Making Separation-First Multi-Stream Audio Watermarking Feasible via Joint Training

Houmin Sun*
Zi Hu*
Digital Innovation Research Center, Duke Kunshan University
Kunshan, China

Linxi Li
Yechen Wang
Liwei Jin
OfSpectrum, Inc.
Los Angeles, USA

Carsten Maple†
University of Warwick
Coventry, United Kingdom
CM@warwick.ac.uk

Ming Li†
School of Artificial Intelligence, The Chinese University of Hong Kong, Shenzhen,
China
mingli369@cuhk.edu.cn

Abstract

Modern audio is created by mixing stems from different sources, raising the question: can we independently watermark each stem and recover all watermarks after separation? We study a separation-first, multi-stream watermarking framework – embedding distinct information into stems using unique keys but a shared structure, mixing, separating, and decoding from each output. A naive pipeline (robust watermarking + off-the-shelf separation) yields poor bit recovery, showing robustness to generic distortions does not ensure robustness to separation artifacts. To enable this, we study separation-aware watermarking in a controlled verification pipeline, where the separator is part of the detector and can be selected or optimized together with the watermarking system. Experiments on speech+music and vocal+accompaniment mixtures show substantial gains in post-separation recovery while maintaining perceptual quality. [1]

CCS Concepts

• **Computing methodologies** → **Signal processing**; • **Security and privacy** → *Information hiding*.

Keywords

audio watermarking, source separation, multi-stream watermarking, joint training

1 Introduction

Modern audio production increasingly relies on generative models, neural editing tools, and stem-based workflows. A released waveform may therefore combine speech, vocals, accompaniment, or sound effects originating from different models, creators, or rights holders. This raises a practical provenance question: can each source stream carry its own watermark, and can these watermarks still be recovered after the streams are mixed and later separated?

Traditional audio watermarking [1] methods are usually fragile. Existing neural audio watermarking systems are typically designed for single-carrier decoding, where the detector receives the watermarked audio after conventional channel distortions such as noise, filtering, compression, or resampling [13, 21]. This setting does not directly address multi-stream mixtures. If several independently watermarked stems are mixed into one waveform, a verifier must first recover source-specific estimates and then decode the corresponding payload from each separated stem.

A natural solution is to apply a music source separator, such as Demucs [7], TF-GridNet [25], Open-Unmix [22], or Spleeter [10], before watermark decoding. However, source separation is not a lossless transformation. Even when the separated stems are perceptually acceptable, the separator may introduce source leakage, phase inconsistency, spectral smearing, and other structured artifacts that corrupt watermark-bearing cues [6]. As a result, robustness to generic acoustic distortions does not necessarily imply robustness to separation-induced distortions.

In this work, we study separation-first multi-stream audio watermarking as a controlled known-key payload recovery problem. The separator is treated as part of the verification pipeline rather than as an arbitrary unknown pre-processing step, allowing the watermarking system and separator to be selected or optimized together at detection time. We evaluated whether separation-aware training can improve post-separation payload recovery while preserving perceptual quality and separation integrity.

Our contributions are summarized as follows:

- We formulate separation-first multi-stream audio watermarking as a provenance-oriented known-key payload recovery task, where multiple stems in the same mixture carry independent payloads and are decoded after source separation.
- We show that strong neural watermarking baselines degrade severely after source separation, indicating that generic robustness does not translate directly to separation robustness.
- We introduce separation-aware training strategies that expose the watermarking system to the separation channel and substantially reduce post-separation BER while maintaining perceptual quality.
- We analyze separator–watermark compatibility, showing that post-separation watermark recovery depends on the

*Equal contribution.
†Corresponding author.
[1]Demo samples are available at https://anonymous114810.github.io/SF-MSAW-JT/.

match between the separator architecture and the watermarking model, rather than on separator quality alone.

2 Related Work

2.1 Neural Audio Watermarking

Audio watermarking embeds recoverable information into audio while preserving perceptual quality. Classical methods use signal-processing operations such as phase coding, spread-spectrum embedding, and low-bit coding [1]. Recent neural methods, including WavMark [3], MaskMark [15], SilentCipher [21], and AURA [12], improve payload capacity and robustness against common distortions such as noise, filtering, compression, and resampling. These systems are mainly designed for single-carrier or mixture-level decoding. In contrast, our work considers a separation-first setting, where independently watermarked stems are mixed and then separated before per-stem payload recovery.

