WHO IS AI SINGING AS? SINGER-TIMBRE MISUSE DETECTION IN AI MUSIC

Anonymous authors
Paper under double-blind review

ABSTRACT

AI-generated music can imitate the vocal identity of real singers, raising a question beyond DeepFake detection: *who is AI singing as?* We study this problem as **real–synthetic singer verification** and identify a substantial **identity gap**: pretrained speaker-verification models that perform well on real singing can fail severely on synthetic vocals. To study this gap, we develop TIMBRETRACE, which adapts pretrained speaker-verification models using mixed real–synthetic supervision. Our analysis shows that pretrained models can retain source-singer characteristics even when converted vocals are perceived as the intended target singer, while post-training shifts representations toward the target singer and improves real–synthetic verification. We evaluate how these gains transfer to Zero-day AI Songs. These results highlight the real–synthetic singer-identity gap as a challenge for singer verification and show that real–synthetic supervision can markedly reduce it.

1 INTRODUCTION

Recent advances in singing voice synthesis have made it increasingly easy to generate songs that imitate real singers. Such songs raise a question beyond generic AI-generated audio detection: *who is AI singing as?* A DeepFake detector determines whether a candidate song is AI-generated (Xie et al., 2024; Zang et al., 2024a; Zhang et al., 2024), but not whose vocal identity it imitates. This distinction matters because an AI-generated song may differ from the target singer's real works in lyrics, melody, and accompaniment while still imitating their vocal identity.

We study this problem as **real–synthetic singer verification**. Given real reference vocals from a target singer and a candidate vocal, the goal is to determine whether they represent the same singer identity. Unlike closed-set singer identification (Desblancs et al., 2024), the target identity is specified through reference recordings. Our setting further crosses the real–synthetic boundary: real references are compared with synthetic vocals. We find a substantial **real–synthetic identity gap**: pretrained speaker- and singer-verification models that perform well on real singing can fail severely on AI-generated vocals. On our benchmark, their real–synthetic AUCs range from 0.36 to 0.58, with several models falling below random ranking despite much stronger real–real verification.

This failure suggests a systematic identity mismatch. A converted vocal originates from a source singer but is transformed toward an intended target singer, potentially retaining residual source-identity cues. Consequently, even when humans perceive the converted vocal as the intended target singer, pretrained models may remain more strongly associated with the source singer. This potential mismatch may explain the poor real–synthetic verification performance of pretrained models.

This gap motivates our central question: **can explicit real–synthetic supervision bridge the mismatch between real references and synthetic target vocals?** To study this question, we construct a benchmark that tracks the source singer, intended target singer, source content, and synthesis condition of each generated vocal. We then adapt pretrained speaker-verification encoders using mixed real and synthetic singer supervision with a standard contrastive triplet objective. By comparing real-only adaptation with mixed real–synthetic adaptation under matched training conditions, we examine whether explicit cross-domain supervision provides benefits beyond ordinary adaptation from speech to singing.

Specifically, our evaluation first measures real–real and real–synthetic verification on held-out source recordings. We further evaluate on Zero-day AI Songs collected online, covering target

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.

singers unseen during post-training and generation checkpoints, model families, and post-processing conditions unknown to us. Together, these evaluations characterize not only how much mixed real–synthetic supervision reduces the identity gap, but also how much of that improvement survives increasingly realistic distribution shifts.

In the broader singer-timbre misuse-detection setting, target-singer verification provides evidence complementary to AI-generation detection. TIMBRETRACE focuses on the former: given real target-singer references, it estimates whether a candidate vocal matches the target identity. We treat AI-generation detection as a separate source of complementary evidence and report its performance and combination with TIMBRETRACE only in the Appendix. In brief, our main study focuses on the **real–synthetic singer-identity gap**: how it arises, whether mixed real–synthetic supervision can reduce it, and how well this improvement transfers to Zero-day AI songs.

2 PRELIMINARIES

2.1 MOTIVATION AND THREAT MODEL

Recent advances in singing voice synthesis and singing voice conversion have made it possible to generate songs that closely imitate the vocal timbre of real singers. Such AI-generated songs are increasingly shared on social-media platforms under labels such as "AI Stefanie Sun" and "AI Jay Chou." When the target singer has not authorized the use of their voice, these songs may create copyright-related, personality-right, and attribution concerns. Existing AI-song detection primarily asks whether a vocal is AI-generated. However, detecting AI generation alone does not answer a second question that is central to singer-timbre misuse: *whose voice is AI singing as?*

We consider an adversary who uses an AI singing voice generation model to synthesize a song that imitates the timbre of a target singer. The generated song may contain lyrics, melody, and accompaniment different from the singer's real works and may be distributed through social-media or music-sharing platforms.

The defender is given a candidate vocal and real reference songs from a claimed target singer, without assuming that the candidate is known to be AI-generated. Given these references, our goal is to determine whether the candidate matches the claimed singer's vocal identity. This task is complementary to AI-generation detection; we focus on target-singer verification and treat DeepFake detection only as complementary evidence analyzed in Section A.3.

Formally, we denote a vocal segment by $\mathbf{x}_{s,i}^{d}$, where $d \in \{\mathrm{r}, \mathrm{ai}\}$ denotes the domain (real or ai), s denotes the singer identity, and i denotes the song segment. Given real reference vocals from a target singer s, our verifier determines whether a candidate vocal represents the same singer identity. We formulate this task as **real–synthetic singer verification**. Unlike conventional singer verification, which compares real recordings, our setting considers identity matching across the real–synthetic boundary. This boundary introduces a substantial domain gap: an AI-generated vocal may preserve residual characteristics of its source singer while perceptually resembling its target singer. Our study therefore focuses on characterizing this real–synthetic identity gap, reducing it through real–synthetic supervision, and evaluating whether the improvement transfers to Zero-day AI songs.

Note that our verifier does not determine whether a candidate is unauthorized, deceptive, or legally infringing. Rather, it provides singer-identity evidence that may support broader provenance, rights-management, moderation, or human-review workflows.

2.2 POSITION TO RELATED WORK

DeepFake AI-song detection. Recent work studied the detection of AI-generated music, including datasets and benchmarks such as FSD, SingFake, CtrSVDD, and SVDD Challenge (Xie et al., 2024; Zang et al., 2024b;a; Zhang et al., 2024; Guragain et al., 2024; Xie et al., 2026; Rahman et al., 2025). These methods mainly determine whether a singing recording is bona fide or AI-generated.

However, they do not determine whose voice is being imitated. An AI song generated with a generic synthetic voice and an AI song imitating a well-known singer may both be classified as DeepFake audio. An existing AI-song DeepFake detector can provide a complementary AI-generation confidence alongside the singer-identity confidence produced by our method.

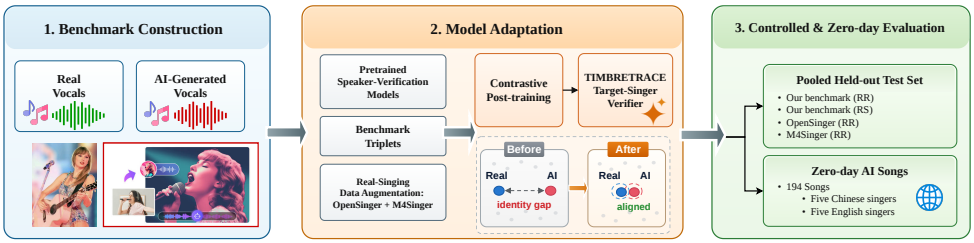


Figure 1: Overview of TIMBRETRACE. We construct a benchmark which supports contrastive post-training of pretrained speaker-verification models, followed by held-out and Zero-day evaluation.

