SRAM: LEARNING SPECTRAL RELATION-AWARE REPRESENTATIONS FOR ENVIRONMENTAL SOUND DEEPFAKE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Environmental sound deepfake detection (ESDD) models are critical for countering fabricated environmental audio, which can mislead users of audio-based assistive technologies and trigger unintended smart-home responses. A reliable ESDD method requires accurate detection even when the source datasets and generators are unseen in training. Existing ESDD methods broadly follow two paradigms: task-specific discriminative models and generative methods based on audio large language models (ALLMs). We find that both approaches have significant limitations: (i) discriminative methods are highly dependent on the seen domain conditions of the training data, and their classification performance drops significantly when the test audio comes from unseen domains; (ii) generative methods generally underperform and incur higher per-sample inference latency and deployment costs. We conduct a large-scale evaluation of 15 ALLMs, spanning five model families, on the ESDD task. The results show that their zero-shot performance is close to random guessing, and unlike the gains reported in speech deepfake detection, their performance remains significantly inferior to that of strong task-specific discriminative models even after fine-tuning. Further analysis shows that a classifier built on an ALLM's audio encoder can outperform the full generative model, while local spectro-temporal residuals and time-frequency geometry reveal differences between real and AI-generated audio. Motivated by these findings, we propose a novel detection framework, called Spectral-Relational Artifact Modeling (SRAM). SRAM combines audio representation learning with spectral relation modeling. It builds on the Qwen3-Omni audio encoder, using cross-layer trajectory refinement to integrate information from multiple encoder depths and sparse evidence pooling to focus on informative temporal tokens. To complement these audio representations, we use a spectrogram-residual branch to capture local spectral differences and a structure-tensor branch to describe local time-frequency geometry. Extensive experiments demonstrate SRAM's strong detection performance, generalization to unseen domains, and robustness to unseen perturbations.

1 INTRODUCTION

With the rapid advancement of audio generative models such as STAR-VAE (Liu et al., 2026), BEAT (Qian et al., 2026), AudioLDM (Liu et al., 2023), AudioLCM (Liu et al., 2024) and HiggsAudio (Boson AI, 2025), synthesized audio has become highly realistic, making it difficult even for normal-hearing individuals to distinguish genuine sounds from fabricated ones. For users of audio-based assistive technologies, especially blind or visually impaired people, ambient sounds are essential for understanding their surroundings. Fake environmental sounds can mislead these users and create serious safety risks. In increasingly prevalent smart home environments, forged environmental sounds may also trigger unintended system responses, causing security holes or financial loss. These risks highlight the need for reliable detection of AI-generated environmental sounds.

Environmental sound deepfake detection (ESDD) aims to distinguish real environmental sound from AI-generated audio. Yin et al. (2025) highlight that environmental sounds differ acoustically from speech and singing and may contain overlapping events, such as alarms, footsteps, and sirens, potentially limiting the direct applicability of speech and singing deepfake detectors.

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.

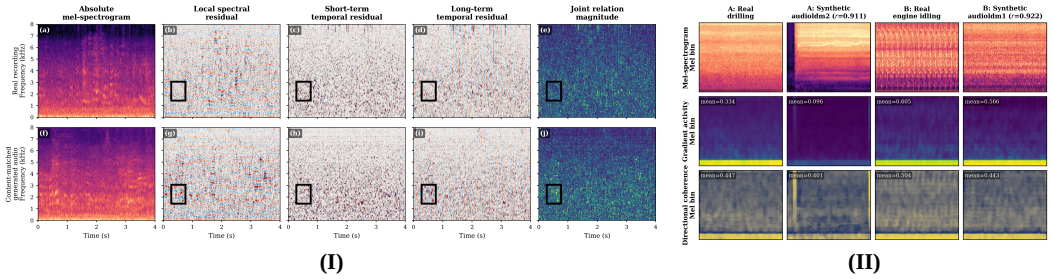


Figure 1: Local spectral artifacts in real–generated pairs. (I) Spectral and temporal residuals highlight local differences (boxes) between a real audio clip (top) and its generated counterpart (bottom); the last column shows their combined magnitude. (II) Structure-tensor descriptors reveal lower gradient activity (A) or lower directional coherence (B) in generated audio. Here, r denotes time-averaged band-energy correlation, and inset values are map means. Color scales are shared within each column in (I) and each real–generated pair in (II).