2.2 Source Separation and Watermark Preservation

Modern source separation systems such as Demucs [7], TF-GridNet [25], Open-Unmix [22], and Spleeter [10] can produce perceptually plausible stems from a mixture. However, separation remains an ill-posed reconstruction problem and may introduce source leakage, phase inconsistency, spectral smearing, and temporal artifacts [6]. These artifacts are particularly harmful to watermark decoding because watermark cues may depend on fine-grained signal structures that are not preserved by separators optimized only for audio reconstruction quality.

2.3 Source Mixing and Separation Robust Steganography

The closest prior work is the source mixing and separation robust audio steganography method of Takahashi et al. [23]. They embed secret messages into individual musical sources, mix them with other sources, apply a pre-trained Demucs separator, and recover the messages from the separated outputs. They also show that multiple sources can carry independent messages, using a time-domain concealer-decoder architecture trained with curriculum learning.

Following the distinction noted by Takahashi et al. [23], steganography emphasizes covert communication, whereas watermarking emphasizes robust ownership verification. We study known-key payload recovery using a shared, key-conditioned AURA backbone with independent keys and payloads across stems. Our focus is separation-aware adaptation of existing watermarking models and its robustness across source domains, acoustic attacks, and separator architectures.

3 Method

3.1 Problem Setup: Separation-First Multi-Stream Watermarking

We consider mixtures formed by multiple stems (e.g., vocal and accompaniment) that may originate from different sources. Each stem carries an independent watermark embedded with a distinct secret key. Given stems $\{x_i\}$, watermarks are embedded independently to obtain $\{\tilde{x}_i\}$. The watermarked stems are mixed into a single waveform:

$$y = \sum_i \tilde{x}_i. \tag{1}$$

A source separator $S(\cdot)$ is applied to the mixture to obtain separated estimates $\{\hat{x}_i\} = S(y)$. The corresponding watermark is then decoded independently from each separated stem $\hat{x}_i$.

3.2 Watermarking Backbone

We adopt AURA [12] as our watermarking backbone, which utilizes a FiLM-conditioned [17] Conformer-based [8] encoder for watermark embedding and a RobustDNN-style [16] convolutional decoder for bit extraction. We also tested other neural watermarking frameworks, including AudioSeal [19] and WavMark [3], but only selected AURA for joint training because it provided the best perceptual quality among the candidates under our preliminary separation-first evaluation.

3.3 Source Separation Network

We use Demucs [7] as the primary source separation backbone in our framework. Demucs is a time-domain separator and is therefore well aligned with our separation-first watermarking setting, where preserving fine temporal structures is important for post-separation payload recovery. We adopt the standard Demucs architecture without modification, so that the observed improvements can be attributed to the proposed separation-aware training pipeline rather than architectural changes to the separator, as detailed in Section 3.5. To examine whether the proposed pipeline generalizes beyond this primary backbone, we further conduct extension experiments with HTDemucs and TF-GridNet, as detailed in Section 4.3.1.

3.4 Impact of Separation on Watermarking

Source separation is inherently lossy [2]. Audio separators often introduce artifacts such as phase misalignment, spectral smearing [24], and cross-source interference (e.g., residual vocals in the separated accompaniment track),limitations explicitly acknowledged by the authors of Demucs themselves [6]. These artifacts act as non-stationary, structured noise that is fundamentally different from the generic distortions (e.g., noise addition, compression) against which most watermarking systems are robust. Consequently, a watermark decoder trained only for generic robustness fails to decode correctly from separated stems, as shown in our experiments (Section 4). This mismatch necessitates a co-design approach where the watermarking system learns to be robust to the specific distortion profile introduced by the separator, and the separator is encouraged to preserve features critical for watermark recovery.

3.5 Joint Training Pipeline

To address the robustness gap identified in Section 3.4, we jointly train the watermark encoder/decoder and Demucs through alternating optimization. Each iteration comprises two updates: (1) freeze the separator and back-propagate the watermark loss through it to update the encoder and decoder; (2) freeze the watermark modules

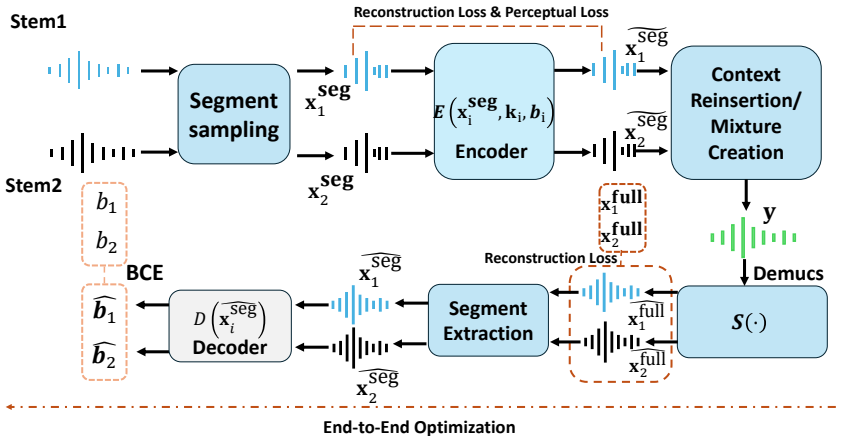


Figure 1: The joint training pipeline for the proposed separation-first audio watermarking framework. In our experiments, $\mathbf{i} = 1, 2$