Speak and singer verification. Speaker and singer verification methods determine whether two recordings are produced by the same speaker or singer. Speaker-verification models compare enrollment and trial recordings using identity representations (Snyder et al., 2018; Desplanques et al., 2020; Chen et al., 2022), while singer-verification methods extend identity modeling to singing and cross-song comparisons (Regnier & Peeters, 2012; Torres et al., 2024). These works provide the technical foundation for comparing vocal identity, but AI-generated songs are outside the scope of their study.

More recently, Desblancs et al. (2024) study singer identification in cloned songs, while Salvi et al. (2025) combine DeepFake triage with singer identification. However, neither directly addresses the real-to-synthetic singer-identity gap: Desblancs et al. (2024) evaluate a closed-set classifier trained on real songs, while Salvi et al. (2025) use test-time DeepFake triage to reduce singer-identification EER. In contrast, we explicitly train on real–synthetic singer pairs to bridge this identity gap and evaluate generalization to unseen songs.

AI-song replication and style similarity. Another line of research studies whether AI-generated music copies existing musical works. These methods usually focus on similarities in lyrics, melody, chord progression, rhythm, arrangement, fingerprints, or overall musical style (Batlle-Roca et al., 2024; Yin et al., 2022; Park et al., 2024). These methods characterize relationships between musical works or styles. Our target variable is instead the identity represented by the singing voice.

Overall, our work formulates the problem as real-to-synthetic singer verification: real recordings provide the target-singer reference, and the detector estimates whether a candidate vocal represents the same singer identity. This identity evidence is complementary to, but independent of, evidence that the candidate itself is AI-generated.

3 THE DESIGN OF TIMBRETRACE

TIMBRETRACE is designed to study and reduce the identity-representation gap between real and synthetic singing, consisting of three phases: **benchmark construction**, **model adaption** and **evaluation**. Figure 1 presents the overall workflow as follows.

3.1 BENCHMARK CONSTRUCTION

Phase 1 constructs a benchmark for real–synthetic singer verification. Existing datasets do not provide sufficient real–synthetic comparisons for determining whether an AI-generated song imitates the timbre of a specific singer. We therefore collect real recordings from multiple singers and construct corresponding AI-generated singing samples.

Real and synthetic song construction. We use publicly released singer-specific checkpoints[1] to generate converted vocals from isolated dry lead vocals. We retain non-overlapping 6-second segments containing at least 3 seconds of voiced content in the real source and apply the same temporal boundaries to its converted versions. Checkpoint provenance, vocal preprocessing, and segment selection details are provided in Section A.2.

[1] https://github.com/yxlllc/DDSP-SVC, https://github.com/justinjohn0306/so-vits-svc-4.0

Same singer, different content. A real song i from singer s is paired with an AI-generated song j produced using the checkpoint of singer s, where the AI-generated sample is converted from a different source song (i.e., $i \neq j$). The two recordings therefore share the target singer identity but differ in lyrics and melody. This pair $(x^r_{s,i}, x^{ai}_{s,j})$ is treated as a positive example. These **real–synthetic positive pairs** encourage the detector to match singer timbre without relying on shared song content. We additionally construct **real–real positive pairs** from two different real songs performed by the same singer to preserve the model's ability to recognize singer identity across different songs.

Different singers, different content. A real song from singer s is paired with an AI-generated song using the checkpoint of another singer t, where $s \neq t$, and the two recordings contain different songs. This pair $(x^r_{s,i}, x^{ai}_{t,j})$ is treated as a negative example. These pairs form the primary **real–synthetic negatives**. We also construct **real–real negative pairs** from real songs performed by different singers.

Tuple construction. From the comparison pool, we construct training triplets that supervise singer identity both within and across the real–synthetic boundary, using the four pair types described above: **real–real positive**, **real–synthetic positive**, **real–real negative**, and **real–synthetic negative**. A triplet consists of an anchor, a positive sample with the same singer, and a negative sample with a different singer:

$$\left(x^r_{s,i}, x^{d^+}_{s,j}, x^{d^-}_{t,k}\right), \qquad d^+, d^- \in \{r, ai\}, \quad s \neq t,$$

where the superscripts r and ai denote real and AI-generated vocals, respectively; s and t denote singer identities; and i, j, k denote song segments.

The triplets can be divided into two groups: **real-only triplets** contain exclusively real vocals and provide conventional singer verification (i.e., $d^+ = d^- = r$); **real-synthetic triplets** contain at least one synthetic vocal, teaching the model to associate real singer references with synthetic vocals of the same target singer and distinguish those of different target singers. Triplets are sampled without requiring content alignment. We separately construct content-aligned negatives for the diagnostic analysis in Section 4.3. This decomposition allows us to examine the contribution of real–synthetic supervision. In our benchmark, real-only triplets provide conventional singer-verification supervision, while real-synthetic triplets provide explicit real–synthetic supervision.

3.2 Model Adaption

Phase 2 adapts pretrained speaker-verification models to real–synthetic singer verification. Rather than introducing a new encoder architecture, we use a standard contrastive post-training setup to study the effect of real–synthetic supervision. Our methodology follows three *simple yet effective* design principles.

Speech-to-singing transfer learning. We initialize TimbreTrace from pretrained speaker-verification models (hereafter referred to as **pretrained model**), which provide general speaker-identity representations learned from larger speech corpora. Because speaking and singing share identity-related characteristics of the same voice-production system, these models provide a strong starting point for singer verification while allowing us to study their limitations on synthetic singing.

Formally, f_θ denotes the encoder initialized from the pretrained model. We perform full-parameter post-training, updating all parameters θ to adapt to singing voices and the real–synthetic boundary.

Contrastive post-training. We post-train the encoder using the triplets constructed in Section 3.1. The reference is a real vocal, while the positive and negative samples may be real or AI-generated. Consequently, the benchmark provides both real-only triplets and real–synthetic triplets.

Given a triplet $(x^r_{s,i}, x^{d^+}_{s,j}, x^{d^-}_{t,k})$, we obtain ℓ_2-normalized embeddings using the encoder f_θ:

$$\mathbf{z}^r_{s,i} = \frac{f_\theta(x^r_{s,i})}{\|f_\theta(x^r_{s,i})\|_2}, \qquad \mathbf{z}^{d^+}_{s,j} = \frac{f_\theta(x^{d^+}_{s,j})}{\|f_\theta(x^{d^+}_{s,j})\|_2}, \qquad \mathbf{z}^{d^-}_{t,k} = \frac{f_\theta(x^{d^-}_{t,k})}{\|f_\theta(x^{d^-}_{t,k})\|_2}. \tag{1}$$

We then compute the positive and negative cosine similarities as

$$s^+ = (\mathbf{z}^r_{s,i})^\top \mathbf{z}^{d^+}_{s,j}, \qquad s^- = (\mathbf{z}^r_{s,i})^\top \mathbf{z}^{d^-}_{t,k}. \tag{2}$$

Here, s^+ measures the similarity between the real reference and a positive sample with the same singer identity, whereas s^- measures its similarity to a negative sample with a different singer identity. The positive and negative samples may be either real or AI-generated.

