ONE DETECTOR CAN CATCH THEM ALL: UNIFIED ON-DEVICE AI AUDIO DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advancements in generative AI technologies have lowered the bar to generate human-like audio ranging from voice, music, to sound effects. Today, AI audio can be used maliciously at low cost, from deepfake calls (voice), to streaming fraud (music), and even to fake emergency/disaster scenarios (sound effects). In response, we need the ability to detect AI-generated audio content. However, adoption of AI audio detectors has been sparse largely due to three limitations: 1) existing detectors are specialized for certain types of AI audio (e.g., voice-only), which do not scale when end-users must employ one detector per use case; 2) detectors are vulnerable to audio transformations, like resampling and compression, that are commonly applied before audio reaches the end-user; and 3) more robust detectors require multiple GB of VRAM that device manufacturers are reluctant to give up for AI audio detection, forcing audio data to be sent to third party cloud services. In this paper, we argue that the "ideal" design for AI audio detection should address all three limitations: a *single*, robust, on-device model, that is capable of simultaneously detecting all three kinds of AI audio (voice, music, and sound effect). Our work presents the first step toward realizing a *unified* AI audio detector. In order to accomplish this, we conduct a large-scale measurement study by first collecting and aggregating 15 AI audio benchmarks across voice, music, and sound effect, including AI samples generated by over 230 unique AI audio generators. Our initial results show that unified AI audio detection is possible (AUROC ≥ 0.91), however achieving low false positive rates sacrifices detection performance (30% TPR at 0.5% FPR). To improve existing detectors, we propose a lightweight, model-agnostic, and data-free training harness called RUBATO that leverages previously unused signals (codec and sampling rate) in audio metadata as additional inputs. Experiments show that RUBATO improves detection rate in low false positive regimes by over 13%.

1 INTRODUCTION

Today, generative AI technologies lower the bar for synthetic media creation, allowing users to create realistic text, images, and audio media with little to no effort. While earlier eras of generative models were expensive to use and introduced easily detectable artifacts in synthetic media (such as distorted hands in images, choppy text, low-quality audio), newer models have significantly narrowed the gap between authentic human and synthetically generated media while being incredibly accessible and cheap. These factors have accelerated AI media into a critical problem, both for the online systems that distribute content, and for the end users who consume it.

Look no further than commonly used digital platforms/solutions like email, social media, music, news, and robocalls, where AI-generated text, images, and audio seem to be almost everywhere (Bestvater et al., 2026; Nguyen, 2026; Wu et al., 2026). On Amazon, book catalogs are flooded with AI-generated works (Authors Guild, 2024). The music industry now receives more AI songs than human songs (Wu et al., 2026; Wendel, 2026). Instagram, YouTube, and FaceBook have all acknowledged the widespread presence of "AI-slop," low cost and low effort content that manufacture revenue through sheer quantity (Ricker, 2026; Elias, 2026; Tangermann, 2026). The human-likeness of AI content has also allowed bad actors to execute convincing scams that have caused losses of over $900 million in just the United States alone (Hughes, 2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Responses to address these issues have come slowly, with one such solution centering around AI detection, where machine learning models are trained to distinguish between human and AI-generated content. This approach has been fruitful in several settings. In text detection, several major universities have verified Pangram's effectiveness at detecting LLM usage in academic assignments (Akinode, 2026), and in music, streaming services and music distributors now check submitted songs with AI music detectors (Afchar et al., 2025; IFPI, 2026). While these solutions are being deployed in enterprise and industry settings, widespread adoption of these tools by end users remains challenging. Some concerns include high false positive rates (FPR) (Gaines, 2025), poor generalization (Pudasaini et al., 2026), and privacy (sending data to detectors hosted on third party cloud services) (Alammyan, 2026). However, as the digital world is increasingly overrun by AI media, we believe that detectors are a *necessary* defense against malicious, potentially deceiving, human-like content for end users. And the only way to achieve widespread adoption is through detectors that are robust and deployable directly on-device (e.g., smartphone, personal computer).

Perhaps the most challenging case of this is in AI audio detection. First, AI audio detectors are not "unified." Whereas text and image detectors are capable of detecting AI text and images regardless of the type of text (essays, emails), or image (artwork, photography), existing AI audio detectors can not. Instead, existing detectors detect only one kind of AI audio, like AI deepfake voices or AI music (but not both). Other kinds of AI audio such as sound effects are not considered at all, despite its capabilities to generate realistic background noises (like in fake videos). Second, audio as a continuous analog signal undergoes many complex transformations in digital signal processing before reaching end users, which makes achieving high detection rates challenging when detectors must minimize false positives (falsely accusing humans) (Emi, 2026). Finally, research on deepfake voice audio detection has found that robust detectors need significantly more resources (Li et al., 2026), often requiring GPU computations with multiple Gigabytes of VRAM that device manufacturers are not willing to provide on user devices. This causes detectors to live in the cloud, and force users to send (potentially) private audio data to third party services.

To address these issues, our work is the first that explores a "unified" AI audio detector, which is designed to detect all three types of AI audio, is robust with low false positives, and can be deployed on-device. To do so, we conduct a measurement study on the feasibility of detecting AI voice, music, and sound effects (SFX) simultaneously using a single model. We first collect and aggregate existing datasets of different AI audio types into a single benchmark, with three million samples representing over 230 unique AI audio generators. Generators range from older pre-transformer approaches like HiFi-GAN (Kong et al., 2020), to modern SoTA black box systems like ElevenLabs and Suno. Since AI SFX has never been considered in AI audio detection, we generate our own AI SFX subset. Using our unified benchmark, we train detectors using existing architectures and find that performance struggles against complex transformations and in low FPR regimes (30% TPR), despite promising overall metrics ($\geq$0.91 AUROC). In response, we propose RUBATO, a model-agnostic, data-free, training harness designed to be compatible with any train-from-scratch AI audio detector, by using sampling rate and compression metadata already embedded in audio files, to improve detection performance. Below we list our key findings and contributions:

- Without a unified AI audio benchmark, we cannot build unified detectors. Thus, we combine existing voice and music datasets into a single benchmark with AI audio sourced from >230 generators (+ human bona fide samples). This includes the first AI SFX dataset, which we generate.
- Initial results using existing detection architectures show that a single detector can separate AI audio from bona fide audio for all three AI audio types (AUROC $\geq$ 0.91), but that detection rate drops when minimizing false positives (only 30% TPR @0.5% FPR).
- Our work proposes RUBATO, a model-agnostic and data-free training harness that improves detection performance (+13% TPR) in low FPR regimes by explicitly conditioning detection models on audio channel metadata found in input audio samples (sampling rate and codec compression). We find that this approach allows smaller, device-friendly architectures to match fine-tuned audio foundation models (Wav2Vec2-BERT).

2 BACKGROUND

We begin by providing relevant background in AI audio generation and detection. See Table 1 for explicit comparisons of our work to existing related work.

Table 1: Our work compared to prior art.

Dataset(s)	# of Samples	# of Generators	Voice	Music	SFX
Reimao & Tzerpos (2019)	54k	–	✓	✗	✗
Müller et al. (2024)	508k	175	✓	✗	✗
Wang et al. (2026)	1M	32	✓	✗	✗
Rahman et al. (2025)	97k	2	✗	✓	✗
Go & Kim (2026)	196k	6	✗	✓	✗
Ours	3M	>230	✓	✓	✓

(a) Our benchmark is the first to consider AI voice, music *and* sound effects (SFX), covering >230 text-to-audio/vocoder systems, and >3M total samples.