and update the separator using only the waveform-domain separation loss. Thus, the separator provides a differentiable distortion channel during watermark learning but is not directly updated by the BCE decoding loss.

The pipeline for each training iteration is detailed below and illustrated in Figure 1:

(1) **Segment Sampling and Watermark Embedding.** For each stem (e.g., vocal and accompaniment), we sample a full-length audio clip (approximately 20 seconds). From each clip, a **2-second segment** is cropped to serve as the carrier for watermarking. The 2-second duration represents a standard trade-off in audio watermarking, providing sufficient temporal context ($L = 88200$ samples at 44.1 kHz) for encoding a payload of 32 bits while maintaining computational efficiency for *AURA**. Using our key-conditioned encoder $E(\cdot, \mathbf{k}_i)$, we embed a unique binary watermark $\mathbf{b}_i$ into each stem's segment, conditioned on a distinct, randomly generated secret key $\mathbf{k}_i$. This yields watermarked segments $\tilde{\mathbf{x}}_i^{\text{seg}} = E(\mathbf{x}_i^{\text{seg}}, \mathbf{k}_i, \mathbf{b}_i)$.

(2) **Context Reinsertion.** The watermarked 2-second segments $\tilde{\mathbf{x}}_i^{\text{seg}}$ are reinserted at their original positions in the otherwise unwatermarked 20-second stems $\mathbf{x}_i^{\text{full}}$, producing full-length watermarked stems $\tilde{\mathbf{x}}_i^{\text{full}}$. This preserves the temporal context expected by Demucs and avoids the boundary and context mismatch introduced by separating isolated short segments.

(3) **Mixture Creation.** The full-length watermarked stems $\tilde{\mathbf{x}}_i^{\text{full}}$ are summed to create the mixture $\mathbf{y} = \sum_i \tilde{\mathbf{x}}_i^{\text{full}}$. Prior to summation, we **apply loudness normalization to each stem individually, targeting -16 LUFS (Loudness Units Full Scale)** using the `pyloudnorm` library. This simulates standard audio mixing practice, ensuring a balanced mixture where no single source dominates. A balanced mixture simplifies the initial separation task, leading to cleaner estimates and reducing a source of interference for the watermark decoder.

(4) **Separation and Decoding.** We distinguish the secret conditioning key $\mathbf{k}_i$, which controls the embedding pattern, from the payload bits $\mathbf{b}_i$, which represent the recoverable watermark message. The decoder predicts $\hat{\mathbf{b}}_i$ from each separated stem. The mixture $\mathbf{y}$ is fed into the Demucs separator $S(\cdot)$, which produces estimates for each stem: $\hat{\mathbf{x}}_i^{\text{full}} = S_i(\mathbf{y})$. From these separated full-length tracks, we extract the 2-second slices $\hat{\mathbf{x}}_i^{\text{seg}}$ that are temporally aligned with the originally watermarked segments. Finally, each $\hat{\mathbf{x}}_i^{\text{seg}}$ is passed to the watermark decoder $D(\cdot)$ to recover the predicted bit sequence $\hat{\mathbf{b}}_i = D(\hat{\mathbf{x}}_i^{\text{seg}})$.

(5) **Alternating update.** Finally, the task-specific losses (detailed in Section 3.7) are calculated. Applying the alternating strategy described above, we first back-propagate the watermark loss $\mathcal{L}_{\text{wm}}$ through the entire differentiable pipeline to update E and D, and subsequently use the separation loss $\mathcal{L}_{\text{sep}}$ to update S.

Table 1: Ablation on the MUSDB18 vocal–accompaniment benchmark under the origin condition. BER is measured from temporally aligned segments after Demucs separation.

Training strategy	Update E, D	Update S	BER (%) ↓
Off-the-shelf cascade	No	No	19.23
Fixed-separator adaptation	Yes	No	4.32
Alternating joint training	Yes	Yes	**1.25**

To identify the source of the recovery improvement, we compare the off-the-shelf cascade, fixed-separator watermark adaptation, and the full alternating joint-training configuration in Table 1. When the separator is kept fixed, back-propagating the watermark loss through its differentiable operations to update the watermark encoder and decoder substantially reduces post-separation BER. This result shows that the primary gain arises from exposing the watermarking modules to the separator-induced distortion path rather than from directly updating the separator with the watermark loss.