Existing audio deepfake detection methods can be broadly categorized into discriminative and generative approaches, but both remain limited in their performance on ESDD. The generative method builds detection on audio large language models (ALLMs) (Gu et al., 2025; Xie et al., 2026): prompted with an audio clip, the model answers in text, and the decision is read from the generated answer or from the logits of the answer tokens (Wang et al., 2025). The promise is a single generalist model, trained on far broader audio, that replaces a task-specific detector. Existing studies (Gu et al., 2025; Xie et al., 2026) mainly focus on Qwen-family models (Chu et al., 2023; 2024) for speech or general audio deepfake detection, leaving ALLMs largely underexplored for ESDD. Our experiments show that generative models underperform strong task-specific discriminative detectors in classification. Discriminative detectors (Fan et al., 2026; Tak et al., 2022; Yin et al., 2025) are relatively compact and accurate, but their generalization remains limited. Even the strongest of them is near-perfect when the source datasets are seen in training, but degrades sharply when they are unseen. This limitation may stem from detectors relying on recording conditions specific to the training data, while failing to learn sufficiently generalizable synthesis cues across domains.

To address these problems, we first conduct a large-scale evaluation of 15 ALLMs from five model families (Chu et al., 2023; 2024; Team, 2025b; Zhang et al., 2024; Team, 2025a; Abouelenin et al., 2025; Team, 2025c) on EnvSDD (Yin et al., 2025), whose test sets cover all combinations of seen and unseen source datasets and seen and unseen generators. Compared with prior studies (Gu et al., 2025; Xie et al., 2026) that focused mainly on the Qwen family, our evaluation covers a broader range of ALLMs for ESDD. However, ALLMs show clear limitations in detection: zero-shot performance is near random, most fine-tuned models still underperform a task-specific discriminative detector, and their large size increases deployment costs. As shown in Figure 9, using only the ALLM audio encoder with a simple MLP classifier performs comparably to or better than the full model. However, its detection performance and generalization to unseen domains still leave room for improvement. We explore whether local spectral relations can reveal additional cues associated with synthesis artifacts. Figure 1(I) illustrates how spectral and temporal residuals highlight local differences between a real audio clip and its generated counterpart. Figure 1(II) further shows that real and AI-generated samples with similar band-energy profiles can still differ in local gradient activity and directional coherence, providing useful cues for distinguishing them. Additional qualitative examples and descriptor definitions are provided in Appendices A.6.1 and A.6.2. These observations suggest that differences between real and AI-generated audio are reflected in local spectro-temporal residuals and local time-frequency geometry. Building on these findings, we propose Spectral-Relational Artifact Modeling (SRAM), which combines audio representation learning with spectral relation modeling. For audio representation learning, we build on the Qwen3-Omni audio encoder (Team, 2025b), use cross-layer trajectory refinement to integrate information from multiple encoder depths (Pasad et al., 2021), and apply sparse evidence pooling to focus on informative temporal tokens. For spectral relation modeling, spectrogram-residual and structure-tensor branches capture local differences and time-frequency geometry, providing complementary artifact cues for detection. We conduct extensive experiments, and the results show that SRAM achieves strong performance in both generalization and robustness.

All in all, our contributions are as follows:

- We systematically benchmark ALLMs for ESDD and find that task-specific discriminative detectors outperform zero-shot and fine-tuned ALLMs in most settings, showing that gains reported for ALLMs in speech deepfake detection do not consistently transfer to ESDD.
- We propose SRAM, which combines audio representation learning with spectral relation modeling. Cross-layer trajectory refinement and sparse evidence pooling help identify informative audio cues, complemented by spectrogram residuals and structure-tensor features that capture local spectral relations.
- Extensive experiments on EnvSDD show that SRAM overall outperforms existing methods in terms of performance, generalization, and robustness.

