campaign.

[11] FCC, "FCC proposes $6 million fine for illegal robocalls that used President Biden deepfake generative AI voice," 2024, https://docs.fcc.gov/public/attachments/DOC-402762A1.pdf.

[12] Y. Li, W. Liu, and D. Lin, "TriniMark: A robust generative speech watermarking method for trinity-level attribution," *arXiv*, 2025.

[13] W. Zong, Y.-W. Chow, W. Susilo, J. Baek, and S. Camtepe, "{AudioMarkNet}: Audio watermarking for deepfake speech detection," in *USENIX Security 25*, 2025, pp. 4663–4682.

[14] H. Liu, M. Guo, Z. Jiang, L. Wang, and N. Z. Gong, "Audiomarkbench: Benchmarking robustness of audio watermarking," *NeurIPS*, vol. 37, pp. 52 241–52 265, 2024.

[15] A. López-López, E. Rosello, and A. M. Gomez, "Speech watermarking removal by dnn-based speech enhancement attacks," in *Proc. IberSPEECH 2024*, 2024, pp. 16–20.

[16] L. Yao, C. Huang, S. Wang, J. Xue, H. Guo, J. Liu, P. Lin, T. Ohtsuki, and M. Pan, "Yours or mine? overwriting attacks against neural audio watermarking," in *AAAI*, vol. 40, no. 33, 2026, pp. 27 756–27 763.

[17] L. Yao, X. Zhong, C. Huang, X. Zhao, H. Guo, A. Li, J. Liu, T. Ohtsuki, and M. Pan, "Audio Pirates: Black-box Audio Watermark Removal via Diffusion Priors," *arXiv:2605.30614*, 2026.

[18] TrustedAudio, "TrustedAudio: Secure, Watermarked Audio Delivery Platform," [Online]. Available: https://www.trustedaudio.com/about, accessed: Jun. 11, 2026.

[19] Google DeepMind, "SynthID: A Tool to Watermark and Identify Content Generated Through AI," [Online]. Available: https://deepmind.google/models/synthid/, accessed: Jun. 11, 2026.

[20] Resemble AI, "Resemble Watermarker," [Online]. Available: https://www.resemble.ai/products/watermarker, accessed: May 29, 2026.

[21] V. M. Manikandan and V. Masilamani, "Histogram shifting-based blind watermarking scheme for copyright protection in 5G," *Comput. Electr. Eng.*, vol. 72, pp. 614–630, 2018.

[22] A. A. Hosny, W. A. Murtada, and M. I. Youssef, "Improving LSB audio steganography using simulated annealing for satellite telemetry," in *Proc. ICENCO*, 2018, pp. 11–16.

[23] C. I. Podilchuk and E. J. Delp, "Digital watermarking: Algorithms and applications," *IEEE Signal Process. Mag.*, pp. 33–46, 2001.

[24] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for images, audio and video," in *Proc. IEEE ICIP*, vol. 3, 1996, pp. 243–246.

[25] K. Pavlović, S. Kovačević, I. Djurović, and A. Wojciechowski, "Robust speech watermarking by a jointly trained embedder and detector using a dnn," *Digital Signal Processing*, 2022.

[26] N. Cvejic and T. Seppänen, "Spread spectrum audio watermarking using frequency hopping and attack characterization," *Signal processing*, vol. 84, no. 1, pp. 207–213, 2004.

[27] Z. Yu, Y. Chang, N. Zhang, and C. Xiao, "SMACK: Semantically meaningful adversarial audio attack," in *USENIX Security*, 2023.

[28] X. Li, C. Yan, X. Lu, Z. Zeng, X. Ji, and W. Xu, "Inaudible adversarial perturbation: Manipulating the recognition of user speech in real time," *arXiv preprint arXiv:2308.01040*, 2023.

[29] K. Wang, X. Xu, L. Lu, Z. Ba, F. Lin, and K. Ren, "{FraudWhistler}: A resilient, robust and plug-and-play adversarial example detection method for speaker recognition," in *USENIX Security*, 2024.

[30] K. Li, X. Hu, I. Grishchenko, and D. Lie, "Harmonicattack: An adaptive cross-domain audio watermark removal," *arXiv*, 2025.

[31] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[32] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *IEEE S&P*, 2020.

[33] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: A query-efficient black-box adversarial attack via random search," in *Proc. ECCV*, 2020, pp. 484–501.

[34] P. O'Reilly, Z. Jin, J. Su, and B. Pardo, "Deep audio watermarks are shallow: Limitations of post-hoc watermarking techniques for speech," *arXiv:2504.10782*, 2025.

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

[36] S.-g. Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, "BigVGAN: A universal neural vocoder with large-scale training," in *Proc. ICLR*, 2023.