Work	Tested Transformations			
	Compression	Resampling	Audio Editing	Multiple Transcoding
Sun et al. (2023)	✗	✓	✓	✗
Rahman et al. (2025)	✗	✗	✗	✗
Dugelay et al. (2026)	✗	✓	✓	✗
Wang et al. (2026)	✓	✓	✗	✗
Li et al. (2026)	✓	✗	✓	✗
Ours	✓	✓	✓	✓

(b) We evaluates detectors against all 4 realistic transformations. We introduce multiple transcoding, where audio is transformed 2-5 times consecutively.

2.1 AI Audio Generators

Today's AI audio generators can be broken down into three categories: 1) voice, 2) music, and 3) sound effects, which mirrors the three main categories in digital audio media (The Los Angeles Film School, 2017). Generators typically use the same architectural backbones, like GANs (Kong et al., 2020), and the more modern latent diffusion (Hadjeres et al., 2026). However, current generators are trained only for a specific category (i.e., voice), and cannot be used to for other types (music or SFX). This significantly broadens the threat of AI audio, since malicious users can execute attacks along several fronts, using many different generators. We include examples of such attack variety.

Voice. AI voices generators are primarily used by bad actors in scams, vishing, and deepfake/impersonation attacks (Adaptive Team, 2026). These attacks often occur in time-sensitive operations, like phone calls, where victims have limited time to make a decision.

Music. Major music streaming platforms, like Spotify, are being flooded with millions of AI-generated songs (Wu et al., 2026). These songs are indistinguishable from human-created music (Wendel, 2025), account for over 50% of new song uploads (Wu et al., 2026; Wendel, 2026), and have already been used in multi-million dollar streaming fraud (Singh, 2026).

Sound effects. New tools like Sony's Woosh model now allow users to generate sounds that match provided videos (Hadjeres et al., 2026). As AI-generated videos play a larger role in manipulating real events, like fake disaster footage (McDonell, 2026), realistic audio generated by AI SFX tools will make these attacks even more difficult to spot.

2.2 Detection

Prior work has only studied AI voice detection (Lavrentyeva et al., 2019; Tak et al., 2021b;a; Jung et al., 2022; Li et al., 2026) and AI music detection (Rahman et al., 2025; Afchar et al., 2025; Dugelay et al., 2026) as separate tasks (none of sound effects). However, as we confirm later, these detectors do not generalize across audio types, preventing them from realizing "unified" AI audio detection. Further, existing work (see Table 1b) only evaluate detectors against single transformations, but audio that reaches end users often go through several rounds of degradations and transformations (we call it "multiple transcoding") whose effects remain understudied. Our work fills in the gap along three fronts: 1) we detect all 3 types of AI audio, 2) we evaluate deployment on device, and 3) we study robustness to realistic transformations, including multiple transcoding.

3 Unified AI Audio Benchmark

Our first goal is to train unified AI audio detectors using a dataset that consists of AI and bona fide audio for all three audio types. Prior work measures detection individually across voice and music (no SFX), so no unified benchmark exists. Thus, we create our own by aggregating existing datasets, and generating AI SFX.

3.1 Collection

We describe our collection process at a high-level. More details can be found in Appendix A.1.

Voice. We aggregate eight existing AI voice deepfake datasets (see Table 5), spanning over 224[1] different generators. Crucially, our benchmark includes modern, SOTA, closed-source text-to-voice models ElevenLabs's v3, OpenAI's TTS-1, as well as open source models like XTTS-v2 (Casanova et al., 2024), Ft-TTS (Chen et al., 2024).

Music. Bona fide music is represented by FMA (Defferrard et al., 2017) and MTG-Jamendo (Bogdanov et al., 2019) datasets. AI music is downloaded from web-crawled releases (Smith, 2026; nyuuzyou, 2026) containing songs generated by five close-sourced generators: Suno, Udio, Riffusion, Mureka, and Sonauto[2]. Suno and Udio are the most widely used on Spotify (Wu et al., 2026).

Sound effects (SFX). There are no existing datasets of AI-generated sound effects. Thus, we generate our own using three available open source AI SFX generators: Stability's Stable Audio Open (Evans et al., 2024), TangoFlux (Hung et al., 2026), and Sony's WOOSH (Hadjeres et al., 2026). AI SFX are generated as counterparts to existing bona fide SFX samples from three available datasets: Google's AudioSet (Gemmeke et al., 2017), ESC-50 (Piczak, 2015), and FSD-50k (Fonseca et al., 2022), using text inputs normalized via Beck & Lerch (2026).

We collect over 3M audio samples, including AI samples from over 230 generators. Our benchmark is divided into train/validation/test, reusing splits when available, with more details in Appendix A.1.

3.2 Audio Channel Transformations

By the time audio reaches the user, it is likely to have been encoded, decoded, resampled, and/or edited, potentially even multiple times. To mimic this process, we implement 16 individual transformations and apply them randomly to our benchmark's test set. Each can be broadly categorized as **compression**, **resampling**, or **audio editing**, with breakdowns of our 16 transformations in Appendix A.1. Since prior work only considers the above transformations individually, we also introduce **multiple transcoding**, which applies 2-5 transformations consecutively for any audio sample. E.g., an audio sample may be edited, compressed, then resampled, before running the detector.

4 Exploratory Experiments

We now train unified AI audio detectors using our benchmark. Here, we describe our experiments and report our findings.

4.1 Experiment Details

Model architectures. We consider two types of detector designs: 1) discriminative models trained from scratch, and 2) Self-Supervised Learning (SSL) foundation models fine-tuned to be discriminative models. These are the two most widely used approaches in the AI audio detection (Li et al., 2026; Dugelay et al., 2026). Our experiments evaluate on 5 architectures total (4 trained from scratch, 1 fine-tuned, see Table 2). All use default training parameters provided by publicly released implementations from Li et al. (2026).

Training setup. Every model is trained using our benchmark's training set for binary classification. We adopt the original design/implementations for all detectors, which fixes input audio to a sampling rate of 16 kHz, and randomly clips a 4s audio chunk. Due to the imbalance in audio types (voice, music, SFX) and bona fide/AI labels, we use a balanced data sampler to prevent bias. All models are trained using data augmentations. This is the primary recommendation in the literature to improve detector robustness (Li et al., 2026). Training samples are randomly resampled[3] and compressed using the same setup from §3.2. We leave more difficult audio editing/transcoding transformations for testing to evaluate our model's robustness to unseen transformations.

[1] At least 224 because it is unknown how many generators are present in the Fake-or-Real (FOR) dataset.

[2] We note that existing benchmarks like SONICS (Rahman et al., 2025) were not included because their audio files were down-sampled well below standard music audio quality. They also only contain Suno and Udio generators, which we were able to find higher quality samples of from other sources.

[3] In our implementation, audio may be resampled twice: once for data augmentations, and once again before model inference to standardize. This behavior is intentional, as audio waveforms may not have the same sampling rate as expected by detectors when it reaches the user's device.

Table 2: Baseline results of existing detector architectures applied to our unified AI audio benchmark. Fine-tuning foundation model Wav2Vec2-BERT performs the best (≥0.91 AUROC), but all models struggle at low FPR. Best scores are **bolded and underlined**. Overall results are shaded.