The full alternating configuration further reduces BER by updating the separator separately with the waveform-domain separation loss while keeping the watermarking modules fixed. This additional gain is smaller than that obtained from fixed-separator watermark adaptation, indicating that separator adaptation provides a measurable but secondary contribution to watermark recovery.

3.6 Datasets and Training Setup

The watermarking model is pre-trained on approximately 150 hours of instrumental music from Free Music Archive (FMA) [5] and 300 hours of speech from the Emilia [9] dataset. This pre-training phase is conducted on four NVIDIA RTX 4090D GPUs for 48 hours. Additional training uses MUSDB18 [18] dataset. The end-to-end joint training is performed on four NVIDIA L20 GPUs for 72 hours. All audio is resampled to 44.1 kHz, and two-second segments are used as watermark carriers. We use the Adam optimizer with a learning rate of 10^{-4} and initialize from pre-trained checkpoints.

3.7 Loss Functions

Training is driven by a separation loss for Demucs [7] and a multi-task objective for*AURA** [12]:

$$\mathcal{L}_{\text{total}} = 100\mathcal{L}_{\text{bce}} + \lambda_{\text{stft}}\mathcal{L}_{\text{stft}} + \lambda_{\text{spec}}\mathcal{L}_{\text{spec}} + \mathcal{L}_{\text{perc}} \tag{2}$$

Here, $\mathcal{L}_{\text{bce}} = \text{BCE}(\mathbf{b}, \hat{\mathbf{b}})$ ensures reliable bit recovery. $\mathcal{L}_{\text{stft}}$ is a multi-resolution log-magnitude STFT loss across $K=3$ resolutions (fft sizes: 1024, 2048, 512): $\mathcal{L}_{\text{stft}} = \frac{1}{K}\sum_{k=1}^{K} |\log(|X_k|) - \log(|\tilde{X}_k|)|_1$, where X_k and $\tilde{X}_k$ are the original and watermarked spectrograms. $\mathcal{L}_{\text{spec}}$ computes the multi-scale Mel-spectrogram L_1 distance. Pre-training follows a two-phase schedule: we initially prioritize bit recovery with minimal spectral weighting and no $\mathcal{L}_{\text{perc}}$. After 20k steps, spectral weights are increased significantly and $\mathcal{L}_{\text{perc}}$ is introduced, maintaining this balance throughout joint training. For imperceptibility, $\mathcal{L}_{\text{perc}}$ combines a BigVGAN discriminator loss [11] and a Noise-to-Mask Ratio loss [14]: $\mathcal{L}_{\text{nmr}} = \frac{1}{CT}\sum_{c,t}(N_{c,t}/M_{c,t})$, where $N_{c,t}$ and $M_{c,t}$ are the noise and masking patterns across C critical bands and T frames, explicitly penalizing distortions that exceed local time–frequency auditory masking thresholds. Crucially, to preserve separation integrity without collapsing the learned watermarking space, the separator is updated *exclusively* via an independent waveform-domain L_1 loss:

$$\mathcal{L}_{\text{sep}} = |\hat{v} - \text{sg}(\tilde{v})|_1 + |\hat{a} - \text{sg}(\tilde{a})|_1 \tag{3}$$

where $\hat{v}, \hat{a}$ are the separated vocal and accompaniment estimates, and $\text{sg}(\cdot)$ is a stop-gradient operation (`detach`) on the watermarked targets.

3.7.1 Attack Simulations. **Basic/Noise:** additive white and peak-normalized pink noise (N, PK); 0.1% random sample suppression (SP); amplitude scaling in $[-1.0, 1.0]$ (AMP) and fixed 20% boosting/ducking (BST, DK); bit-depth quantization (QT); and FFT-based global phase rotation (PHS).

Filter: biquad low-pass filtering with 3–6 kHz cutoffs (LP), band-pass filtering (BF), moving-average smoothing with a 2–10 sample window (SM), and random time–frequency spectrogram masking (SPAUG).

Time/Pitch: resampling down to 16 kHz (RS), a single echo with 100 ms delay (Echo), convolutional reverberation (RV), pitch-preserving speed changes (SPD), speed-preserving pitch shifts (PCH), and combined speed–pitch changes (SPCH).