[37] X. Li, K. Li, Y. Zheng, C. Yan, X. Ji, and W. Xu, "Safeear: Content privacy-preserving audio deepfake detection," in *ACM CCS*, 2024.

[38] H. Liu, T. Wang, R. Fu, J. Yi, Z. Wen, and J. Tao, "Unifyspeech: A unified framework for zero-shot text-to-speech and voice conversion," *arXiv preprint arXiv:2301.03801*, 2023.

[39] B. C. J. Moore, *An Introduction to the Psychology of Hearing*, 6th ed. Brill, 2012.

[40] J. H. McDermott and A. J. Oxenham, "Music perception, pitch, and the auditory system," *Current Opinion in Neurobiology*, vol. 18, no. 4, pp. 452–463, 2008.

[41] E. W. Large and M. R. Jones, "The dynamics of attending: How people track time-varying events," *Psychological Review*, 1999.

[42] A. D. Patel and J. R. Iversen, "The evolutionary neuroscience of musical beat perception: The ASAP hypothesis," *Front. Syst.*

[43] S. McAdams, S. Winsberg, S. Donnadieu, G. De Soete, and J. Krimphoff, "Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes," *Psychol.*, 1995.

[44] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, 3rd ed. Springer, 2007.

[45] "ITU-R BS.1770-5: Algorithms to measure audio programme loudness and true-peak audio level," ITU, Tech. Rep., 2023.

[46] J. Deng, Y. Chen, and W. Xu, "Fencesitter: Black-box, content-agnostic, and synchronization-free enrollment-phase attacks on speaker recognition systems," in *ACM CCS*, 2022.

[47] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Acoust. Speech Signal Process.*, 1984.

[48] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Proc. IEEE ICASSP*, 1979.

[49] C. Valentini-Botinhao, "Noisy reverberant speech database for training speech enhancement algorithms and TTS models," CSTR, University of Edinburgh, Technical Report, 2017.

[50] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Interspeech*, 2017.

[51] M. Dong, X. Yang, Y. Wu, and J.-H. Xue, "Metric learning via maximizing the lipschitz margin ratio," *arXiv*, 2018.

[52] R. Duan, Z. Qu, L. Ding, Y. Liu, and Z. Lu, "Parrot-trained adversarial examples: Pushing the practicality of black-box audio attacks against speaker recognition models," in *NDSS*, 2024.

[53] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Trans. Audio Speech Lang. Process.*, 2011.

[54] G. A. Giudici, "python-stretch: Pitch shifting and time stretching for python," [Online]. Available: https://pypi.org/project/python-stretch/, 2025, accessed: Jun. 9, 2026.

[55] C. Wang, F. Lin, H. Yan, T. Wu, W. Xu, and K. Ren, "{VibSpeech}: Exploring practical wideband eavesdropping via bandlimited signal of vibration-based side channel," in *USENIX Security*, 2024.

[56] Z. Zhang, D. Wang, Q. Yang, P. Huang, J. Pu, Y. Cao, K. Ye, J. Hao, and Y. Yang, "{SafeSpeech}: Robust and universal voice protection against malicious speech synthesis," in *USENIX Security*, 2025.

[57] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[58] E. Skovenborg and S. H. Nielsen, "Evaluation of different loudness models with music and speech material," in *AES*, 2004.

[59] Y. Jadoul, B. Thompson, and B. De Boer, "Introducing parselmouth: A python interface to praat," *Journal of Phonetics*, 2018.

[60] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proc. IEEE PACRIM*, 1993, pp. 125–128.

[61] J. Foote, M. Cooper, and U. Nam, "Audio retrieval by rhythmic similarity," in *ISMIR*, 2002.

[62] C. Richey, M. A. Barrios, Z. Armstrong *et al.*, "VOiCES corpus," in *Proc. Interspeech*, 2018, pp. 1566–1570.

[63] A. Kulkarni, H. M. Tran, A. Kulkarni, S. Dowerah, D. Lolive, and M. Magimai Doss, "Exploring generalization to unseen audio data for spoofing: Insights from ssl models," in *ISCA*, 2024.

[64] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *NeurIPS*, pp. 17 022–17 033, 2020.

[65] B. Chhaglani, Y. Gao, J. Richter, X. Li, S. Zadissa, T. Pruthi, and A. Lovitt, "Artifree: Detecting and reducing generative artifacts in diffusion-based speech enhancement," in *ICASSP*, 2026.

[66] Y. Jiang, Z. Ning, S. Wang, C. Wang, M. Bi, P. Zhu, L. Xie, and Z. Fu, "Ref-vc: Robust, expressive and fast zero-shot voice conversion with diffusion transformers," 2025.