	Model	Dataset											
		Voice			Music			SFX			Balanced Overall		
		EER (↓)	AUROC (↑)	TPR @0.5% FPR (↑)	EER (↓)	AUROC (↑)	TPR @0.5% FPR (↑)	EER (↓)	AUROC (↑)	TPR @0.5% FPR (↑)	EER (↓)	AUROC (↑)	TPR @0.5% FPR (↑)
From Scratch	LCNN (Lavrentyeva et al., 2019)	0.34	0.73	**0.10**	0.14	0.94	0.45	0.15	0.93	0.46	0.21	0.88	0.26
	RawNet2 (Tak et al., 2021b)	0.41	0.63	0.03	0.19	0.90	0.41	0.18	0.90	0.40	0.26	0.82	0.14
	RawGAT-ST (Tak et al., 2021a)	0.31	0.76	0.07	0.14	0.94	0.43	0.16	0.92	0.37	0.20	0.89	0.23
	AAIST (Jung et al., 2022)	0.36	0.72	0.09	0.13	0.95	0.49	0.16	0.91	0.37	0.21	0.87	0.16
Fine-tuned SSL	Wav2Vec2-BERT (Seamless Communication et al., 2023)	**0.17**	**0.91**	0.06	**0.10**	**0.96**	**0.65**	**0.12**	**0.95**	**0.58**	**0.13**	**0.94**	**0.39**

Testing. We evaluate detectors on our benchmark's testing dataset, including voice, music, and SFX. Due to data imbalance, we also introduce a balanced overall subset, which randomly samples the same number of testing samples for each audio type. More details in Appendix A.2.

Metrics. In our deployment scenario, false positives (bona fide audio sample wrongly classified as AI) have significant consequences, and should be minimized. Therefore, we focus our evaluation primarily on **true positive rate at a fixed false positive rate of 0.5%** (Jabarian & Imas, 2025), and include more generic metrics AUROC and Equal Error Rate (EER) for reference. Some may argue that even 0.5% FPR is too high, as millions of input samples would still equate to thousands of false accusations. Thus, we also explore performance at lower FPR regimes in §6.

4.2 Results

Table 2 reports our full results. All models struggle to achieve high TPR at 0.5% FPR across all AI audio subtypes (avg. 24% TPR), but some show promising signs (≥0.91 AUROC). Fine-tuning Wav2Vec2-BERT, a large foundation model, performs the best, but this detector is much larger in size than the 4 from-scratch alternatives (1000x parameters and storage), and consumes significantly more CPU/GPU resources (multiple GB VRAM, others can run on CPU). This hinders its deployment capabilities, especially on consumer devices. Below, we provide more detailed analysis into our experimental results, including performance across different audio types and transformations. Our findings in this section help motivate the methodology proposed for RUBATO next in §5.

Audio type. The leftmost columns of Table 2 show that voice deepfakes are the hardest to detect. Even Wav2Vec2-BERT, the best performing detector overall, achieves only 6% TPR at 0.5% FPR on the voice subset, compared to 65% and 58% for music and SFX, despite ≥0.91 AUROC on all three subsets. Since our voice test set is based primarily off of the challenging ASVSpoof5, which uses 32 generators unseen during training, the low performance on voice is not surprising. In fact, our unified Wav2Vec2-BERT detector performs close to/on-par with existing detectors in the deepfake detection based on AUROC and EER prior to our augmentation framework (Wang et al., 2026). We also confirm this by fine-tuning Wav2Vec2-BERT on only the voice subset of our benchmark, which results in marginally better AUROC (+1%), and equal TPR at 0.5% FPR. However, this model is unable to generalize to AI music and SFX audio, with TPR under 10% for both. Our results highlight one key takeaway: existing detectors do not lack capacity to learn separate features for unified AI audio detection, but degrade in the extreme tails of low false positives.

Audio transformations. We report the impact that the four audio transformations from §3.2 have on detection. Results can be found in Table 3. On average, TPR decreases by more than 11% when transformations are applied for both Wav2Vev2-BERT and trained-from-scratch detectors. The more complex transformations (audio edits, and multiple transcoded audio) exhibit the sharpest decrease in TPR. Notably, simple transformations, such as resampling and compression, have negative effects **even though models see resampling and compression during training**.

At a high level, our initial attempt at training unified AI audio detectors shows promise (AUROC ≥0.91), but real-world deployment remains challenging (only 30% TPR at 0.5% FPR). We find this is because 1) different AI audio types do not share features (generators produce different traces), and 2) audio transformations further degrade detection (-11%), even when models are trained with augmentations. These results revealed a critical bottleneck in training unified AI audio detectors:

Table 3: Audio transformations reduce detection rate. Audio editing and multi transcoding are the hardest (TPR decreased over 10% across all settings). Darker indicates worse performance.

Audio Transformation	Δ TPR @0.5% FPR vs. raw untransformed audio							
	Wav2Vec2-BERT				From Scratch (avg.)			
	Voice	Music	SFX	Overall	Voice	Music	SFX	Overall
Resampling	0.012	-0.006	-0.014	-0.003	-0.001	-0.041	-0.028	-0.024
Compression	-0.043	-0.064	-0.117	-0.075	-0.040	-0.063	-0.091	-0.065
Audio Editing	-0.121	-0.168	-0.304	-0.198	-0.037	-0.115	-0.266	-0.140
Multi Transcoding	-0.163	-0.216	-0.360	-0.246	-0.072	-0.156	-0.290	-0.173

existing approaches perhaps rely too much on identifying AI signals from just the waveform, which is evidently difficult when there are hundreds of generators and challenging transformations.

5 RUBATO: A MODEL-AGNOSTIC TRAINING HARNESS

Our goal is to improve detection performance in low FPR regimes for unified AI audio detection. However, rather than propose an alternative architecture, add more data, or increase model size, we believe there are many valuable signals in existing audio data that are currently unused. The idea is simple: if detectors know more about how audio have been modified/transformed prior to detector inference, they should perform better. This section details how we arrived at two such signals (sampling rate and codec), and how we designed RUBATO, a model-agnostic and data-free harness, for any train-from-scratch detector to leverage the additional signals with little to no additional costs. A high-level overview of RUBATO is detailed in Figure 1.

5.1 IDENTIFYING ADDITIONAL SIGNALS

Looking at signals used by traditional detection approaches, we notice two key components: 1) detectors only consider the audio waveform as model inputs; 2) audio preprocessing fixes sampling rate (i.e., 16 kHz), to create a uniform input size. These generalizations are designed primarily for detecting AI voice, but are insufficient for our unified benchmark. More signal is needed.

Audio channel metadata. While the audio waveform itself is the final "product" in which devices use to play audio media, audio media is almost always sent over the network in a compressed form that the user must decode before recovering a playable waveform. As such, the file must also contain proper metadata for at least two basic properties: **sampling rate**, and **compression algorithm** (codec). RUBATO includes both as *additional inputs* to the detector because 1) they are robust, as audio would be **unplayable** without correctly reported metadata, and 2) a detector made aware of this information should be able to take advantage of the predictable ways sampling rate and codec modify audio waveforms.

High frequency signals. Down-sampling to a fixed rate of 16 kHz removes audio waveform signals. Today, it is common for platforms like YouTube to distribute audio at much higher sampling rates (i.e., 48 kHz), which can capture sound frequencies up to 3x that of 16 kHz. Thus, existing approaches are throwing away critical information in higher frequency bands. RUBATO removes resampling from preprocessing, and allows detectors to capture higher frequency signals. We discuss more on the tradeoffs of capturing high frequency signals in Appendix A.3.

5.2 IMPLEMENTATION

RUBATO is designed to fit on top of existing train-from-scratch detectors so that they can learn richer detection boundaries based on audio container metadata, sampling rate and codec. The goal is to do so with *minimal* architectural changes. Below, we describe our implementation. We note that RUBATO cannot be applied to fine-tuned foundation models (like Wav2Vec2-BERT), as we cannot retrain them from scratch with larger input sizes.