Let s_n^+ and s_n^- denote the positive and negative similarities for the n-th triplet in a batch of size B. The encoder is optimized with a soft-margin triplet contrastive objective:

$$\mathcal{L}_{\text{con}} = \frac{1}{B}\sum_{n=1}^{B} \tau \log\left(1 + \exp\left(\frac{s_n^- - s_n^+ + m}{\tau}\right)\right), \tag{3}$$

where m is the similarity margin, and $\tau > 0$ controls the smoothness of the objective. The loss encourages samples with the same singer to be more similar than samples with different singers, including comparisons that cross the real–synthetic boundary.

Real-only singer supervision. We additionally construct real-only triplets from OpenSinger (Huang et al., 2021) and M4Singer (Zhang et al., 2022): $(x^r_{s,i}, x^r_{s,j}, x^r_{t,k}), i \neq j, s \neq t$. Together with the real-only triplets from our benchmark, these samples provide singer-domain adaptation without synthetic identity supervision and increase singer diversity.

This separation allows us to distinguish ordinary adaptation to real singing from adaptation involving synthetic vocals. In Section 4.3, we therefore compare *real-only* supervision, *real–synthetic* supervision, and their combination under matched training conditions, testing whether explicit real–synthetic supervision provides benefits beyond conventional singer adaptation.

3.3 Controlled and Zero-day Evaluation

We evaluate TimbreTrace on a controlled held-out benchmark as well as Zero-day AI Songs.

Benchmark testing. The held-out test set contains real–real (RR) and real–synthetic (RS) pairs from our benchmark and real–real (RR) pairs from OpenSinger and M4Singer. All test samples are excluded from post-training, and our benchmark is source-recording-disjoint. We report performance for each subset and the pooled test set.

Zero-day AI song testing. We further evaluate on AI songs collected from online platforms such as YouTube. All of the singers and songs are unseen during post-training, while the generation systems and post-processing are unknown. We refer to this setting as *Zero-day AI Song testing*. Each candidate is compared with real references from its target singer using the same preprocessing pipeline. This setting evaluates whether improvements learned from real–synthetic supervision transfer to unseen singers and real-world AI songs.

4 Evaluation

4.1 Experimental Setup

Datasets and Models. We use three complementary datasets: (1) our benchmark from Section 3.1, containing 22 singers, 110 source recordings, and 2,310 AI-generated versions (60,764 segments; 101.27 hours); (2) the real-singing datasets OpenSinger (Huang et al., 2021) (76 singers, 1,133 recordings, 22,050 segments; 34.50 hours) and M4Singer (Zhang et al., 2022) (20 singers, 699 recordings, 15,440 segments; 24.85 hours); and (3) an external set collected from online platforms, with five Chinese-language and five English-language singers absent from post-training. The external set contains 100 real and 94 AI-generated recordings (1,933 six-second segments; 3.22 hours).

We evaluate ten pretrained models. The 16-kHz speaker models are ERes2Net (Chen et al., 2023), CAM++ (Wang et al., 2023), ECAPA-TDNN (Desplanques et al., 2020), WavLM+ECAPA (Chen et al., 2022; Desplanques et al., 2020), and a GE2E-trained speaker encoder (Wan et al., 2018). The 44.1-kHz singer models are CONT, CONT-VC, BYOL, UNIF, and VICReg (Torres et al., 2024), which share an EfficientNet-B0 encoder but use different self-supervised objectives.

Validated on OpenSinger and M4Singer, we select ERes2Net, CAM++, and ECAPA-TDNN for post-training. We fine-tune their full encoders on the same training data with a soft-margin triplet objective ($m = 0.1$, $\tau = 0.07$) and AdamW (learning rate 10^{-5}, weight decay 10^{-4}, batch size 16

Model	SR (kHz)	OpenSinger						M4Singer					
		Different Song			Same Song			Different Song			Same Song		
		EER ↓	AUC ↑	TPR ↑	EER ↓	AUC ↑	TPR ↑	EER ↓	AUC ↑	TPR ↑	EER ↓	AUC ↑	TPR ↑
ERes2Net	16	**8.2**	**0.968**	**75.3**	**4.8**	**0.989**	**90.3**	**10.0**	**0.953**	**56.0**	**5.0**	**0.984**	72.0
CAM++	16	11.1	0.953	67.5	5.8	0.983	85.4	13.0	0.934	39.0	6.0	0.977	**74.5**
ECAPA-TDNN	16	14.6	0.923	58.4	10.1	0.960	75.1	16.3	0.906	44.8	8.5	0.965	74.0
CONT	44.1	19.3	0.883	18.8	14.3	0.929	37.9	22.7	0.847	18.3	15.5	0.931	41.0
CONT-VC	44.1	22.2	0.848	15.8	17.8	0.904	32.6	24.5	0.834	13.3	16.0	0.908	24.5
BYOL	44.1	22.2	0.856	24.7	15.1	0.922	47.9	22.8	0.854	13.3	19.0	0.914	38.5
WavLM+ECAPA	16	24.2	0.834	28.2	17.6	0.910	48.5	18.0	0.881	15.5	10.5	0.940	37.5
UNIF	44.1	24.7	0.817	8.7	20.4	0.868	18.5	29.8	0.756	13.8	22.0	0.853	24.5
VICReg	44.1	32.5	0.737	6.8	29.5	0.781	13.7	37.2	0.698	7.5	31.0	0.782	17.5
GE2E	16	33.2	0.738	14.2	23.9	0.840	34.3	28.7	0.787	12.5	16.0	0.912	51.0

Table 1: Singer-verification performance of pretrained models on OpenSinger and M4Singer. EER and TPR@1%FAR (TPR) are reported in percentages. Bold values indicate the best result in each column; shaded rows indicate the selected models.

triplets). Training lasts 10 epochs, with one epoch of linear warmup followed by cosine learning-rate decay. For CAM++ and ECAPA-TDNN, we freeze batch normalization running statistics but train the affine parameters to stabilize adaptation with small batches. We select checkpoints on the dev set. For comparison, we also train each model with the standard AAM-Softmax loss (Deng et al., 2019) under the same settings instead of our triplet contrastive objective.

Data splits. For our benchmark, we partition data by source recording rather than by segment or verification pair. All real segments and converted versions derived from the same source recording are assigned to the same split. The training, development, and test splits contain 88, 12, and 10 source recordings, respectively. Singer identities are shared across splits; therefore, this setting evaluates transfer to held-out source recordings rather than unseen singers.

For OpenSinger and M4Singer, we use singer-disjoint partitions. The training/development/test splits contain 46/15/15 singers for OpenSinger and 12/4/4 singers for M4Singer. Triplets are constructed exclusively from recordings within their assigned split.

Evaluation Metrics. We evaluate performance using equal error rate (EER), area under the ROC curve (AUC), and true positive rate at a 1% false acceptance rate (TPR@1%FAR); lower EER and higher AUC and TPR@1%FAR are better. For post-trained models, we report the mean over three training seeds, which vary triplet sampling and data order; evaluation pairs are fixed across seeds.