[67] S.-H. Wang, J. Shi, J. Tian, H. Wu, and S. Watanabe, "Do neural codecs generalize? a controlled study across unseen languages and non-speech tasks," 2026.

[68] W. Wang, Y. Song, and S. Jha, "Generalizable zero-shot speaker adaptive speech synthesis with disentangled representations," 2023.

[69] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *ICLR*, 2021.

[70] L. Juvela and X. Wang, "Collaborative watermarking for adversarial speech synthesis," in *ICASSP*, 2024.

[71] M. K. Singh, N. Takahashi, W. Liao, and Y. Mitsufuji, "SilentCipher: Deep Audio Watermarking," in *Interspeech 2024*.

[72] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[73] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *AISS*, 2018, pp. 99–112.

[74] Y. Hu, Y. Liu, S. Lv, M. Xing, S. Zhang, Y. Fu, J. Wu, B. Zhang, and L. Xie, "DCCRN: Deep Complex Convolution Recurrent Network for Phase-Aware Speech Enhancement," in *Interspeech 2020*.

[75] K. Ito and L. Johnson, "The LJ Speech Dataset," https://keithito.com/LJ-Speech-Dataset/, 2017, [Online]. Accessed: Jun. 9, 2026.

[76] K. Zhou, B. Sisman, R. Liu, and H. Li, "Emotional voice conversion: Theory, databases and esd," *Speech Communication*, 2022.

[77] Y. Choi and B. Lee, "Pansori: ASR corpus generation from open online video contents," in *IEEE Seoul Sect.*, 2018.

[78] E. Salesky, M. Wiesner, J. Bremerman, R. Cattoni, M. Negri, M. Turchi, D. W. Oard, and M. Post, "The multilingual tedx corpus for speech recognition and translation," *arXiv*, 2021.

[79] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," 2015.

[80] A. Kassis and U. Hengartner, "Unmarker: a universal attack on defensive image watermarking," in *2025 IEEE S&P*.

[81] Z. Jiang, J. Zhang, and N. Z. Gong, "Evading watermark based detection of ai-generated content," in *ACM CCS*, 2023.

[82] J. Lin and M. Juarez, "A crack in the bark: Leveraging public knowledge to remove {Tree-Ring} watermarks," in *USENIX*, 2025.

[83] Y. Hu, Z. Jiang, M. Guo, and N. Gong, "A transfer attack to image watermarks," in *International Conference on Learning Representations*, vol. 2025, 2025, pp. 18 101–18 127.

[84] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE S&P workshops*.

[85] H. Liu, Z. Yu, M. Zha, X. Wang, W. Yeoh, Y. Vorobeychik, and N. Zhang, "When evil calls: Targeted adversarial voice over ip network," in *ACM CCS*, 2022.

[86] Q. Wang, B. Zheng, Q. Li, C. Shen, and Z. Ba, "Towards query-efficient adversarial attacks against automatic speech recognition systems," *IEEE TIFS*, 2020.

[87] P. Cheng, Y. Wang, P. Huang, Z. Ba, X. Lin, F. Lin, L. Lu, and K. Ren, "Alif: Low-cost adversarial audio attacks on black-box speech platforms using linguistic features," in *2024 IEEE S&P*.

[88] Z. Fang, T. Wang, L. Zhao, S. Zhang, B. Li, Y. Ge, Q. Li, C. Shen, and Q. Wang, "Zero-query adversarial attack on black-box automatic speech recognition systems," in *ACM CCS*, 2024.

[89] H. Wu, C. Liu, J. Chen, R. Du, K. He, Y. Zhang, C. Wu, T. Zhang, Q. Guo, and J. Zhang, "When translators refuse to translate: A novel attack to speech translation systems," in *USENIX Security*, 2025.

[90] G. Chen, S. Chen, L. Fan, X. Du, Z. Zhao, F. Song, and Y. Liu, "Who is real bob? adversarial attacks on speaker recognition systems," in *2021 IEEE S&P*.

[91] Z. Li, Y. Wu, J. Liu, Y. Chen, and B. Yuan, "Advpulse: Universal, synchronization-free, and targeted audio adversarial attacks via subsecond perturbations," in *2020 ACM CCS*.

[92] Z. Du, Q. Chen, S. Zhang, K. Hu, H. Lu, Y. Yang, H. Hu, S. Zheng, Y. Gu, Z. Ma, Z. Gao, and Z. Yan, "Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens," *arXiv preprint arXiv:2407.05407*, 2024.

[93] S. E. Eskimez, X. Wang, M. Thakker, C. Li, C.-H. Tsai, Z. Xiao, H. Yang, Z. Zhu, M. Tang, X. Tan, Y. Liu, S. Zhao, and N. Kanda, "E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts," in *2024 IEEE Spoken Language Technology Workshop (SLT)*, 2024.