4 Experiments

Model Variants. We evaluate several watermarking baselines alongside two separation-aware variants. *AURA** denotes our implementation of the published AURA watermarking backbone [12], trained on the datasets described in Section 3.6. *AURA*-Demucs$_{joint}$* denotes the full separation-aware model trained on FMA–Emilia speech+music mixtures. *AURA*-Demucs$_{finetuned}$* denotes the dataset-adapted variant further trained on MUSDB18 vocal/accompaniment mixtures, therefore achieves substantially better BER on the vocal+accompaniment benchmark.

We evaluated WavMark, AudioSeal, *AURA**, and 2 separation-aware variants – *AURA*-Demucs$_{joint}$*, and *AURA*-Demucs$_{finetuned}$*—based both on robustness and perceptual quality. Robustness was quantified by the average bit error rate (BER) under the evaluation pipeline described in Section 4.3, which includes watermark encoding, loudness alignment, mixing, Demucs separation, and per-stem decoding. All baselines are evaluated using their native operating configurations to avoid disadvantaging any method by forced resampling or non-native payload settings. Specifically, WavMark uses its native sampling rate with a 32-bit payload per 1-second window and a randomly sampled key, AudioSeal uses its native sampling rate with a 16-bit payload per 1-second window, and AURA* uses its 44.1 kHz sampling rate with a 32-bit payload per 2-second window and random keys. All methods are then tested under the same separation-first pipeline, with BER computed over each method's decoded payload bits. All methods share the same evaluation settings as the joint training configuration. Test data consist of a 15-hour speech+music set, synthesized from non-overlapping subsets of Emilia [9] and FMA [5], and a 10-hour vocal+accompaniment set extracted from MUSDB18 [18]. All test data were strictly excluded from training datasets.

4.1 Evaluation Metrics

We evaluate audio fidelity using Signal-to-Noise Ratio (SNR) and Scale-Invariant Signal-to-Noise Ratio (SI-SNR) to quantify waveform reconstruction (accounting for gain fluctuations), and ViSQOL [4] for human-perceived quality on a 1–5 Mean Opinion Score (MOS) scale. In order to evaluate the robustness of audio watermarking, we report aggregate BER, computed as the total number of incorrectly decoded bits divided by the total number of decoded bits over all test windows and samples.

4.2 Perceptual Quality and Separation Integrity

We evaluate two complementary aspects of audio fidelity: the imperceptibility of embedded watermarks and the integrity of separated stems after joint training.

Watermark imperceptibility. Table 3 reports watermark fidelity without separation. The Joint and Finetuned models achieve SNRs of 28.77 and 18.56 dB, respectively, with ViSQOL [4] scores above 4.5, indicating high perceptual quality. Their lower fidelity relative to the standalone baseline reflects the additional objective of maintaining watermark recovery after separation.

Separation integrity. A critical question is whether optimizing for watermark robustness degrades the core functionality of the Demucs separator. Table 4 addresses this by comparing separation quality before and after joint training using standard source separation metrics (SDR and SI-SDR). The results demonstrate that separation integrity is well preserved: vocal SDR changes minimally from 9.36 to 8.89, while accompaniment SDR slightly improves from 12.50 to 13.19. We attribute this asymmetry to the

Table 2: Average Bit Error Rate (BER, %) under separation-first evaluation across different benchmarks.

Model	Speech + Music Scenarios				Vocal + Accompaniment Scenarios			
	Origin	Basic/Noise	Filter	Time/Pitch	Origin	Basic/Noise	Filter	Time/Pitch
WavMark	35.45	37.37	38.38	41.69	30.70	31.64	33.53	37.84
AudioSeal	26.88	27.07	29.46	37.00	19.75	20.01	24.82	34.91
*AURA**	15.10	16.15	17.86	35.85	19.23	19.31	20.82	34.86
AURA-Demucs*$_{joint}$	**0.43**	**1.07**	**2.81**	**20.43**	15.21	17.55	19.53	31.68
AURA-Demucs*$_{finetuned}$	1.15	1.82	3.47	21.26	**1.25**	**1.82**	**3.65**	**31.60**

Table 3: Perceptual quality comparison without separation.

Model	SNR	ViSQOL	SI-SNR
*AURA**	30.66	4.72	33.00
AURA-Demucs*$_{joint}$	28.77	4.72	32.80
AURA-Demucs*$_{finetuned}$	18.56	4.51	26.79

Table 4: Separation Performance of Demucs

Timeline	Vocal		Accompaniment	
	SDR	SI-SDR	SDR	SI-SDR
Before	9.36	8.66	12.50	12.30
After	8.89	8.10	13.19	12.98

interplay between watermark preservation and spectral density. The sparser vocal track forces the separator to retain watermark-bearing perturbations, marginally reducing its SDR. Conversely, the complex accompaniment spectrum allows joint training to effectively fine-tune the separator's instrumental representations, slightly improving its SDR. These negligible differences confirm that our framework enhances watermark robustness without compromising the separator's ability to produce high-quality stems.