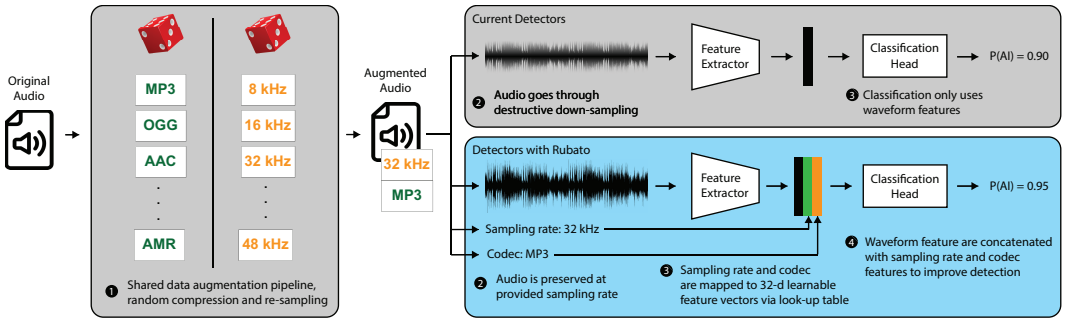

Figure 1: RUBATO (in blue) is a training harness that incorporates audio channel metadata and high frequency signals into existing detectors without changing its architectures or adding new data.

Learned embeddings as additional inputs. To include audio channel metadata as additional model inputs, we create two separate look-up tables: one enumerates the 6 widely used sampling rates (see §3), and the other codecs (6 from §3 + 1 catch-all row for unseen/raw audio). Each enumeration maps to a randomly initialized n-dimensional vector, and is learnable during backpropagation. During model inference, both sampling rate and codec embeddings are concatenated to the final hidden features of the model's output, before a standard classification head (see Figure 1).

High frequency awareness. RUBATO avoids destructive down-sampling by capturing higher frequency audio signals with a widened input (max 48 kHz). All train-from-scratch architectures adapt to larger inputs and avoid significant architectural changes by simply collapsing larger features into a fixed vector size before classification. Thus, RUBATO allows existing detectors to capture high frequency audio signals without making them bigger. Specific implementation details for each architecture can be found in Appendix A.3.

6 EXPERIMENTS WITH RUBATO

We show that RUBATO, a simple model-agnostic training harness, enables smaller and more efficient detector architectures to approach and even match performance of Wav2Vec2-BERT. This enables practical deployment of unified, on-device AI audio detectors without sacrificing detection performance. In this section, we repeat our experiments from §4, and compare baselines performance to detectors trained using RUBATO. After exploring detection results (§6.1), we dive deeper into how RUBATO interacts with unified AI audio detection via ablation studies (§6.2), and finally discuss deployment of unified AI audio detectors (§6.3).

6.1 DETECTION RESULTS

Table 4 shows that RUBATO improves overall detection for all train-from-scratch model architectures. RUBATO even allows LCNN, a small detector with 500k parameters to match Wav2Vec2-BERT, an audio foundation model with over 500M parameters. Here, we dive deeper into our results, including RUBATO's improved robustness, and performance at different FPRs.

Performance across transformations. Previously in §4, we find transformations like audio editing and multi-transcoding significantly degrade detection performance. Figure 3 shows that RUBATO improves detection in all transformation settings for our best performing detector (LCNN). RUBATO beats the baselines even on the two hardest transformations: editing and multi-transcoding. Further, RUBATO continues to outperform baselines even as more transformations are chained in succession via multi-transcoding (see Appendix Figure 8). These results show that the extra signals RUBATO adds are beneficial even when waveforms are degraded and noise levels are high.

Performance at different FPRs. In Figure 4, we also show that LCNN + RUBATO consistently detects more AI audio than standard LCNN even when we vary the fixed false positive rate. The improvement is most notable in low FPR regimes (Δ = +10% even at 0.1% FPR), but still shows benefit when false positives are more tolerable (Δ = +3% at 5% FPR).

Table 4: RUBATO, a model-agnostic, data free training harness improves overall detection performance of all train-from-scratch models, with LCNN matching Wav2Vec2-BERT, a large fine-tuned audio foundation model.

Data	TPR @0.5% FPR (Base / with RUBATO)				
	Train from Scratch				Fine-tuned
	LCNN	RawNet2	RawGAT-ST	AASIST	Wav2Vec2-BERT
Voice	**0.10** / 0.08	**0.03** / **0.03**	**0.07** / 0.06	**0.09** / 0.04	0.06
Music	0.45 / **0.58**	0.41 / **0.49**	0.43 / **0.44**	0.49 / **0.53**	0.65
SFX	0.46 / **0.61**	0.40 / **0.41**	**0.37** / 0.32	0.37 / **0.42**	0.58
Overall	0.26 / **0.39**	0.14 / **0.18**	0.23 / **0.24**	0.16 / **0.29**	0.39

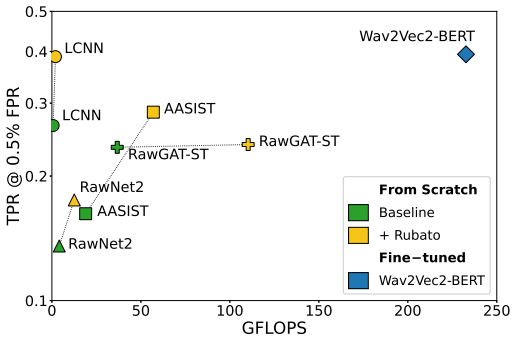

Figure 2: RUBATO improves performance and keeps detectors lightweight (low GFLOPS).

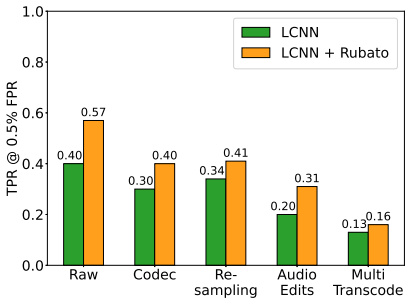

Figure 3: RUBATO improves detection even in the most difficult settings (audio edits and multi-transcode).

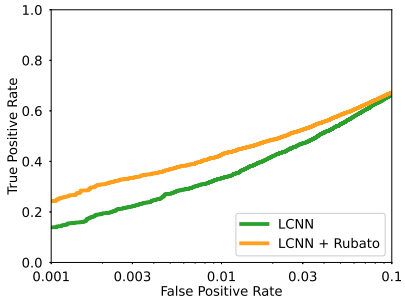

Figure 4: RUBATO especially improves performance at lower false positive rates, which is key for real-world deployments.

We want to stress that RUBATO doesn't change the underlying data (data-free), or model architecture (model-agnostic), yet is able to improve detection performance across all settings nearly for free, by leveraging metadata signals that *already exist* within audio samples. Full results and metrics for RUBATO can be found in Appendix Table 6.

6.2 RUBATO ABLATION STUDIES

Now, we explore the individual impact of RUBATO's different components to understand why it improves detection.

Model input ablations. First, we ablate each individually component of RUBATO on our best performing detector LCNN. Our results find that allowing detectors to attend to high frequency signals yields the largest performance gain (+10%), with learned sampling rate and codec embeddings contributing slightly less (both +8%). In comparison, when RUBATO is implemented with all three components, the total gain is +13%. When available, higher frequency audio signals provide detectors with more evidence, which naturally leads to the highest improvement in precision.