[94] 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*, 2023.

[95] A. Caillon and P. Esling, "Rave: A variational autoencoder for fast and high-quality neural audio synthesis," *arXiv*, 2021.

[96] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "AudioLDM: Text-to-audio generation with latent diffusion models," in *ICML*, 2023.

[97] R. Huang, M. Lam, J. Wang, D. Su, D. Yu, Y. Ren, and Z. Zhao, "Fastdiff: A fast conditional diffusion model for high-quality speech synthesis," in *IJCAI*, 2022.

[98] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *Proc. Inst. Phonetic Sci.*, vol. 17, 1993, pp. 97–110.

[99] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio Speech Lang. Process.*, 2007.

[100] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong *et al.*, "Speechbrain: A general-purpose speech toolkit," *arXiv*, 2021.

[101] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *ICML*, 2023.

[102] P. Arias-Cabarcos, T. Habrich, K. Becker, C. Becker, and T. Strufe, "Inexpensive brainwave authentication: new techniques and insights on user acceptance," in *USENIX security*, 2021.

[103] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *JASA*, 1937.

[104] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[105] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian journal of statistics*, pp. 65–70, 1979.

Appendix A. Feature-variation in Generative Models

We evaluate feature changes on seen and unseen speech using RAVE [95], Mel-VAE [96], HiFi-GAN [64], BigV-GAN [36], FastDiff [97], and DiffWave [69].

Datasets. All models are trained on LJ Speech and evaluated on VoxCeleb1. We randomly sample 500 utterances from each dataset as *seen* and *unseen* data, respectively, and reconstruct each utterance using the corresponding model.

Metrics. We measure output–input variations across four perceptual features using RMS difference [58], F_0 difference [59], [98], MCD [60], [99], and EED [61]. Following prior work [65], [66], all metrics are normalized to $[0, 1]$, with lower values indicating smaller variations.

Appendix B. Academic Models Evaluation

Threshold Evaluation. We use each scheme's default threshold κ; for Collaborative and RobustDNN, we balance FPR and false negative rate (FNR) following [14]. Table 8 reports the thresholds and low FNRs.

TABLE 8: Detection threshold κ for different watermarks.

Metric	Timbre	AudioSeal	WavMark	AudioMarkNet	SilentCipher	Collaborative	RobustDNN
Threshold κ	0.90	0.50	0.50	0.99	0.50	0.50	0.99
FPR	0.00	0.00	0.00	0.00	0.00	0.02	0.00
FNR	0.00	0.00	0.00	0.00	0.03	0.00	0.06

Impact of Fine-Tuning. We analyze perceptual changes during fine-tuning using the same 500 English samples in Table 4. We evaluate four perceptual features using F_0, MCD, RMS, and EED differences, together with STOI.

Results. Fig. 10 (a), (b), (c), and (d) show that all feature deviations decrease during fine-tuning, while STOI improves from 0.82 to 0.90, with most gains occurring within 20k steps. Fig. 10 (f) and (g) visualize the spectrogram changes.

Computation Overhead. We evaluate ScrambleMark at inference time on an NVIDIA RTX 4090 GPU and an Intel Core i9-14900K CPU, using DiffWave fast sampling with six sampling steps. The ScrambleMark model occupies 422 MiB of GPU memory and 546 MiB of system RAM, while the peak GPU memory usage reaches 1,174 MiB for a single input. Generating one candidate takes 0.41 s per candidate and about 5 s for ten-candidate generation and selection.

Appendix C. False Positive Rate Validation of Recovery

A recovery objective should maintain a low FPR to avoid falsely attributing non-target speech to the target watermark.

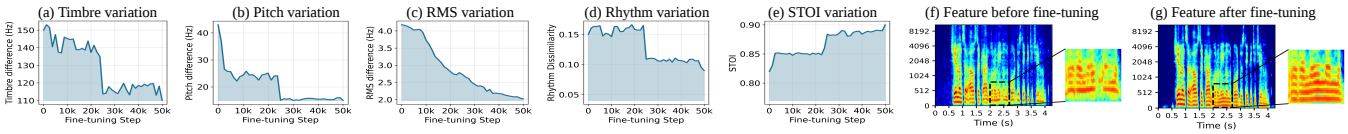


Figure 10: Perceptual feature variation across fine-tuning steps, and harmonic features before and after fine-tuning.

Setup. For each scheme, we randomly select 500 clean samples and generate target and non-target watermarked versions. We apply each recovery objective using 500 equal-width frequency bands and report the resulting FPR in Table 9. Collaborative supports only binary watermark-presence detection and thus cannot evaluate non-target messages; its entries are marked as "–".