4.2 Experimental Results

Pretrained model selection. We evaluate ten pretrained embedding models on OpenSinger and M4Singer real–real singer verification pairs under different-song and same-song conditions. All models are evaluated without task-specific fine-tuning, with audio resampled to each model's required rate. As shown in Table 1, ERes2Net, CAM++, and ECAPA-TDNN achieve the three lowest EERs and highest AUCs across all four settings. Their different-song EERs are 8.2%, 11.1%, and 14.6% on OpenSinger, and 10.0%, 13.0%, and 16.3% on M4Singer, respectively. ERes2Net has the lowest EER and highest AUC throughout; on same-song pairs on M4Singer, CAM++ has the highest TPR@1%FAR (74.5% versus 74.0% for ECAPA-TDNN and 72.0% for ERes2Net). We select these three models for post-training based on their strong real-singing verification performance.

Benchmark testing. Table 3 reports verification results on four subsets and their pooled evaluation. **RR** denotes real–real verification; **RS** compares a real reference with an AI-generated candidate. Our benchmark provides both RR and RS pairs, while OpenSinger and M4Singer provide RR pairs. We evaluate ten pretrained models and compare our post-training with AAM-Softmax on ERes2Net, CAM++, and ECAPA-TDNN.

Pretrained models perform substantially worse on our RS subset than on the public RR subsets. For example, ERes2Net obtains EERs of 5.91% on OpenSinger and 8.03% on M4Singer, but 45.28% on our RS subset. Across the ten models, RS AUC ranges from 0.3569 to 0.5828, showing that strong real-singing verification does not necessarily transfer to generated singing.

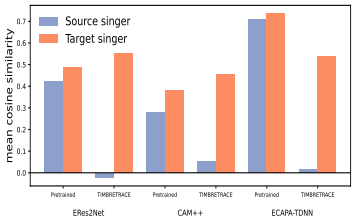


Figure 2: Source–target similarity before and after TIMBRETRACE post-training.

Model	Chinese			English			Overall		
	EER ↓	AUC ↑	TPR ↑	EER ↓	AUC ↑	TPR ↑	EER ↓	AUC ↑	TPR ↑
ERes2Net	19.55	0.8771	24.89	18.20	0.9078	37.22	18.70	0.8898	26.58
+ TIMBRETRACE	**15.64**	**0.9165**	**36.36**	**15.21**	**0.9288**	**40.77**	**16.65**	**0.9153**	**33.82**
CAM++	16.70	0.9046	36.59	18.76	0.8961	40.87	18.84	0.8959	37.08
+ TIMBRETRACE	**16.14**	**0.9213**	**37.61**	**13.49**	**0.9340**	**51.83**	**15.49**	**0.9215**	**39.55**
ECAPA-TDNN	21.02	0.8644	26.70	18.75	0.8911	46.65	20.85	0.8666	28.67
+ TIMBRETRACE	**16.51**	**0.9130**	**35.45**	**14.20**	**0.9309**	**48.48**	**16.61**	**0.9144**	**39.01**

Table 2: Verification on Zero-day AI Songs. Overall metrics are recomputed from pooled within-language pair scores. EER and TPR@1%FAR (TPR) are in %; bold denotes improvement over the pretrained model.

Our post-training improves EER and AUC over the pretrained model on every subset for all three backbones. ERes2Net achieves the lowest pooled EER, decreasing from 21.76% to 7.01%, while AUC rises from 0.8744 to 0.9764. Its largest EER reduction is on our RS subset, from 45.28% to 9.08%, with AUC rising from 0.5563 to 0.9603. ECAPA-TDNN reaches a pooled EER of 7.84% and AUC of 0.9704; CAM++ reduces pooled EER from 22.57% to 10.48%.

AAM-Softmax also improves over pretraining, but our method achieves lower pooled EER on all three backbones: 7.01% versus 8.97% for ERes2Net, 10.48% versus 11.10% for CAM++, and 7.84% versus 10.45% for ECAPA-TDNN. The difference is largest on our RS subset, where our method reduces EER relative to AAM-Softmax from 15.26% to 9.08% for ERes2Net, 19.30% to 14.55% for CAM++, and 15.19% to 10.01% for ECAPA-TDNN. AAM-Softmax performs better on some RR subsets, particularly for CAM++, so our advantage is concentrated in generated singing and the pooled evaluation.

Post-training also improves RR verification over pretraining. For ERes2Net, EER falls from 17.60% to 11.21% on our benchmark (RR), 5.91% to 4.55% on OpenSinger, and 8.03% to 3.70% on M4Singer. Thus, the RS improvement does not come at the expense of the evaluated RR subsets.

Source–target identity analysis. Interestingly, several pretrained models have RS AUC below 0.5, suggesting a systematic mismatch in identity ranking. We hypothesize that converted vocals retain source-singer characteristics in these representations, placing them closer to the source than to the intended target.

We test this on positive real–synthetic pairs with known source and target singers. For each converted vocal, we compute cosine similarity to real references from both singers and report mean similarities, their difference (target minus source), and the proportion of trials where target similarity is higher. We compare pretrained and TIMBRETRACE post-trained encoders on the same trials (Figure 2).

In human validation (Section A.1), 96% of evaluated AI-generated vocals are perceived as more similar to the target than to the source. Pretrained representations, however, frequently favor the source. After TIMBRETRACE post-training, relative similarity shifts toward the target, substantially reducing this **human–model singer-identity mismatch** and helping explain conventional models' failures on AI-generated vocals.

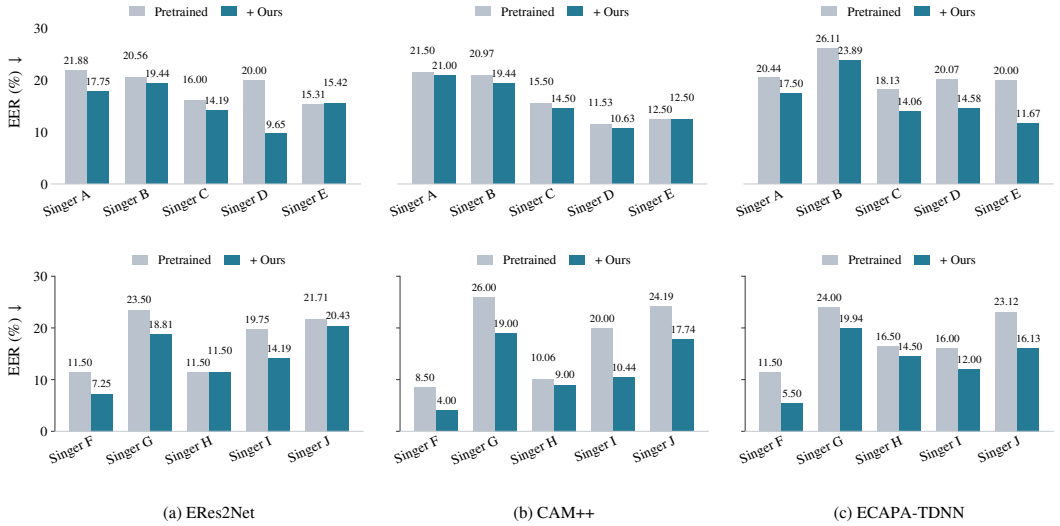
Zero-day AI Song Testing. Table 2 reports results for the Chinese and English subsets and their pooled evaluation. All pairs are formed within a language group. For the pooled results, we combine scores and labels from 14,080 Chinese and 15,776 English pairs and compute metrics from one ROC curve using a common threshold. Thus, pooled metrics are not subset averages, and no cross-language pairs are introduced.