Learned signal embeddings. However, it is less clear why the sampling rate and codec embeddings also benefit, given our baseline models were already trained with random resampling and compression baked into the training augmentation pipeline. Here, we investigate the trainable embeddings by using multidimensional scaling (MDS) to visualize the cosine distance between embeddings for the 6 sampling rates, and 7 codecs. Figures 5-6 reveal natural clustering between rates and codecs. Low sampling rates primarily used for speech are grouped (8-16 kHz), and a similar pattern can be observed with codecs, where generic audio internet streaming codecs (Opus, AAC, Vorbis) are split from speech only codecs (EVS, Speex, AMR). Perhaps even more interesting is that our embedding for raw/uncompressed audio is most similar to Opus, which is the highest quality lossy codec we tested (see Figure 7 in Appendix for hierarchical clustering of codec embeddings). This

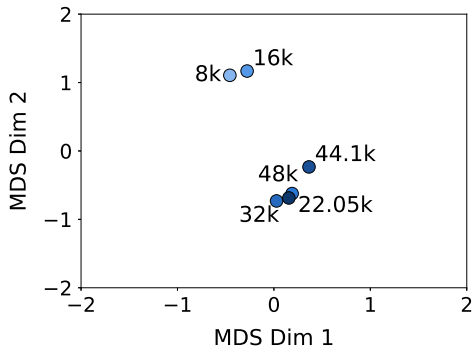

Figure 5: Sampling rate influences AI detection decision primarily based on whether it is lower (8-16 kHz) or higher (22.05 kHz+) fidelity.

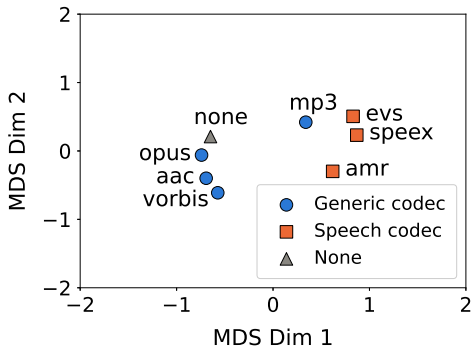

Figure 6: Compression algorithms inject predictable artifacts into waveforms, most notably split by generic audio codecs vs. speech.

means AI detectors are capable of distinguishing low-level noise/features that individual codecs and resampling inject into raw audio, which helps increase confidence/precision in realistic audio media pipelines where audio transformations are unavoidable.

6.3 Deployment

Finally, we evaluate the deployment capabilities of unified AI audio detectors. Figure 2 shows the performance gain that train-from-scratch detectors achieve with sightly extra computational overhead from RUBATO (GFLOPS). This extra computation is largely negligible–our best performing detector LCNN + RUBATO has a real-time factor (RTF) of $<$0.04 on a Snapdragon(R) X Elite laptop for 4 second audio chunks, and requires just 6MB of storage. Full comparisons of RTF, storage, parameter count, and GFLOPS can be found in Appendix Table 7.

7 Conclusion

Modern generative systems synthesize convincing speech, music, and sound effects, creating a need for detectors that operate across audio types rather than address each modality in isolation. In this work, we take the first step toward this goal by studying unified on-device AI audio detection. To enable this study, we construct a unified benchmark containing more than three million audio samples and outputs from over 230 generators, including the first dataset specifically designed for detecting AI-generated SFX. Our evaluation shows that existing detector architectures can succeed in detection with over 0.91 AUROC. However, performance drops sharply at a 0.5% false positive rate, particularly for audio subjected to multiple rounds of transformations. We introduce RUBATO, a lightweight, model-agnostic, and data-free training harness that augments waveform representations with codec and sampling rate information while preserving high-frequency signals that conventional preprocessing discards. Across multiple train-from-scratch architectures, RUBATO improves performance in low-false-positive regimes (up to +13%), enabling compact models to match substantially larger foundation models in performance. Our results suggest that improving AI audio detection does not necessarily require larger architectures or additional training data. Instead, detectors can make better use of information already available in the audio processing and delivery pipeline.

Ultimately, a practical AI audio detector must generalize across content types and generators, remain reliable after audio has passed through real communication and media pipelines, operate at false-positive rates acceptable to end users, and fit within the resource and privacy constraints of consumer devices. This work shows that such a detector is possible, identifies the low-false-positive regime as its central technical bottleneck, and demonstrates that audio-channel context provides a promising path toward unified AI audio detection that are robust and on-device.

AI USE STATEMENT

In this paper, we used generative AI tools to create synthetic AI sound effects as part of our AI audio benchmark. We did not use generative AI tools to design our methodology, clean/reformat our benchmark, or interpret our results. All other required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to create and edit code used in our experiments. We have reviewed all LLM generated code scripts and tested for correctness. We take responsibility for the final artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our study involves impact on two primary stakeholders. 1) For end users who employ AI audio detectors, we acknowledge that these detectors can lead to false positives (falsely accusing human content of being AI), and should not be used as irrefutable evidence that audio is AI generated. These tools are however, extremely useful to provide additional signal into an audio waveform's provenance, where detection outputs should always be corroborated with additional evidence. 2) our benchmark is sourced from publicly available datasets. Thus, in accordance with existing dataset release practices, we will not redistribute existing datasets that are a part of our unified AI audio benchmark. Instead, we will provide code (see reproducibility statement) for users to download existing datasets to reproduce our benchmark. We also decided not to release the AI SFX subset that we generated due to the unclear ownership of AI generated audio. Like with other parts of the benchmark, we will provide code to re-generate this subset.

REPRODUCIBILITY STATEMENT

We are committed to providing reproducible artifacts for the main contributions of our work. Upon acceptance, we will provide a link to our source code, which includes automated scripts to reproduce our benchmark, as well as implementations and training scripts for all of our detection models. This will also include our implementation of RUBATO. Additional details on data processing and RUBATO implementation can be found in the Appendix.

REFERENCES

Adaptive Team. AI deepfake scams: How voice and video impersonation attacks work, real examples, and the defense playbook that stops them. AdaptiveSecurity, Jul. 2026. https://www.adaptivesecurity.com/blog/ai-deepfake-scams.

Darius Afchar, Gabriel Meseguer-Brocal, Kamil Akesbi, and Romain Hennequin. A fourier explanation of AI-music artifacts. In *Proc. of ISMIR*, 2025.

Destiny Akinode. Third-party Pangram evaluations. Pangram, Aug. 2026. https://www.pangram.com/blog/third-party-pangram-evals.

Anangsha Alammyan. Is it safe to paste your essay into an AI detector? (privacy explained). Winston AI, Jun. 2026. https://gowinston.ai/is-it-safe-to-paste-essay-into-ai-detector/.

Authors Guild. AI is driving a new surge of sham "books" on Amazon. Authors Guild News, Mar. 2024. https://authorsguild.org/news/ai-driving-new-surge-of-sham-books-on-amazon/.

Jun Woo Beck and Alexander Lerch. Sound effects dataset unification with the universal category system. In *Proc. of DAFx*, 2026.

Samuel Bestvater, Aaron Smith, Carson Terbush, Chris Baronavski, and Janakee Chavda. How much of the internet is written with AI. Pew Research Center, Aug. 2026. https://www.pewresearch.org/data-labs/2026/08/20/how-much-of-the-internet-is-written-with-ai/.

Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. The MTG-Jamendo dataset for automatic music tagging. In *Machine Learning for Music Discovery Workshop, ICML*, 2019.

Edresson Casanova et al. XTTS: a massively multilingual zero-shot text-to-speech model. In *Proc. of Interspeech*, 2024.