TABLE 9: FPR across recovery objectives.

Objective	Type	AudioS.	WavM.	AudioM.	Silent.	Colla.	Robust.
Percp. Recovery	clean	0.00	0.00	0.00	0.00	0.00	0.00
	untarget	0.00	0.00	0.00	0.00	-	0.00
Bit Recovery	clean	0.00	0.00	0.00	0.00	0.00	1.00
	untarget	0.00	0.00	0.00	0.00	-	1.00

Results. Table 9 shows near-zero FPR for perceptual recovery across all schemes, while bit recovery yields 1.00 FPR on RobustDNN. We therefore use perceptual recovery by default and bit recovery only when it maintains low FPR.

Appendix D. Robustness and Sensitivity.

D.1. Statistical Robustness

We evaluate the stability of ScrambleMark's performance under random factors by independently repeating the entire pipeline 15 times, while fixing the evaluation samples and deterministic configurations across runs. We use the test sets from Tables 4 and 5 for academic and commercial models, respectively.

TABLE 10: ASR and STOI statistics over 15 runs.

Metric	Statistic	AudioSeal	WavMark	AudioMark.	SilentC.	Collab.	Robust.	Timbre*	SynthID	Resemble	TrustedA.
ASR	Mean	1.00	1.00	0.98	1.00	0.94	0.87	1.00	0.90	0.98	1.00
	Variance	0.0000	0.0000	0.0002	0.0000	0.0005	0.0005	0.0000	0.0007	0.0003	0.0000
STOI	Mean	0.90	0.91	0.89	0.91	0.90	0.89	0.92	0.91	0.90	0.90
	Variance	0.0001	0.0002	0.0002	0.0003	0.0002	0.0003	0.0003	0.0002	0.0002	0.0003

Results. As shown in Table 10, ScrambleMark achieves mean ASRs of 0.87–1.00 across academic models and commercial platforms, while the ASR variance remains below 0.001 in all cases. The mean STOI scores range from 0.89 to 0.92, with even smaller variances of 0.0001–0.0003. Overall, ScrambleMark maintains stable performance across independent runs of its randomized pipeline.

D.2. Detector Threshold Sensitivity

Beyond the fixed thresholds in Table 4, we evaluate ScrambleMark across different thresholds to assess sensitivity. One way to make the detector more challenging for ScrambleMark is to increase the FPR by lowering the detection threshold. As more clean audio is classified as watermarked, ScrambleMark-processed audio also becomes more likely to be detected as watermarked. We progressively lower each detector's threshold by reducing the required matching bits to obtain the FPR ranges in Table 11. The original thresholds in Table 8 serve as the baseline.

TABLE 11: Threshold sensitivity across FPR levels.

FPR	AudioSeal		AudioMarkNet		SilentCipher		Collaborative		RobustDNN		Timbre*		Avg.
	κ	ASR	κ	ASR	κ	ASR	κ	ASR	κ	ASR	κ	ASR	ASR
Baseline	0.500	1.00	0.990	1.00	0.500	1.00	0.500	0.96	0.990	0.86	0.900	1.00	0.97
1-2%	0.100	1.00	0.688	0.97	0.475	0.97	0.500	0.96	0.984	0.86	0.700	0.99	0.96
2-5%	0.050	0.98	0.625	0.95	0.450	0.94	0.107	0.95	0.977	0.84	0.600	0.95	0.94
5-8%	0.020	0.92	0.563	0.91	0.425	0.93	0.004	0.91	0.965	0.82	0.500	0.93	0.90
8-15%	0.010	0.92	0.500	0.90	0.400	0.92	0.001	0.91	0.953	0.80	0.400	0.92	0.90

Results. As shown in Table 11, relaxing the detection thresholds makes watermark detection more permissive and gradually reduces the average ASR from 0.97 to 0.90. Even at the highest FPR range (8–15%), ScrambleMark still achieves 0.80–0.92 ASR across detectors. For WavMark, the FPR remains 0 even at a zero threshold because it returns no matching score when no watermark is detected [14]; ScrambleMark maintains 1.00 ASR in this case.

D.3. ScrambleMark Parameters

We evaluate ScrambleMark's sensitivity to four parameters: (i) the evasion–quality weight ($\lambda = 0.25, 0.5, 1, 2$), (ii) diffusion schedule steps (30, 40, 50, and 70), (iii) candidate count ($N_\ell = 1, 3, 5, 10$), and (iv) initial band count (1, 50, 100, and 200). We use the same academic and commercial test sets as Tables 4 and 5. For (iii), the 3- and 5-candidate settings use levels approximately spanning the full intensity range. The 1-candidate setting cycles through all ten.