4.3 Robustness Against Separation

To evaluate whether watermarking methods can preserve decoding accuracy after source separation, we adopt a separation-first evaluation standard aligned with the training pipeline in Section 3.5. Watermarks are embedded independently into two-second segments from distinct constituent sources, which are subsequently mixed and separated via Demucs. The Bit Error Rate (BER) is computed using the temporally aligned segments extracted from each separated stem.

As demonstrated in Table 2, conventional neural watermarking baselines, such as WavMark and AudioSeal, exhibit prohibitively high bit error rates post-separation, with the BER deteriorating to 20–30% across both benchmarks. The vanilla *AURA** provides only marginal improvements, implying that robustness to generic acoustic distortions does not inherently translate to resilience against source separation artifacts.

In contrast, the separation-aware variants substantially improve post-separation decoding compared with the generic watermarking baselines, but their behavior differs across domains. On the speech+music benchmark, both variants achieve low BERs after separation, reducing the origin-condition BER to 0.43% for *AURA*-Demucs*$_{joint}$ and 1.15% for *AURA*-Demucs*$_{finetuned}$. Under additional basic/noise and filtering attacks, both variants remain below 4% BER, whereas the non-separation-aware baselines remain above 15%. This confirms that exposing the watermarking system to the separation channel is crucial for robust post-separation decoding.

On the vocal+accompaniment benchmark, however, the two variants exhibit different behavior. The fine-tuned model achieves the strongest performance, reducing the origin-condition BER to 1.25% and maintaining low BER under basic/noise and filtering attacks. In contrast, the fully joint variant remains less effective in this domain, with a 15.21% BER in the origin condition. This suggests that unrestricted co-adaptation between the watermarking modules and the separator may not always transfer cleanly across source domains, especially for real musical stems with dense harmonic structure and strong source overlap. The fine-tuned setting, which adapts the separator while keeping the watermarking model fixed, appears to better preserve the learned watermark representation in this case.

Overall, these results show that separation-aware training is necessary for reliable multi-stream watermark recovery after source separation, while also indicating that the choice of adaptation strategy is domain-dependent. Cross-stream interference can be substantially mitigated, but achieving uniformly low BER across different mixture types remains an important design consideration. We also observe that the Time/Pitch category remains the most difficult setting. These transformations alter temporal alignment and phase structure, which can simultaneously affect source separation and bit-level watermark decoding. Therefore, although separation-aware training greatly improves robustness under various conditions, robustness to time-scale, pitch transformations remains an open challenge.

4.3.1 Separator Generalization and Domain Compatibility. A pivotal inquiry is whether a watermarking framework co-optimized with a specific separator (e.g., Demucs) can generalize to alternative separation architectures at inference time. As tabulated in Table 5, our empirical cross-testing reveals that separation-aware watermarking is highly coupled with the distortion profiles and signal representations of the training channel. Direct model replacement across completely different separator families induces severe decoding degradation.

To trace the boundaries of this compatibility and understand how different separation paradigms affect watermark survival, we

Table 5: Separator-family generalization under separation-aware joint training.

Separator	Domain	BER (%) ↓
Demucs	time	1.10
HTDemucs	time	1.12
TF-GridNet	complex time-frequency	≈ 50

extend our joint training pipeline to two distinct architectural alternatives: **HTDemucs** [20] (a time-domain, hybrid transformer variation) and **TF-GridNet** (a state-of-the-art frequency-domain separator).

Our empirical results show a clear contrast between the two separator families. The pipeline consistently converges with **HTDemucs**, achieving BER and perceptual quality comparable to the default Demucs setup. In contrast, under the same separation-aware training protocol, repeated runs with **TF-GridNet** did not converge to reliable watermark recovery, with the BER remaining close to chance level (approximately 50%) after multiple training attempts.

We hypothesize that this discrepancy arises from how the separators reconstruct low-energy time-frequency details. AURA* embeds and decodes payloads through subtle magnitude-spectral patterns, which can be treated by a separator as non-source perturbations rather than source-consistent content. TF-GridNet performs complex spectral mapping and explicitly reconstructs source spectra, so such weak watermark-bearing patterns may be suppressed, redistributed across sources, or decorrelated from the decoder's expected structure. In contrast, Demucs-style and HTDemucs separators combine waveform-domain reconstruction with less direct spectral masking, which may better preserve the residual spectral perturbations learned by AURA* under our training protocol. These results suggest that post-separation watermark recovery depends not only on separation quality, but also on whether the separator's reconstruction bias preserves the specific signal structures used by the watermarking model.