2 Related Work

2.1 Audio Deepfake Detection Methods

Audio deepfake detection research mainly focuses on speech, using synthesis artifacts and speech-specific cues. AASIST (Jung et al., 2022) jointly models spectral and temporal artifacts through heterogeneous graph attention. Some approaches investigate spectrogram modeling with vision Transformers (Modak et al., 2025), spectro-temporal cross-attention with multi-scale convolution (Hao et al., 2025), and multi-scale waveform–spectrogram fusion (Jin et al., 2025). Another line of work focuses on specific components of speech: BTS-E (Doan et al., 2023) models relationships between breathing, talking, and silence, while Sivaraman et al. (2025) train detectors separately on voiced and unvoiced regions and fuse their scores. Beyond feature modeling, researchers have explored pretrained wav2vec 2.0 representations with data augmentation (Tak et al., 2022) and GAN-based augmentation with contrastive learning (Kashyap et al., 2025). To adapt to emerging attacks while mitigating forgetting, RAIS (Febrinanto et al., 2025) uses auxiliary labels to guide replay sample selection, while RegO (Chen et al., 2025b) adjusts parameter updates according to neuron importance. CD-DAT (Chen et al., 2025a) combines domain adversarial training with causality distillation for continual adaptation to unlabeled target domains. Parameter efficiency is also considered in QT-CNN (Lin et al., 2025), which uses quantum-to-classical parameter mapping to reduce trainable parameters. Beyond speech, several studies use general-audio representations for ESDD. Ouajdi et al. (2024) use CLAP embeddings to detect synthesized sounds. Yin et al. (2025) combine BEATs (Chen et al., 2023) with AASIST, while Xie et al. (2024) use frozen EAT features for detection and localization. DSSL (Fan et al., 2026) further combines general-audio and speech representations for unified detection across speech, environmental sounds, singing, and music. Audio large language models (ALLMs) offer another approach by formulating detection as audio question answering. ALLM4ADD (Gu et al., 2025) uses Qwen-Audio-Chat (Chu et al., 2023) as its default backbone and applies supervised fine-tuning (SFT) for authenticity detection. It reports improvements over task-specific discriminative baselines on ASVspoof2019 LA. Xie et al. (2026) combine frequency-time reasoning demonstrations with supervised and reinforcement fine-tuning to produce decisions and explanations across multiple audio types. Our work evaluates multiple ALLM families for ESDD and introduces SRAM, which combines fine-tuned audio representations with local spectral differences and time-frequency structure.

2.2 Environmental Sound Deepfake Detection Datasets

Existing datasets address different forms of environmental audio forgery, from acoustic scene manipulation to sound synthesis. SceneFake (Yi et al., 2024) manipulates the background acoustic scene of speech recordings, while EnvFake (Cheng et al., 2024) changes acoustic scenes without altering linguistic or visual content to study audio-visual scene consistency. For synthesized environmental sounds, Ouajdi et al. (2024) evaluate audio generated by 44 systems from the DCASE Foley sound synthesis challenge (Choi et al., 2023), focusing on monophonic sounds with in-domain evaluation. FakeSound (Xie et al., 2024) addresses partially manipulated audio by masking and regenerating sound-event segments in AudioCaps audio clips (Kim et al., 2019), supporting both detection and localization. EnvSDD (Yin et al., 2025) broadens coverage to monophonic and polyphonic sounds from six source datasets, using five text-to-audio models and two of these models for audio-to-audio generation. Its evaluation covers all combinations of seen and unseen source datasets and generators