Results. We summarize the key observations from Table 12: (i) Increasing λ improves quality but reduces evasion; $\lambda = 1$ provides a strong balance (0.97 ASR/0.91 STOI), while $\lambda = 0.25$ achieves 0.99/0.81 and $\lambda = 2$ achieves 0.90/0.94. (ii) Increasing diffusion steps improves performance up to 50 steps (0.97/0.91), with negligible gains at 70 (0.97/0.92). (iii) More candidates consistently improve both ASR and STOI, from 0.87/0.81 with one candidate to 0.97/0.91 with ten. (iv) Initial band has little impact (0.95–0.97 average ASR), as optimization converges to approximately 250-300 bands across different initializations, beyond which further splitting minimally reduces the objective loss. Overall, these results characterize the trade-offs of our default settings.

D.4. Multi- and No-Watermark Evaluation

Embedding-Order Study. Multiple watermarks may improve robustness against evasion, but sequential embedding

TABLE 12: Sensitivity analysis of ScrambleMark parameters.

Target	Evasion–quality Trade-off Weight								Diffusion Schedule Steps								Candidate-Count								Initial-Band-Count							
	0.25		0.5		1		2		30		40		50		70		1		3		5		10		1		50		100		200	
	ASR	STOI	ASR	STOI	ASR	STOI	ASR	STOI	ASR	STOI	ASR	STOI	ASR	STOI	ASR	STOI	ASR	STOI	ASR	STOI	ASR	STOI	ASR	STOI	ASR	STOI	ASR	STOI	ASR	STOI
AudioSeal	1.00	0.81	1.00	0.85	1.00	0.91	0.93	0.93	0.96	0.86	0.98	0.88	1.00	0.91	1.00	0.90	0.91	0.80	0.93	0.82	0.97	0.85	1.00	0.91	1.00	0.91	0.99	0.91	1.00	0.89	1.00	0.89
AudioMarkNet	1.00	0.80	0.98	0.83	0.98	0.88	0.90	0.94	0.95	0.86	0.97	0.87	0.98	0.89	0.98	0.90	0.85	0.81	0.89	0.83	0.94	0.83	0.98	0.88	0.98	0.90	0.95	0.90	0.98	0.92	0.98	0.91
SilentCipher	1.00	0.81	1.00	0.86	1.00	0.92	0.91	0.94	0.97	0.87	0.99	0.89	1.00	0.92	1.00	0.91	0.92	0.80	0.93	0.81	0.97	0.86	1.00	0.92	0.99	0.92	0.97	0.91	0.97	0.92	1.00	0.90
Collaborative	1.00	0.82	0.96	0.85	0.96	0.90	0.92	0.93	0.90	0.86	0.93	0.86	0.96	0.90	0.97	0.91	0.88	0.83	0.90	0.83	0.92	0.87	0.96	0.90	0.97	0.91	0.96	0.89	0.97	0.92	0.98	0.91
RobustDNN	0.95	0.80	0.90	0.83	0.86	0.93	0.80	0.94	0.84	0.85	0.84	0.87	0.86	0.93	0.89	0.93	0.79	0.83	0.81	0.85	0.82	0.88	0.86	0.93	0.89	0.88	0.86	0.90	0.86	0.92	0.88	0.90
Timbre*	1.00	0.81	1.00	0.85	1.00	0.91	0.93	0.95	0.97	0.87	0.99	0.88	1.00	0.91	1.00	0.92	0.89	0.82	0.92	0.85	0.97	0.87	1.00	0.91	0.99	0.90	0.97	0.91	1.00	0.90	1.00	0.89
SynthID	0.95	0.81	0.93	0.83	0.93	0.92	0.81	0.94	0.90	0.86	0.88	0.85	0.93	0.92	0.93	0.93	0.78	0.81	0.80	0.84	0.89	0.88	0.93	0.92	0.87	0.91	0.90	0.90	0.87	0.90	0.90	0.91
ResembleAI	1.00	0.83	1.00	0.85	1.00	0.91	0.93	0.94	0.93	0.85	1.00	0.87	1.00	0.91	1.00	0.93	0.88	0.82	0.90	0.83	0.95	0.87	1.00	0.91	0.97	0.91	0.98	0.92	0.97	0.91	1.00	0.90
TrustedAudio	1.00	0.80	1.00	0.82	1.00	0.90	0.95	0.93	1.00	0.86	1.00	0.87	1.00	0.90	1.00	0.92	0.90	0.80	0.95	0.92	0.98	0.88	1.00	0.90	1.00	0.90	1.00	0.92	1.00	0.88	1.00	0.90
Average	**0.99**	**0.81**	**0.97**	**0.84**	**0.97**	**0.91**	**0.90**	**0.94**	**0.94**	**0.86**	**0.95**	**0.87**	**0.97**	**0.91**	**0.97**	**0.92**	**0.87**	**0.81**	**0.89**	**0.84**	**0.93**	**0.87**	**0.97**	**0.91**	**0.96**	**0.90**	**0.95**	**0.91**	**0.96**	**0.91**	**0.97**	**0.90**