Yushen Chen et al. F5-TTS: A fairytaler that fakes fluent and faithful speech with flow matching. *arXiv preprint arXiv:2410.06885*, 2024.

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. FMA: A dataset for music analysis. In *Proc. of ISMIR*, 2017.

Emile Dugelay et al. Improved robustness in AI-generated music detection. In *Proc. of ISMIR*, 2026.

Jennifer Elias. YouTube chief says 'managing AI slop' is a priority for 2026. CNBC, Jan. 2026. https://www.cnbc.com/2026/01/21/youtube-chief-says-managing-ai-slop-is-a-priority-for-2026-.html.

Bradley Emi. All about false positives in AI detectors. Pangram, Sep. 2026. https://www.pangram.com/blog/all-about-false-positives-in-ai-detectors.

Zach Evans et al. Stable audio open. *arXiv preprint arXiv:2407.14358*, 2024.

Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. FSD50K: an open dataset of human-labeled sound events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.

Lee V. Gaines. Teachers are using software to see if students used AI. what happens when it's wrong? NPR, Dec. 2025. https://www.npr.org/2025/12/16/nx-s1-5492397/ai-schools-teachers-students.

Jort F. Gemmeke et al. Audio Set: An ontology and human-labeled dataset for audio events. In *Proc. of ICASSP*, 2017.

Seonghyeon Go and Yumin Kim. HAIM: Human-AI music datasets for AI music production tracking benchmark. *arXiv preprint arXiv:2606.01686*, 2026.

Gaetan Hadjeres et al. Woosh: A sound effects foundation model. *arXiv preprint arXiv:2604.01929*, 2026.

Keagan Hughes. FBI report highlights $900m lost to AI scams in 2025. Wavy News Stream, Apr. 2026. https://www.wavy.com/news/fbi-report-highlights-900m-lost-to-ai-scams-in-2025/.

Chia-Yu Hung et al. TangoFlux: Super fast and faithful text to audio generation with flow matching and clap-ranked preference optimization. In *Proc. of ICLR*, 2026.

IFPI. Music companies unite in new commitment to combat streaming fraud in order to protect artists, songwriters and fans. IFPI News, Sep. 2026. https://www.ifpi.org/music-companies-unite-in-new-commitment-to-combat-streaming-fraud-in-order-to-protect-artists-songwriters-and-fans/.

Brian Jabarian and Alex Imas. Artificial writing and automated detection. Becker Friedman Institute for Economics, Sep. 2025. https://bfi.uchicago.edu/working-papers/artificial-writing-and-automated-detection/.

Jee-weon Jung et al. AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In *Proc. of ICASSP*, 2022.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Proc. of NeurIPS*, 2020.

Galina Lavrentyeva et al. STC antispoofing systems for the ASVspoof2019 challenge. *arXiv preprint arXiv:1904.05576*, 2019.

Xiang Li, Pin-Yu Chen, and Wenqi Wei. Measuring the robustness of audio deepfake detection under real-world corruption. In *Proc. of CODASPY*, 2026.

Stephen McDonell. China's new challenge as natural disasters strike - fake AI videos. BBC, Jul. 2026. https://www.bbc.com/news/articles/cx27mjvxgg1o.

Nicolas M Müller et al. MLAAD: The multi-language audio anti-spoofing dataset. In *Proc. of IJCNN*, 2024.

Damian Murphy, Helena Daffern, Gavin Kearney, and Jude Brereton. Open acoustic impulse response library (OpenAIR). University of York, 2010. https://www.york.ac.uk/physics-engineering-technology/research/communication-technologies/projects/open-acoustic-impulse-response-library/.

Nicole Nguyen. AI images are everywhere. Here's what they do to our brains, and what we can do. The Wall Street Journal, Aug. 2026. https://www.wsj.com/tech/ai/ai-slop-blurring-reality-podcast-06a9e879.

nyuuzyou. suno. HuggingFace, 2026. https://huggingface.co/datasets/nyuuzyou/suno.

Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proc. of ACM MM*, 2015.

Shushanta Pudasaini, Luis Miralles-Pechuán, David Lillis, and Marisa Llorens Salvador. Why AI-generated text detection fails: Evidence from explainable AI beyond benchmark accuracy. *arXiv preprint arXiv:2603.23146*, 2026.

Md Awsafur Rahman, Zaber Ibn Abdul Hakim, Najibul Haque Sarker, Bishmoy Paul, and Shaikh Anowarul Fattah. SONICS: Synthetic or not – identifying counterfeit songs. In *Proc. of ICLR*, 2025.

Ricardo Reimao and Vassilios Tzerpos. FoR: A dataset for synthetic speech detection. In *Proc. of IEEE SpeD*, 2019.

Thomas Ricker. Instagram cracks down on AI accounts pretending to be human. The Verge, Aug. 2026. https://www.theverge.com/tech/986593/instagram-addresses-fake-ai-profile-slop.

Seamless Communication et al. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*, 2023.

Abhinav Singh. First-of-its-kind AI music fraud: US man bags $8 million in royalties by faking billions of streams. NDTV, Mar. 2026. https://www.ndtv.com/feature/first-of-its-kind-ai-music-fraud-us-man-bags-8-million-in-royalties-by-faking-billions-of-streams-11257499.

Andy Smith. AI music deduplicated. HuggingFace, 2026. https://huggingface.co/datasets/ai-music/ai-music-deduplicated.

Chengzhe Sun, Shan Jia, Shuwei Hou, and Siwei Lyu. AI-synthesized voice detection using neural vocoder artifacts. In *Proc. of CVPR Workshops*, 2023.

Hemlata Tak et al. End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection. *arXiv preprint arXiv:2107.12710*, 2021a.

Hemlata Tak et al. End-to-end anti-spoofing with RawNet2. In *Proc. of ICASSP*, 2021b.

Victor Tangermann. Boomers on Facebook are leaving comments that make it clear that they have no idea that AI-generated thirst traps aren't real. Futurism, Aug. 2026. https://futurism.com/artificial-intelligence/boomers-facebook-ai-thirst-traps.

The Los Angeles Film School. The importance of sound, Jun. 2017. https://www.lafilm.edu/blog/the-importance-of-sound/.

Xin Wang et al. ASVspoof 5: Evaluation of spoofing, deepfake, and adversarial attack detection using crowdsourced speech. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 2026.

Jesper Wendel. Deezer/Ipsos survey: 97% of people can't tell the difference between fully AI-generated and human made music – clear desire for transparency and fairness for artists. Deezer Newsroom, Nov. 2025. https://newsroom-deezer.com/2025/11/deezer-ipsos-survey-ai-music/.

Jesper Wendel. Deezer: AI music has surpassed 50% of new music uploads for the first time. Deezer Newsroom, Jun. 2026. https://newsroom-deezer.com/2026/07/ai-music-exceeds-50-percent-daily-uploads-deezer/.

Stanley Wu, Josephine Passananti, Viresh Mittal, Wenxin Ding, Haitao Zheng, and Ben Y. Zhao. An empirical analysis of AI slop in music streaming. *arXiv preprint arXiv:2606.18052*, 2026.

Table 5: Benchmark composition.