Post-training improves aggregate EER and AUC across both languages and all three backbones, although per-singer gains vary, especially for TPR@1%FAR. Pooled EER falls from 18.70% to 16.65% for ERes2Net, 18.84% to 15.49% for CAM++, and 20.85% to 16.61% for ECAPA-TDNN. Their TPR@1%FAR rises from 26.58% to 33.82%, 37.08% to 39.55%, and 28.67% to 39.01%, respectively. Post-trained CAM++ achieves the best pooled EER (15.49%), AUC (0.9215), and TPR@1%FAR (39.55%).

The gains differ by language: CAM++ EER falls from 16.70% to 16.14% on Chinese pairs and from 18.76% to 13.49% on English pairs. ERes2Net and ECAPA-TDNN also improve EER, AUC,

Model	Our benchmark (RR)		Our benchmark (RS)		OpenSinger (RR)		M4Singer (RR)		Overall (pooled)	
	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑
ERes2Net	17.60	0.8629	45.28	0.5563	5.91	0.9755	8.03	0.9833	21.76	0.8744
+ TIMBRETRACE	11.21	**0.9653**	**9.08**	**0.9603**	4.55	**0.9908**	**3.70**	**0.9958**	**7.01**	**0.9764**
+ AAM-Softmax	11.50	0.9502	15.26	0.9129	**4.49**	0.9888	5.41	0.9916	8.97	0.9662
CAM++	17.74	0.8647	45.21	0.5333	8.80	0.9652	9.11	0.9729	22.57	0.8515
+ TIMBRETRACE	12.49	0.9317	14.55	0.8974	6.95	0.9704	7.75	0.9808	10.48	0.9437
+ AAM-Softmax	**11.14**	0.9478	19.30	0.8619	6.40	0.9757	7.21	0.9824	11.10	0.9484
ECAPA-TDNN	17.18	0.8999	42.80	0.5828	11.38	0.9484	9.65	0.9661	20.64	0.8714
+ TIMBRETRACE	**11.14**	0.9597	10.01	0.9600	6.77	0.9740	4.33	0.9922	7.84	0.9704
+ AAM-Softmax	11.92	0.9550	15.19	0.9056	6.83	0.9740	5.86	0.9840	10.45	0.9554
CONT	23.35	0.8342	48.55	0.4792	17.42	0.9029	19.21	0.8971	30.37	0.7717
CONT-VC	24.13	0.8191	49.11	0.4859	20.37	0.8717	19.21	0.8973	31.27	0.7483
BYOL	24.77	0.8399	52.16	0.4802	23.08	0.8521	20.56	0.8884	32.24	0.7373
WavLM+ECAPA	22.36	0.8128	59.69	0.3569	16.62	0.9025	8.12	0.9736	27.85	0.7781
UNIF	29.38	0.7636	54.51	0.4350	23.57	0.8527	20.29	0.8889	33.79	0.7349
VICReg	27.54	0.7952	55.15	0.4241	29.42	0.7921	22.45	0.8818	34.17	0.7005
GE2E	24.13	0.8126	48.76	0.5064	27.94	0.7991	17.76	0.9031	31.88	0.7516

Table 3: Singer verification performance on our test set, reported for its four constituent subsets and overall. "+TIMBRETRACE" denotes the corresponding pretrained model after the proposed post-training. EER (%) is lower-is-better; AUC is higher-is-better. Best results are bold; shaded rows indicate the selected models and their post-training results.



(a) ERes2Net (b) CAM++ (c) ECAPA-TDNN

Figure 3: EER before and after post-training for ten unseen singers: five Chinese-language singers (top) and five English-language singers (bottom). EER and its corresponding threshold are computed separately for each reference singer.

and TPR@1%FAR in both groups. These results support transfer to unseen singers and externally sourced AI-generated songs in both evaluated languages.

DeepFake detection. DeepFake detection provides the complementary AI-generation signal in our overall misuse-detection setting. We evaluate existing DeepFake detectors on both our benchmark and Zero-day AI Songs, with detailed results reported in Appendix A.3. The detectors perform strongly on our benchmark but degrade substantially on Zero-day AI Songs, indicating that AI-generation detection also faces generalization challenges in realistic unseen settings (Table 7).

4.3 ABLATION STUDY

We conduct ablation studies to analyze the contributions of training-data construction.

Effect of real–synthetic supervision. Our key finding is that adaptation to real singing alone is insufficient to bridge the real–synthetic identity gap. We therefore reorganize the training triplets by supervision type rather than dataset source. We examine the contributions of three post-training settings: **Real-only** uses all-real triplets from our benchmark, OpenSinger, and M4Singer; real-synthetic (**RS-only**) uses benchmark triplets containing synthetic vocals; and Full training data (**Full**) combines both sources of supervision. All settings use the same pretrained initialization and matched optimization budget. We evaluate all three settings on the same test set as in Table 3.

Training data	EER ↓	AUC ↑	TPR ↑
ERes2Net	21.76	0.8744	51.77
+ RS-only	13.94	0.9311	56.23
+ Real-only	20.26	0.8988	52.45
+ Full	**7.01**	**0.9764**	**85.61**
CAM++	22.57	0.8515	48.97
+ RS-only	16.61	0.9097	48.76
+ Real-only	21.29	0.8808	45.03
+ Full	**10.48**	**0.9437**	**65.96**
ECAPA-TDNN	20.64	0.8714	38.78
+ RS-only	13.85	0.9258	49.35
+ Real-only	16.64	0.9101	49.50
+ Full	**7.84**	**0.9704**	**82.40**

Table 4: Effect of RS-only and Real-only triplets. TPR@1%FAR (TPR) and EER are reported in %; best results per model are bold.

As shown in Table 4, RS-only achieves lower EER and higher AUC than Real-only across all three models, with higher or comparable TPR@1%FAR, demonstrating the value of our training data for the benchmark task. For example, ERes2Net achieves an EER of 13.94% with RS-only, compared with 20.26% with Real-only. Nevertheless, public real-singing data provides additional benefits when combined with our benchmark. Relative to RS-only, full training data further reduces EER from 13.94% to 7.01% for ERes2Net, from 16.61% to 10.48% for CAM++, and from 13.85% to 7.84% for ECAPA-TDNN. AUC also improves for every model, while TPR@1%FAR increases by 29.38, 17.20, and 33.05 percentage points, respectively. Full training data achieves the best results across all metrics and models, supporting the complementary value of our benchmark and public real–real supervision on the benchmark.

Effect of song-content alignment. To examine whether aligning song content in negative pairs improves learning, we additionally construct pairs in which different singer timbres are applied to the same song, denoted as $(x^r_{s,i}, x^{ai}_{t,i})$. We compare two matched training settings: *reference-aligned* negatives share the reference's source recording and temporal window, whereas *unaligned* negatives are randomly sampled. All other training conditions, including references, synthetic positives, negative target singers, conversion checkpoints, and training-set size, are held fixed. Because both settings use a matched subset of the training data, their results are not directly comparable to Table 3.