4.3.2 Blind Sliding-Window Detection. To evaluate watermark detection without oracle knowledge of the embedding location, we apply the decoder to the entire audio track using a sliding-window search. Stereo audio is first converted to mono by averaging the two channels. We use a 2-second decoding window (88,200 samples at 44.1 kHz) with a stride of 0.5 seconds (22,050 samples), resulting in a 75% overlap between adjacent windows. If regular sliding does not cover the end of a track, an additional end-aligned window is evaluated. Audio shorter than 2 seconds is zero-padded to the required input length.

For each window w_t, the decoder predicts a 32-bit payload $\hat{\mathbf{b}}_t$. Given the registered target payload $\mathbf{b}$, the window-level BER is defined as

$$\mathrm{BER}(w_t) = \frac{d_{\mathrm{H}}\left(\hat{\mathbf{b}}_t, \mathbf{b}\right)}{32}, \tag{4}$$

where $d_{\mathrm{H}}(\cdot,\cdot)$ denotes the Hamming distance. The track-level detection score is the minimum BER over all candidate windows:

Table 6: Blind sliding-window detection under four attack categories.

Category	TP / Pos.	TPR (%)	FP / Neg.	FPR (%)
Origin	1000 / 1000	100.00	0 / 1000	0.00
Basic/Noise	4000 / 4000	100.00	0 / 4000	0.00
Filter	4999 / 5000	99.98	0 / 5000	0.00
Time/Pitch	3706 / 5000	74.12	0 / 5000	0.00

$$\mathrm{BER}_{\mathrm{best}} = \min_t \mathrm{BER}(w_t). \tag{5}$$

A target watermark is declared present when an exact 32-bit match in at least one window. For each attack condition, we evaluate 1,000 watermarked tracks as positive samples and the corresponding 1,000 clean tracks as negative samples.

As shown in Table 6, the detector achieves a 100% TPR under the origin condition and under low-pass filtering, amplitude scaling, pink noise, high-pass filtering, smoothing, boosting, ducking, band-pass filtering, and time stretching. Detection also remains reliable under resampling, spectrogram augmentation, and temporal offset, with TPRs of 99.8%, 99.9%, and 99.2%, respectively. In contrast, speed and combined speed–pitch transformations reduce the TPR to 12.5% and 59.1%. These results identify transformations that substantially modify temporal and pitch structure as the principal detection boundary of the current system.

No false positives are observed in any evaluated condition. Across the 15 attack conditions, this corresponds to zero false detections among 15,000 attack-conditioned negative trials. The mean best BER of clean tracks remains between 0.3745 and 0.3812 across conditions, substantially above the exact-match detection threshold. The macro-average TPR across all conditions is 91.37%, while the observed macro-average FPR is 0%. These results demonstrate that blind sliding-window matching enables reliable track-level verification under most evaluated distortions, while also making the failure boundary under speed-related transformations explicit.

5 Conclusion

We investigated separation-first multi-stream audio watermarking and showed that robustness to common distortions does not necessarily imply robustness to source separation. By incorporating the separator into the training pipeline, our separation-aware models substantially improve post-separation payload recovery while preserving perceptual quality and separation integrity. The results also highlight key limitations: time/pitch transformations remain challenging, and the learned watermarking model does not transfer reliably across arbitrary separator architectures. In particular, compatible time-domain separators such as Demucs and HTDemucs perform better than frequency-domain alternatives in our current pipeline. Future work will explore separator-agnostic watermark cues and extend the framework beyond two-stem mixtures.

Acknowledgments

This research is funded in part by the Guangdong Science and Technology Plan (2023A1111120012).

6 Generative AI Use Disclosure

Large Language Models (LLMs) were used solely for manuscript polishing (e.g., rephrasing and grammar checks) to improve clarity and readability. The LLMs were not used for ideation, methodology, experimental design, data analysis, or result interpretation. All scientific content was produced and verified by the authors.

References

[1] Walter Bender, Daniel Gruhl, Norishige Morimoto, and Anthony Lu. 1996. Techniques for data hiding. *IBM Systems Journal* 35, 3.4 (1996), 313–336.

[2] Estefania Cano, Derry FitzGerald, Antoine Liutkus, Mark D. Plumbley, and Fabian-Robert Stöter. 2019. Musical Source Separation: An Introduction. *IEEE Signal Processing Magazine* 36, 1 (2019), 31–40.