Category	Sub-datasets	Train	Dev	Test	# Bona fide	# AI	# of Generators	Generator(s) / Notes
Voice	ASVspoof5	✓	✓	✓	188,819	815,262	32	8 generators for training, 8 for dev, 16 for val (no overlap)
	MLAAD				0	455,540	175	Open/commercial TTS systems (XTTS-v2, F5-TTS, ..., *etc.*)
	WaveFake	✓			0	117,983	10	Neural vocoders (MelGAN, Parallel WaveGAN, ..., *etc.*)
	LJSpeech	✓			13,100	0	–	Bona fide speech
	M-AILABS (EN)	✓			23,559	0	–	Bona fide speech
	M-AILABS (EN2)	✓			29,088	0	–	Bona fide speech
	Fake-or-Real (FoR)	✓	✓	✓	34,605	6,308	–	Aggregate of multiple commercial/open TTS engines, unknown generators
	In-the-Wild			✓	19,963	11,816	–	Uncurated real-world deepfakes, unknown generators
	SONAR			✓	0	1,074	7	VALL-E, NaturalSpeech3, FlashSpeech, VoiceBox, PromptTTS2, AudioGen, OpenAI TTS
Music	FMA-medium	✓	✓	✓	24,985	0	–	Bona fide music
	MTG-Jamendo	✓	✓	✓	55,701	0	–	Bona fide music
	AI-Music-Deduplicated	✓	✓	✓	0	583,199	4	Mureka, Udio, Riffusion, Sonauto
	Nyuuzyou-Suno	✓	✓	✓	0	464,209	1	Suno
SFX	AudioSet-balanced	✓	✓	✓	22,834	0	–	Bona fide sounds
	ESC-50	✓	✓	✓	2,000	0	–	Bona fide sounds
	FSD50K	✓	✓	✓	33,065	0	–	Bona fide sounds
	AI-SFX-SAO	✓	✓	✓	0	55,377	1	Stable Audio Open
	AI-SFX-TangoFlux	✓	✓	✓	0	55,377	1	TangoFlux
	AI-SFX-Woosh	✓	✓	✓	0	55,377	1	Woosh (Sony AI)

A Appendix

A.1 Benchmark Details

A.1.1 Train / Dev / Test Dataset Split

An overview of how we split our data can be found in Table 5. When existing datasets provide train/dev/test splits, we reuse them.

Voice. Datasets like MLAAD and M-AILABS are used exclusively for training as suggested by the corresponding authors. In-the-Wild and SONAR are used as challenging testing data and not included in training. WaveFake is used for training-only since it predates many of the now popular T2S generators.

Music and SFX. We divide music and SFX data randomly into a 70/20/10 train/dev/test split.

A.1.2 Test Audio Transformations

Compression. We consider 3 speech-specific codecs primarily used for phone calls (AMR, EVS, and Speex), as well as 4 generic audio codecs (MP3, AAC, Vorbis, and Opus) traditionally used to distribute audio media on the internet. Bitrate is varied randomly when available.

Resampling. Audio resampling is commonly used to adapt media to the physical components of specific players and audio streams. Higher sampling rates allow waveforms to capture higher frequency sounds, while the process of down-sampling is a lossy process that destroys high frequency signals. We consider 6 of the most commonly used sampling rates: **8 kHz** (narrowband), **16 kHz** (wideband), **22.05 kHz**, **32 kHz**, **44.1 kHz** (CD Quality), and **48 kHz** (internet streaming).

Audio editing. We also consider more advanced audio editing techniques. These are **1) impulse response (IR) reverb**, which places audio into different environments (i.e., speaking in a cathedral) using the OpenAIRS library (Murphy et al., 2010), **2) pitch shift**, which increases or decreases audio pitch (-4 to +4), and **3) time stretch**, which stretches or squeezes audio along the time domain (0.8x to 1.2x).

In our testing dataset, voice samples see all 16 transformations, while music/SFX see 13 (excluding voice-only codecs AMR, EVS, and Speex).

A.2 Initial Experiment Details

Training augmentations. During training, every audio sample has a 70% chance of being compressed (codec is randomly chosen), with a target sampling rate randomly chosen from our list of 6 sampling rates.

Table 6: Full RUBATO results

	Model	Dataset											
		Voice			Music			SFX			Balanced Overall		
		EER (↓)	AUROC (↑)	TPR @0.5% FPR (↑)	EER (↓)	AUROC (↑)	TPR @0.5% FPR (↑)	EER (↓)	AUROC (↑)	TPR @0.5% FPR (↑)	EER (↓)	AUROC (↑)	TPR @0.5% FPR (↑)
From Scratch	LCNN	0.34	0.73	**0.10**	0.14	0.94	0.45	0.15	0.93	0.46	0.21	0.88	0.26
	LCNN + RUBATO	0.38	0.67	0.08	0.11	0.95	0.58	**0.12**	0.94	**0.61**	0.20	0.89	**0.39**
	RawNet2	0.41	0.63	0.03	0.19	0.90	0.41	0.18	0.90	0.40	0.26	0.82	0.14
	RawNet2 + RUBATO	0.43	0.61	0.03	0.12	0.95	0.49	0.17	0.91	0.41	0.25	0.84	0.18
	RawGAT-ST	0.31	0.76	0.07	0.14	0.94	0.43	0.16	0.92	0.37	0.20	0.89	0.23
	RawGAT-ST + RUBATO	0.35	0.71	0.06	0.14	0.94	0.44	0.18	0.90	0.32	0.23	0.86	0.24
	AASIST	0.36	0.72	0.09	0.13	0.95	0.49	0.16	0.91	0.37	0.21	0.87	0.16
	AASIST + RUBATO	0.34	0.73	0.04	0.11	**0.96**	0.53	0.14	0.93	0.42	0.20	0.89	0.29
Fine-tuned SSL	Wav2Vec2-BERT	**0.17**	**0.91**	0.06	**0.10**	**0.96**	**0.65**	**0.12**	**0.95**	0.58	**0.13**	**0.94**	**0.39**

Testing. Our test set is first divided into audio samples with the same audio type and label (6 subsets). Within each subset, samples are divided equally into 5 parts. These are 1) raw audio without transformations (for simplicity, we assume provided audio is raw) 2) compressed audio, 3) resampled audio, 4) edited audio, and 5) multiple transcoded audio (see §3.2).

A.3 RUBATO IMPLEMENTATION DETAILS

Audio channel metadata embeddings. We use randomized learnable embeddings with dimension size $n = 32$ in our experiments. This adds 680 extra parameters to each model.

Length-agnostic models. For architectures that are inherently length-agnostic (RawGAT-ST, RawNet2, and AASIST), parameter counts remains exactly the same before and after RUBATO's high frequency awareness changes. The only change was in RawGAT-ST, where we had to add a single adaptive 1-dimensional pooling bock (no additional parameters).

Simple linear CNNs. For much older and simpler CNNs, like LCNN, where linear layers have hard-coded sizes, we had to manually configure the model to accept (potentially) larger input sizes. This is done by fixing the model to accept by default features for a 4s waveform at 48 kHz, and zero pad if sampling rate was lower (no energy at higher frequencies). The downstream effect to the model architecture is minimal, since the larger features are collapsed into the same-sized hidden feature before classification head. Unlike the length-agnostic models, this increases the parameter count for LCNN (one singe larger linear layer for collapsing hidden features right before head). However, as we show in §6, the increase in overhead is insignificant, and is overshadowed by the sharp increase in detection performance.

Tradeoffs of capturing higher frequency signals. One could argue that capturing higher frequency noise/signals in AI audio detection is less robust as many use low/high pass filters to remove these signals. While this may have been necessary in small-scale AI detection experiments, where modes could easily overfit to training data, our benchmark is large, and contains enough amount of audio signals that prevent overfitting through sheer scale. Here, removing signals from audio files by down-sampling or filtering removes critical information from audio.