Setting	EER ↓	AUC ↑	TPR ↑
ERes2Net			
Reference-aligned	12.10	0.9458	68.10
Unaligned	**10.54**	**0.9501**	**74.80**
CAM++			
Reference-aligned	13.34	0.9336	64.69
Unaligned	**12.74**	**0.9351**	**66.11**
ECAPA-TDNN			
Reference-aligned	11.46	**0.9508**	74.88
Unaligned	**11.39**	0.9436	**77.64**

Table 5: Effect of song-content alignment in synthetic negative sampling. EER and TPR@1%FAR (TPR) are in %; best results per model are bold.

Table 5 shows that song-content alignment has little effect on performance. Unaligned sampling performs slightly better for ERes2Net and CAM++, while the two strategies yield comparable results for ECAPA-TDNN. This result is somewhat counter-intuitive: although strictly matching the musical content of negative pairs appears to provide a cleaner control for learning singer timbre, it does not improve verification under our evaluated setting. Thus, controlling negative-pair content is not necessary for effective real–synthetic singer adaptation, and we use unconstrained negative sampling in our main training setup.

5 Conclusion

We study singer-timbre misuse from a perspective beyond DeepFake detection: *who is AI singing as?* We formulate this problem as **real–synthetic singer verification** and reveal a substantial identity gap in pretrained verification models, whose representations can favor the source singer even when converted vocals are perceived as the intended target. To reduce this gap, we developed TimbreTrace, which introduces explicit real–synthetic singer supervision into pretrained verification models. Our experiments show that this supervision provides substantial gains beyond adaptation to real singing alone, while Zero-day evaluation shows that these gains transfer to unseen singers and externally sourced AI songs.

AI USE STATEMENT

We used generative AI tools in two ways. First, we used AI-based voice-generation and voice-conversion models to construct the synthetic singing data required for training and evaluation. Second, we used generative AI tools to assist with language editing, including improving the grammar, clarity, conciseness, and presentation of the manuscript. The research problem, methodology, experimental design, implementation, analysis, and scientific conclusions were developed and verified by the authors. All AI-generated data and AI-assisted text were reviewed by the authors for consistency with the intended experimental design and technical content. We take full responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the detection of AI-generated vocals that imitate the timbre of real singers. Our synthetic data are generated solely for research on singer-timbre verification. We recognize that voice-generation technologies can be used for both legitimate creative purposes and unauthorized impersonation, and our method does not determine whether a particular use is authorized or constitutes legal infringement. Instead, TIMBRETRACE provides acoustic evidence of target-singer likeness that may support broader moderation or review processes. Our human evaluation is limited to perceptual judgments of singer-identity similarity and does not collect sensitive personal information from participants.

REPRODUCIBILITY STATEMENT

Section 4.1, Section A.2 and our anonymous supplement provide singer identifiers and song identifiers; conversion-system names, versions, checkpoint provenance, training-data provenance, and redistribution terms; vocal-separator version and configuration; source-, singer-, checkpoint-, and generator-level split manifests. Our code, metadata and configurations can be viewed at: `https://anonymous.4open.science/r/TimbrewTrace-72C8`. Audio and checkpoints will be redistributed only where their original licenses permit.

REFERENCES

Roser Batlle-Roca, Wei-Hsiang Liao, Xavier Serra, Yuki Mitsufuji, and Emilia Gómez. Towards assessing data replication in music generation with music similarity metrics on raw audio. *arXiv preprint arXiv:2407.14364*, 2024.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

Yafeng Chen, Siqi Zheng, Hui Wang, Luyao Cheng, Qian Chen, and Jiajun Qi. An enhanced res2net with local and global feature fusion for speaker verification. *arXiv preprint arXiv:2305.12838*, 2023.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019.

Dorian Desblancs, Gabriel Meseguer-Brocal, Romain Hennequin, and Manuel Moussallam. From real to cloned singer identification. *arXiv preprint arXiv:2407.08647*, 2024.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

Anmol Guragain, Tianchi Liu, Zihan Pan, Hardik B Sailor, and Qiongqiong Wang. Speech foundation model ensembles for the controlled singing voice deepfake detection (ctrsvdd) challenge 2024. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 774–781. IEEE, 2024.

Rongjie Huang, Feiyang Chen, Yi Ren, Jinglin Liu, Chenye Cui, and Zhou Zhao. Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3945–3954, 2021.

Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas Evans. Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 6367–6371. IEEE, 2022.

Saebyul Park, Halla Kim, Jiye Jung, Juyong Park, Jeounghoon Kim, and Juhan Nam. Quantitative analysis of melodic similarity in music copyright infringement cases. In *ISMIR*, pp. 799–806, 2024.

Awsaf Rahman, Zaber Ibn Abdul Hakim, Najibul Haque Sarker, Bishmoy Paul, and Anowarul Fattah. Sonics: Synthetic or not - identifying counterfeit songs. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 21421–21451, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/36e2967f87c3362e37cf988781a887ad-Paper-Conference.pdf.

Lise Regnier and Geoffroy Peeters. Singer verification: singer model. vs. song model. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 437–440. IEEE, 2012.

Md Sahidullah, Tomi Kinnunen, and Cemal Hanilçi. A comparison of features for synthetic speech detection. 2015.

Davide Salvi, Hendrik Vincent Koops, and Elio Quinton. Not all deepfakes are created equal: Triaging audio forgeries for robust deepfake singer identification. *arXiv preprint arXiv:2510.17474*, 2025.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5329–5333. IEEE, 2018.

Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher. End-to-end anti-spoofing with rawnet2. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6369–6373. IEEE, 2021.

Bernardo Torres, Stefan Lattner, and Gael Richard. Singer identity representation learning using self-supervised techniques. *arXiv preprint arXiv:2401.05064*, 2024.

Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4879–4883. IEEE, 2018.

Hui Wang, Siqi Zheng, Yafeng Chen, Luyao Cheng, and Qian Chen. Cam++: A fast and efficient network for speaker verification using context-aware masking. *arXiv preprint arXiv:2303.00332*, 2023.

Yuankun Xie, Jingjing Zhou, Xiaolin Lu, Zhenghao Jiang, Yuxin Yang, Haonan Cheng, and Long Ye. Fsd: An initial chinese dataset for fake song detection. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4605–4609. IEEE, 2024.

Yuankun Xie, Ruibo Fu, Xiaopeng Wang, Zhiyong Wang, Songjun Cao, Long Ma, Haonan Cheng, and Long Ye. Detect all-type deepfake audio: Wavelet prompt tuning for enhanced auditory perception. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35922–35930, 2026.

Zongyu Yin, Federico Reuben, Susan Stepney, and Tom Collins. Measuring when a music generation algorithm copies too much: The originality report, cardinality score, and symbolic fingerprinting by geometric hashing. *SN Computer Science*, 3(5):340, 2022.

Yongyi Zang, Jiatong Shi, You Zhang, Ryuichi Yamamoto, Jionghao Han, Yuxun Tang, Shengyuan Xu, Wenxiao Zhao, Jing Guo, Tomoki Toda, et al. Ctrsvdd: A benchmark dataset and baseline analysis for controlled singing voice deepfake detection. *arXiv preprint arXiv:2406.02438*, 2024a.