can introduce overwriting [16], where later watermarks interfere with earlier ones. We compare two strategies: *Robust-First (RF)*, which places more robust watermarks earlier, and *Weak-First (WF)*, which places less robust ones earlier. We construct three orders for each strategy by varying the order within each robustness group. Using SynthID-watermarked samples from Table 5 as the base, we sequentially embed the remaining nine watermarks; Table 13 lists each order from first to last. Each setting is repeated 10 times with different messages. We report matching scores only for academic models, as commercial scores are inaccessible.

TABLE 13: Matching scores across embedding orders.

Setting	Order (first → last)	10	9	8	7	4	5	3	Avg.
Baseline	–	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
RF-I	1-2-4-3-5-7-6-10-8-9	0.735	1.000	0.958	0.727	0.994	0.959	0.991	0.909
RF-II	1-2-5-4-3-6-7-9-10-8	0.952	1.000	1.000	0.807	0.995	0.931	0.995	0.954
RF-III	1-2-3-5-4-7-6-9-8-10	1.000	1.000	0.961	0.752	0.997	0.946	0.990	0.949
WF-I	1-2-9-10-8-6-7-5-4-3	0.411	0.450	0.603	0.709	0.997	0.977	1.000	0.735
WF-II	1-2-8-9-10-7-6-4-3-5	0.479	0.495	0.579	0.704	0.995	1.000	0.993	0.749
WF-III	1-2-10-8-9-6-7-3-5-4	0.395	0.538	0.582	0.713	1.000	0.983	0.992	0.743

1: SynthID; 2: Resemble AI; 3: AudioMarkNet; 4: RobustDNN; 5: Timbre; 6: TrustedAudio; 7: Collaborative; 8: AudioSeal; 9: WavMark; 10: SilentCipher.

Results. Table 13 shows clear overwriting effects across sequentially embedded watermarks. RF preserves substantially higher average matching scores (0.909–0.954) than WF (0.735–0.749), as weaker watermarks are more vulnerable to subsequent overwriting.

Multi-Watermark Evaluation. We evaluate ScrambleMark against RF-(i)–(iii), with WF-(i)–(iii) and Random as baselines, using the same 42 SynthID samples from Table 5. Each watermark is embedded and verified as detectable before attack; we define All-Watermark ASR (AW-ASR) as the fraction of samples that simultaneously evade all ten detectors and Single-Watermark ASR (SW-ASR) is the fraction of multi-watermarked samples that evade an individual embedded watermark. Each setting is repeated five times.

TABLE 14: ScrambleMark against multiple watermarks.

Setting	Metric	AudioSeal	WavMark	AudioMark.	SilentC.	Collab.	Robust.	Timbre	SynthID	Resemble	TrustedA.
RF	SW-ASR	1.00	1.00	1.00	1.00	1.00	0.91	1.00	0.93	1.00	1.00
WF	SW-ASR	1.00	1.00	0.99	1.00	1.00	0.96	1.00	0.96	1.00	1.00
Random	SW-ASR	1.00	1.00	1.00	1.00	1.00	0.95	1.00	0.95	1.00	1.00

Overall Multi-Watermark Performance

RF		WF		Random	
AW-ASR	STOI	AW-ASR	STOI	AW-ASR	STOI
0.90	0.91	0.95	0.90	0.92	0.90

Results. ScrambleMark achieves AW-ASRs of 0.90, 0.95, and 0.92 under RF, WF, and Random, respectively, with RF providing the strongest defense. Multi-watermark embedding can also weaken individual watermarks through overwriting; for example, SynthID's SW-ASR increases from 0.91 in the single-watermark baseline (Table 10) to 0.93–0.96 across the three multi-watermark embedding orders. These results demonstrate the benefit of structural evasion in transferring across diverse watermark configurations.

No-Watermark. We further evaluate whether applying ScrambleMark to unwatermarked audio unnecessarily degrades speech utility. We randomly select 200 clean samples from each of the five languages in Section 6.1, all verified as unwatermarked by the evaluated detectors, and apply ScrambleMark. We measure STOI, speaker preservation rate (SPR) using SpeechBrain [100], and word error rate (WER) using Whisper large-v3 [101]. SPR measures the fraction of outputs preserving the original speaker, while WER is computed between clean and perturbed audio transcripts.