A.4 DETAILED RUBATO RESULTS

Table 6 shows RUBATO's results compared to baseline including reference metrics AUROC and EER. Overall, AUROC and EER are roughly the same before and after applying RUBATO, but TPR at 0.5% FPR increases across all architectures. As we stated in §6, RUBATO doesn't add new data or change the fundamental architecture of these detectors. The additional signals it provides sharpens detectors specifically in low false positive rate regimes, which is critical when considering real world deployment.

Multiple transcoding. We tested all models on multiple transcoding, where audio samples go through multiple rounds of transformations consecutively before classification. As we found in §4 and §6, this is by far the most challenging transformation. In Figure 8, we show chaining more

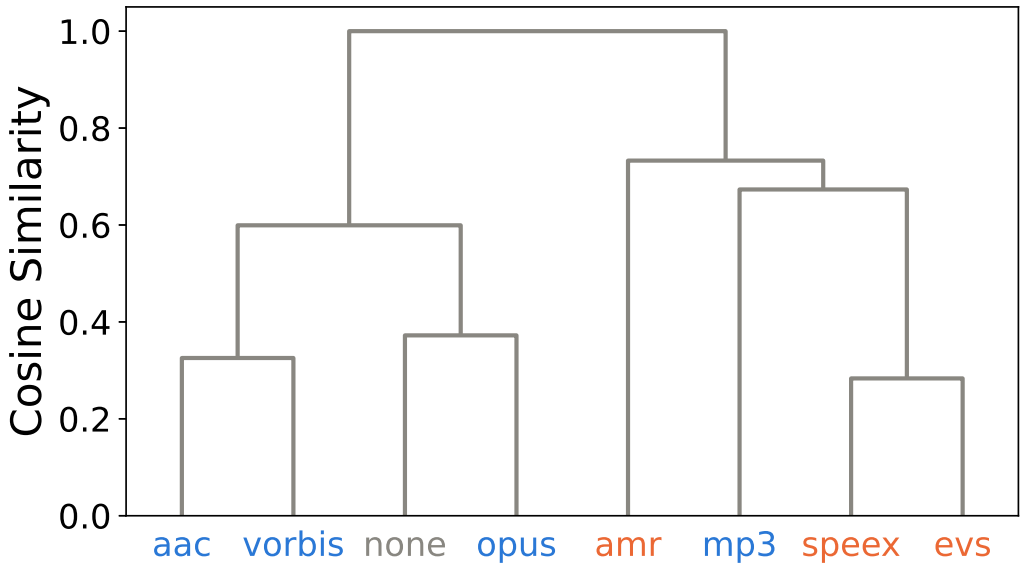

Figure 7: Hierarchical clustering of codec embeddings from our ablation experiments. Distances between codecs reveal natural clustering between generic audio compression (AAC, Vorbis, Opus), and speech compression (AMR, Speex, EVS). Opus, the highest quality codec, is most similar to our "none" embedding, which represents raw/uncompressed audio.

Table 7: Full deployment metrics for all unified detectors.

Model	Metrics				
	# Params	Size (GB)	RTF	GFLOPS	TPR @0.5% FPR
LCNN	562914	0.00211	0.015345	0.685976	0.264891
LCNN + RUBATO	1597634	0.005965	0.039531	1.999928	0.388558
RawNet2	17621410	0.065683	0.06563	4.140811	0.135481
RawNet2 + RUBATO	17624068	0.065695	0.280976	12.577657	0.175052
RawGAT-ST	437034	0.001698	0.550733	36.720003	0.234655
RawGAT-ST + RUBATO	437658	0.001703	1.347652	110.2828	0.238331
AASIST	297866	0.00118	0.256367	18.967616	0.162164
AASIST + RUBATO	298796	0.001186	0.840064	57.004877	0.285337
Wav2Vec2-BERT	581544770	2.166664	0.814722	232.706741	0.393722

transformations together degrades detection on average from 50-60% TPR at 0.5% FPR (raw), to below 20% (5 transformations). However, RUBATO is still better than baselines, which suggests that some sampling rate and codec signals survive even after multiple rounds of degradation.

Learned codec embeddings. Earlier in §4, we showed that the learned features for sampling rate and codecs were naturally clustered, which helps explain why RUBATO improves upon the baseline. Figure 7 further emphasizes this point, and shows that when we used hierarchical clustering on codec embeddings, they also follow natural patterns. Opus, which is the highest quality codec is most similar to raw untransformed audio, while AMR (the oldest codec) is largely separated from newer codecs like MP3, Speex, and EVS. This is further evidence that RUBATO extracts meaningful signals from knowledge of what codecs are used, as these features suggest rich embeddings that help unified audio detectors cut through the noisiness of our large benchmark.

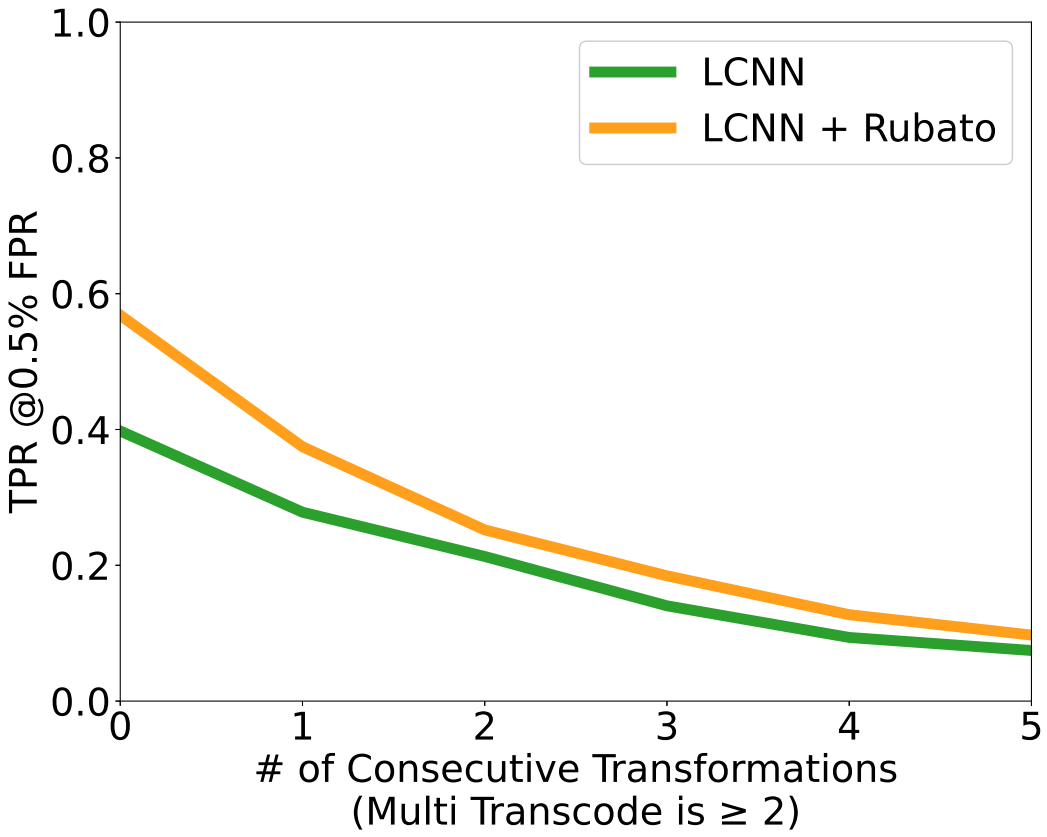

Figure 8: RUBATO improves detection even when transformations are applied consecutively to AI samples.