Yongyi Zang, You Zhang, Mojtaba Heydari, and Zhiyao Duan. Singfake: Singing voice deepfake detection. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12156–12160. IEEE, 2024b.

Lichao Zhang, Ruiqi Li, Shoutong Wang, Liqun Deng, Jinglin Liu, Yi Ren, Jinzheng He, Rongjie Huang, Jieming Zhu, Xiao Chen, and Zhou Zhao. M4singer: A multi-style, multi-singer and musical score provided mandarin singing corpus. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 6914–6926. Curran Associates, Inc., 2022. doi: 10.52202/068431-0501. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/2de60892dd329683ec21877a4e7c3091-Paper-Datasets_and_Benchmarks.pdf.

You Zhang, Yongyi Zang, Jiatong Shi, Ryuichi Yamamoto, Jionghao Han, Yuxun Tang, Tomoki Toda, and Zhiyao Duan. Svdd challenge 2024: A singing voice deepfake detection challenge evaluation plan. *arXiv preprint arXiv:2405.05244*, 2024.

A Appendix

A.1 Human Validation of Singer Timbre

For synthetic songs, the target label denotes the intended singer identity specified by the conversion checkpoint. Because checkpoint identity does not guarantee perceptual identity preservation, we conduct a human validation study on 50 randomly selected generated songs. Three expert listeners (one co-author, two musical experts) independently identify the singer whose timbre each generated vocal most closely resembles (label 0 for the source singer, label 1 for the target singer). We assign a perceived singer label when at least two listeners agree. For 48 of the 50 songs, the resulting perceived label matches the intended target identity. The detailed sampled songs as well as the total ratings per anonymous expert are open-sourced in our supplement. This evaluation provides a human validation of timbre similarity for the sampled songs.

A.2 Benchmark Construction Details

Conversion checkpoints and provenance. We construct our benchmark using two publicly released, singer-specific singing voice conversion (SVC) models: so-vits-svc 4.0[2] and DDSP-SVC[3]. Specifically, we collect 22 singer-specific checkpoints along with their corresponding configuration files. For each song, we use the checkpoints to generate converted singing voices, so our benchmark covers outputs from two distinct architectures. This setup allows us to evaluate singer recognition and verification across different model families and their generated voice characteristics. Their training-data provenance is incomplete; we therefore cannot establish whether their training recordings overlap with our collected recordings.

Vocal preprocessing. We process recordings as 44.1 kHz stereo audio using `audio-separator` version 0.47.0.[4] The pipeline removes instrumental accompaniment, separates lead from backing vocals, and reduces reverberation. The resulting dry lead vocals serve as the real recordings and as inputs to voice conversion. The separator operates in FP32 with a normalization threshold of 0.99 and an amplification threshold of 0.0.

Segment selection. We divide each real dry-vocal recording into non-overlapping 6-second windows, discarding incomplete trailing windows. We estimate voiced activity using RMVPE on a mono 16 kHz version of the source vocal, with an inference threshold of 0.03. A frame is considered voiced when its estimated fundamental frequency is positive. At a 10 ms frame interval,

[2] https://github.com/svc-develop-team/so-vits-svc

[3] https://github.com/yxlllc/DDSP-SVC

[4] https://github.com/nomadkaraoke/python-audio-separator

we retain every complete window containing at least 300 voiced frames, corresponding to 3 seconds of voiced content. We apply the same temporal boundaries to all converted versions of each source recording to preserve source-content alignment. Segment selection depends only on voiced activity in the real source, not on generated-audio quality or verification scores. Retained segments are stored as 6-second, 44.1 kHz mono PCM16 WAV files and resampled to the input rate required by each verification backbone. All segments and converted versions derived from the same source recording belong to the same data split.

A.3 Evaluation of Singing Deepfake Detection

We evaluate two official CtrSVDD configurations (Zang et al., 2024a; Zhang et al., 2024), using LFCC (Sahidullah et al., 2015) and raw-waveform (Tak et al., 2021) front ends with an AASIST-based back end (Jung et al., 2022). We use the released pretrained checkpoints without additional training on our data. We test both models on our test set and the external online-song collection used for Zero-day evaluation. For this experiment, each audio sample is labeled as real or AI-generated, rather than evaluated through singer-verification pairs.

As shown in Table 6, both baselines perform well on our benchmark, achieving EERs below 2.5% and TPR@1%FAR above 96%. However, their performance declines on the external collection: EER increases to 35.00% for LFCC and 22.76% for RawNet, while TPR@1%FAR falls to 7.05% and 20.68%, respectively. These results indicate a substantial transfer gap for the two evaluated checkpoints under our external evaluation conditions. They do not establish that all singing deepfake detectors exhibit the same limitation.

Deepfake detection and singer verification address complementary questions. A deepfake detector assesses whether a recording is AI-generated, whereas our verification task assesses whether its vocal identity matches a reference singer. Consequently, even accurate real-versus-generated classification does not identify whose vocal identity a generated recording resembles. We therefore evaluate these tasks separately; the present experiment does not measure the performance of a combined detection-and-verification system.

Test set	Baseline	EER ↓	AUC ↑	TPR ↑
Our Test	LFCC	2.42	0.9970	97.06
	RawNet	1.59	0.9989	96.98
Unseen AI singer	LFCC	35.00	0.7040	7.05
	RawNet	22.76	0.8449	20.68

Table 6: Real-versus-generated singing detection. EER and TPR@1%FAR (TPR) are percentages.

A.4 Joint evaluation with deepfake detection

We evaluate whether combining AI-generation detection with target-singer verification identifies generated vocals that match a reference singer. We reuse the development-selected, post-trained ERes2Net checkpoints evaluated in Table 3 and pair them with either LFCC+AASIST or RawNet+AASIST. Both deepfake detectors use their official pretrained weights without adaptation to our data. For a candidate vocal x and an real reference r, we define the joint decision as

$$\hat{y}(x, r) = \mathbb{I}[d(x) \geq \tau_{\text{AI}}]\, \mathbb{I}[v(x, r) \geq \tau_{\text{ID}}]\,,$$

where higher $d(x)$ indicates stronger evidence of AI generation and $v(x, r)$ is the cosine similarity between the candidate and reference embeddings.

We evaluate on the in-house portion of the held-out test set (Our benchmark). The evaluation contains 5,636 pairs, equally distributed across four groups defined by candidate reality (real or generated) and identity match (matching or non-matching), with 1,409 pairs per group. Only generated candidates matching the reference singer are positive examples. The remaining three groups are negatives, including real recordings of the reference singer.

We use the development split for checkpoint selection and joint-threshold calibration. For calibration, we combine the existing 2,560 RS pairs with 2,560 RR pairs constructed from the same

development split on our benchmark, yielding 1,280 pairs in each of the four groups. These pairs reuse development recordings and do not form an independent validation set. After selecting the checkpoint and fixing both thresholds, we evaluate the joint system on the source-recording-disjoint test split, which is used exclusively for final evaluation.