[3] Guangyu Chen, Yu Wu, Shujie Liu, Tao Liu, Xiaoyong Du, and Furu Wei. 2024. WavMark: Watermarking for Audio Generation. *arXiv preprint arXiv:2308.12770* (2024).

[4] Michael Chinen, Felicia S. C. Lim, Jan Skoglund, Nikita Gureev, Feargus O'Gorman, and Andrew Hines. 2020. ViSQOL v3: An Open Source Production Ready Objective Speech and Audio Metric. *arXiv preprint arXiv:2004.09584* (2020).

[5] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. 2016. FMA: A dataset for music analysis. *arXiv preprint arXiv:1612.01840* (2016).

[6] Alexandre Défossez. 2022. Hybrid Spectrogram and Waveform Source Separation. *arXiv preprint arXiv:2111.03600* (2022).

[7] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. 2019. Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed. arXiv:1909.01174 [cs.SD] https://arxiv.org/abs/1909.01174

[8] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented Transformer for Speech Recognition. *arXiv preprint arXiv:2005.08100* (2020).

[9] Haorui He, Zengqiang Shang, et al. 2024. Emilia: An Extensive, Multilingual, and Diverse Speech Dataset for Large-Scale Speech Generation. *arXiv preprint arXiv:2407.05361* (2024).

[10] Romain Hennequin, Anis Khlif, et al. 2020. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software* 5, 50 (2020), 2154.

[11] Sang-gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. 2023. BigVGAN: A Universal Neural Vocoder with Large-Scale Training. *arXiv preprint arXiv:2206.04658* (2023).

[12] Linxi Li, Liwei Jin, Yechen Wang, Houmin Sun, Zi Hu, and Carsten Maple. 2026. AURA: A Stegaformer-Based Scalable Deep Audio Watermark with Extreme Robustness. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 16522–16526. doi:10.1109/ICASSP55912.2026.11462806

[13] Chang Liu, Jie Zhang, et al. 2023. DeAR: A Deep-learning-based Audio Re-recording Resilient Watermarking. *arXiv preprint arXiv:2212.02339* (2023).

[14] Martin Moritz, Toni Olán, and Tuomas Virtanen. 2024. Noise-to-Mask Ratio Loss for Deep Neural Network Based Audio Watermarking. In *Proc. IS2*. 1–6.

[15] Patrick O'Reilly, Zeyu Jin, Jiaqi Su, and Bryan Pardo. 2024. Maskmark: Robust Neuralwatermarking for Real and Synthetic Speech. In *Proc. ICASSP*. 4650–4654.

[16] Kosta Pavlović, Slavko Kovačević, Igor Djurović, and Adam Wojciechowski. 2022. Robust speech watermarking by a jointly trained embedder and detector using a DNN. *Digital Signal Processing* 122 (2022), 103381.

[17] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. 2017. FiLM: Visual Reasoning with a General Conditioning Layer. *arXiv preprint arXiv:1709.07871* (2017).

[18] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. 2017. The MUSDB18 corpus for music separation.

[19] Robin San Roman, Pierre Fernandez, Alexandre Défossez, Teddy Furon, Tuan Tran, and Hady Elsahar. 2024. Proactive Detection of Voice Cloning with Localized Watermarking. arXiv:2401.17264 [cs.SD] https://arxiv.org/abs/2401.17264

[20] Simon Rouard, Francisco Massa, and Alexandre Défossez. 2022. Hybrid Transformers for Music Source Separation. *arXiv preprint arXiv:2211.08553* (2022).

[21] Mayank Kumar Singh, Naoya Takahashi, Weihsiang Liao, and Yuki Mitsufuji. 2024. SilentCipher: Deep Audio Watermarking. In *Interspeech 2024*. doi:10.21437/Interspeech.2024-174

[22] Fabian-Robert Stöter, Stefan Uhlich, et al. 2019. Open-Unmix - A Reference Implementation for Music Source Separation. *Journal of Open Source Software* 4, 41 (2019), 1667.

[23] Naoya Takahashi, Mayank Kumar Singh, and Yuki Mitsufuji. 2022. Source Mixing and Separation Robust Audio Steganography. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[24] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. 2006. Performance measurement in blind audio source separation. *IEEE Trans. Audio, Speech, Lang. Process.* 14, 4 (2006), 1462–1469.

[25] Zhong-Qiu Wang, Samuele Cornell, Shukjae Choi, Younglo Lee, Byeong-Yeol Kim, and Shinji Watanabe. 2023. TF-GridNet: Making Time-Frequency Domain Models Great Again for Monaural Speaker Separation. *arXiv preprint arXiv:2209.03952* (2023).