TABLE 15: Speech utility on unwatermarked audio.

Lang.	B-HSJA			DiffErase-L			DiffErase-M			ScrambleMark		
	STOI	SPR	WER	STOI	SPR	WER	STOI	SPR	WER	STOI	SPR	WER
English	0.81	0.88	11.74%	0.75	0.83	25.19%	0.73	0.80	23.61%	0.91	0.98	0.31%
Chinese	0.83	0.85	13.96%	0.72	0.83	19.90%	0.74	0.79	21.38%	0.92	0.96	0.72%
Korean	0.82	0.85	26.81%	0.71	0.81	27.63%	0.70	0.76	23.27%	0.91	0.99	0.53%
French	0.85	0.87	14.36%	0.72	0.83	23.04%	0.71	0.79	19.91%	0.93	0.97	0.49%
German	0.84	0.82	18.62%	0.70	0.85	20.01%	0.73	0.80	24.55%	0.92	1.00	1.05%
Avg.	**0.83**	**0.85**	**17.10%**	**0.72**	**0.83**	**23.15%**	**0.72**	**0.79**	**22.54%**	**0.92**	**0.98**	**0.62%**

Results. ScrambleMark preserves clean-audio utility, achieving 0.92 STOI, 0.98 SPR, and 0.62% WER on average, compared with 0.72–0.83 STOI, 0.79–0.85 SPR, and 17.10–23.15% WER for the baselines (Table 15). We also observe no false watermark detections on ScrambleMark-processed clean audio across all evaluated models.

D.5. Human-Study Statistical Significance

We test whether ScrambleMark's perceptual advantage over the baselines is statistically significant and consistent across participants. For each platform, eight participants each evaluate 120 clips (30 utterances $\times$ 4 methods); because participant groups differ across platforms, we analyze each platform separately and average the 30 utterance-level ratings within each participant and method to preserve the participant as the unit of analysis. Following [102], we use the Friedman test [103] for overall differences and paired Wilcoxon signed-rank tests [104] to compare ScrambleMark with each baseline, with Holm correction [105] for the three comparisons within each platform. We consider Holm-adjusted $p < 0.05$ statistically significant.

Results. The Friedman tests yield $\chi^2(3) = 24.00$ for both SynthID and TrustedAudio and $\chi^2(3) = 22.20$ for Resemble AI, with all $p < 0.001$. These results indicate highly

consistent rank differences across participants on all three platforms. The pairwise Wilcoxon tests show $W = 0$ and raw $p = 0.0078$ across all comparisons between ScrambleMark and the baselines, indicating that, for each comparison, all participant-level mean differences favor ScrambleMark. After Holm correction within each platform, all comparisons remain significant (Holm-adjusted $p = 0.0234 < 0.05$). Thus, the MOS advantages reported in Table 7 are consistent across participants rather than driven by a small subset.

Appendix E. Ethics Considerations

E.1. Vulnerability Disclosure

We responsibly disclosed ScrambleMark to all affected vendors before public release, providing technical details, verification materials, and source code when requested. All vendors acknowledged our reports and had the opportunity to investigate and mitigate the issue. Google confirmed that the issue may be publicly disclosed.

E.2. Human Subjects Research

Prior to the study, our protocol received institutional IRB approval. Participation was voluntary with informed consent, minimal risk, and no compensation, deception, or personally identifiable information (PII) collection; only anonymized ratings were retained. Participants were recruited online from three universities in the same country and had sufficient English proficiency for the approximately 35-minute task.

Appendix F. Meta-Review

The following meta-review was prepared by the program committee for the 2027 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

F.1. Summary

This paper presents ScrambleMark, a zero-query, watermark-agnostic black-box evasion attack against speech watermarking systems. ScrambleMark disrupts the harmonic structure of watermarked speech to evade detection and is evaluated against both academic and commercial watermarking systems.

F.2. Scientific Contributions

- Identifies an Impactful Vulnerability
- Creates a New Tool to Enable Future Science
- Provides a Valuable Step Forward in an Established Field

F.3. Reasons for Acceptance

1) The paper identifies an impactful vulnerability in modern speech watermarking systems. The authors show that structural perturbations can evade watermark detection without querying the target detector during evasion, and demonstrate the attack against both academic and commercial systems.
2) The paper provides a valuable step forward in an established field. ScrambleMark achieves high attack success while preserving substantially better speech quality than existing generative removal attacks.
3) The paper creates a new tool to enable future science. ScrambleMark provides a concrete attack framework that can be used to study the robustness of speech watermarking systems against structural perturbations and to motivate the development of stronger defenses.