For the real–real branch, identity-matching pairs use distinct segments from the same source recording, whereas non-matching pairs use different singers. Reference recordings with fewer than two real segments are excluded from this branch. These additional pairs form a separate calibration manifest and do not alter the original data splits. We jointly select $(\tau_{\mathrm{AI}}, \tau_{\mathrm{ID}})$ to maximize development TPR subject to a pooled development FPR of at most 1%. Ties are broken by lower FPR, then by higher τ_{AI}, and finally by higher τ_{ID}. We fix the selected thresholds for all test evaluations.

As shown in Table 7, the RawNet+AASIST combination achieves 82.11% TPR at an observed test FPR of 0.88%, while the LFCC+AASIST combination achieves 70.40% TPR at the same FPR to two decimal places. The pooled FPR aggregates all three negative groups. For RawNet+AASIST, the corresponding group-specific FPRs are 1.92% for generated non-matching candidates, 0.71% for real matching candidates, and 0.00% for real non-matching candidates. Precision is computed under the 25% positive prevalence of this evaluation protocol.

These results support the feasibility of combining AI-generation detection and target-singer verification on the benchmark. They do not establish joint-system generalization to unseen online recordings or determine whether an identified imitation is authorized.

Deepfake detector	TPR ↑	FPR ↓	Precision ↑
LFCC+AASIST	70.40	0.88	96.40
RawNet+AASIST	**82.11**	0.88	**96.90**

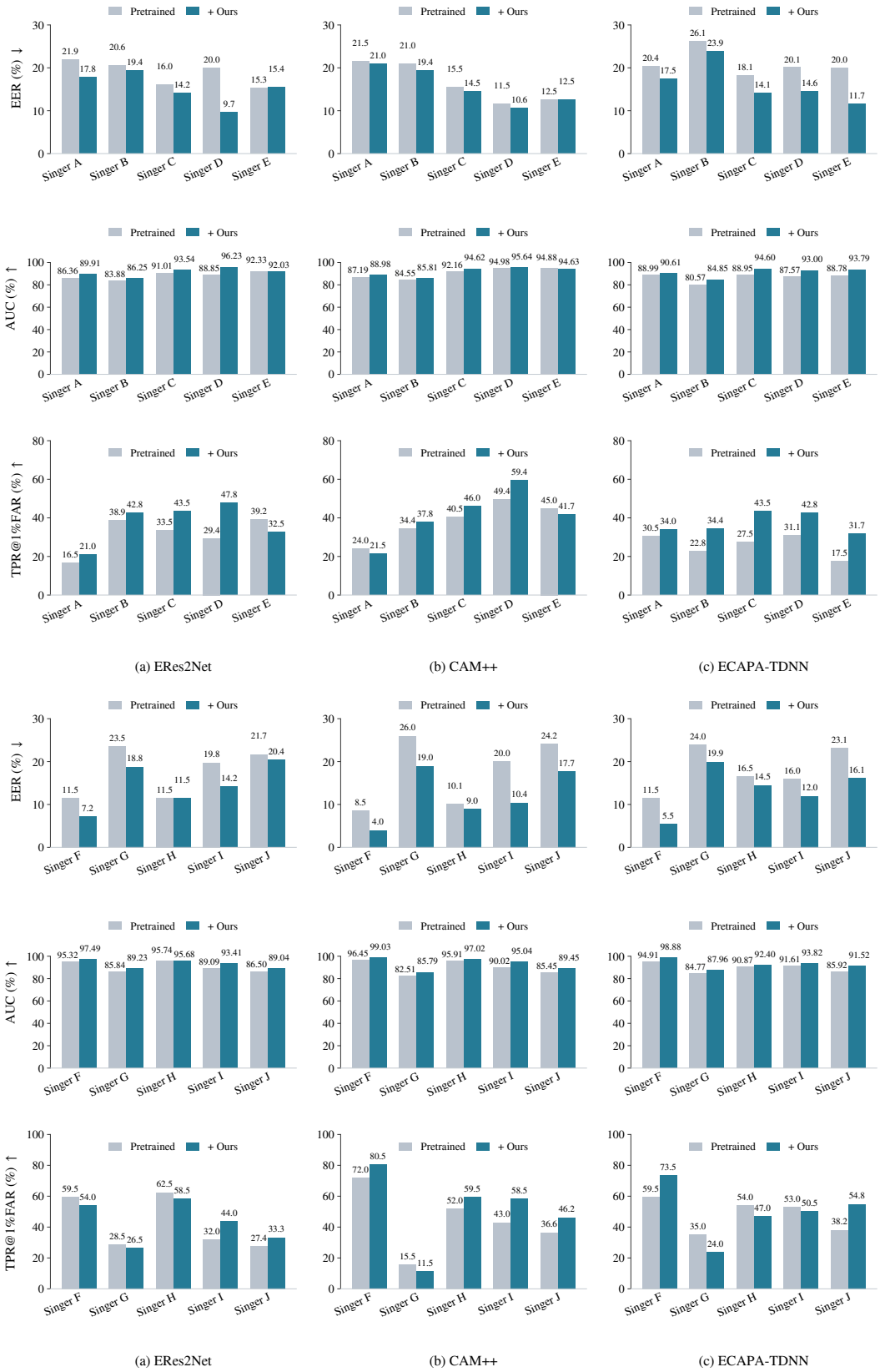
Table 7: Joint evaluation on our benchmark using an AND rule with post-trained ERes2Net. Thresholds are calibrated on development data (FPR $\leq 1\%$) and fixed for testing. Values are in %; positive prevalence is 25%.

A.5 Detailed Results on Zero-day AI Songs

Figure 4 complements the EER results in the main text by reporting all three verification metrics before and after post-training for each unseen singer and backbone. Singer identifiers A–E denote the Chinese-language singers, and F–J denote the English-language singers, consistent with the main text. Metrics and thresholds are computed separately within each reference-singer subset.

Across the 30 singer–model combinations, post-training reduces EER in 27, increases AUC in 27, and improves TPR@1%FAR in 20. ERes2Net achieves its largest EER reduction on Singer D, from 20.00% to 9.65%, accompanied by an increase in TPR@1%FAR from 29.44% to 47.78%. CAM++ reduces EER from 20.00% to 10.44% on Singer I, while ECAPA-TDNN reduces EER from 20.00% to 11.67% on Singer E. ECAPA-TDNN improves EER and AUC for all ten singers, although its TPR@1%FAR gains are not uniform.

The detailed results reveal variation across identities, backbones, and operating points. For Singer A, CAM++ improves EER and AUC, but TPR@1%FAR decreases from 24.00% to 21.50%. For Singer E, ERes2Net slightly worsens EER and AUC, while CAM++ leaves EER unchanged and slightly lowers AUC; both models exhibit lower TPR@1%FAR. Among the English-language singers, ECAPA-TDNN improves EER and AUC for Singers G–I but yields lower TPR@1%FAR. These results demonstrate improved verification for most evaluated singer–model combinations, while showing that better overall discrimination does not necessarily translate into improved performance at a strict false-acceptance operating point.



(a) ERes2Net (b) CAM++ (c) ECAPA-TDNN

(a) ERes2Net (b) CAM++ (c) ECAPA-TDNN

Figure 4: Verification performance before and after post-training for five Chinese-language singers (top) and five English-language singers (bottom). Metrics and thresholds are computed separately for each reference singer.