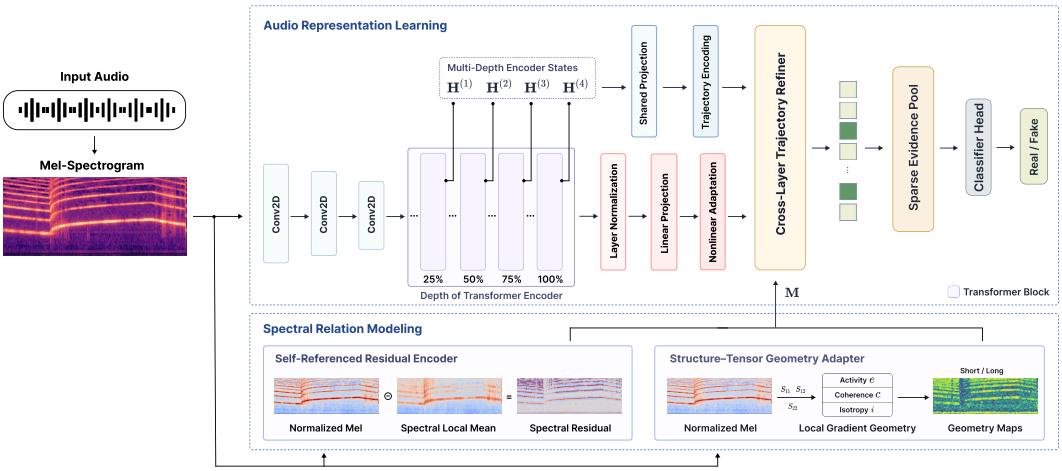


Figure 2: Overview of SRAM, combining audio representation learning and spectral relation modeling. The self-referenced residual encoder and structure-tensor geometry adapter extract local spectro-temporal residuals and gradient geometry from the normalized mel-spectrogram. These spectral artifact cues and a trajectory code from multiple encoder depths guide iterative token refinement, followed by sparse evidence pooling and a linear classifier head for real/fake prediction.

for both generation paradigms. More recently, SynSFX (Li et al., 2026) introduces sound-effect deepfakes from seven text-to-audio models, with a shared-prompt subset for cross-generator analysis. Its evaluation mainly examines generalization to unseen generators.

3 Method

3.1 Overview

Given an environmental sound clip, let $\mathbf{X} \in \mathbb{R}^{T \times F}$ denote its mel-spectrogram with T frames and F bins. Our goal is to predict $y \in \{0, 1\}$, where 0 and 1 denote real and synthesized, respectively. As shown in Figure 2, SRAM combines audio representation learning with spectral relation modeling. The audio encoder produces temporal tokens, while two spectral branches extract local spectro-temporal residuals and time-frequency geometry. We add these spectral artifact cues to the tokens through a learned gate, then refine them using both these spectral artifact cues and changes in encoder states across layers. Finally, sparse evidence pooling combines selected local artifact cues with global token statistics to form a clip representation for real/fake classification.

3.2 Audio Representation Learning

We first describe how SRAM encodes an audio clip, refines its temporal tokens with spectral artifact cues, and pools them for classification. These cues are extracted by the two branches described in Section 3.3.

Audio encoder. We use the audio encoder of Qwen3-Omni (Team, 2025b) and discard all other components. We initialize the encoder with its pretrained weights and fine-tune all its parameters jointly with the rest of SRAM. The input $\mathbf{X}$ is encoded into N tokens, $\mathbf{H} \in \mathbb{R}^{N \times d}$, where d is the encoder feature dimension. These tokens are then projected to D dimensions, yielding $\mathbf{Z} \in \mathbb{R}^{N \times D}$. Let $\mathbf{M} \in \mathbb{R}^{N \times D}$ denote the spectral artifact features aligned with these tokens. We incorporate these features using a cross-gate $\mathbf{A}_{\text{cross}} = \sigma(\text{FFN}([\mathbf{Z}; \mathbf{M}]))$ to condition the encoder tokens: $\tilde{\mathbf{Z}} = \mathbf{Z} + \mathbf{A}_{\text{cross}} \odot \mathbf{M}$. The gate controls how much of these features is added to each token.

Cross-layer trajectory refinement. Audio encoder layers capture different information, and the final layer may not contain all the cues useful for detection (Pasad et al., 2021). We therefore use four equally spaced encoder layers and treat their states $\{\mathbf{H}^{(1)}, \ldots, \mathbf{H}^{(4)}\}$ as samples along one trajectory. To reduce sensitivity to activation scale across layers, we apply a parameter-free layer norm followed by a shared projection. We then describe how the projected states change across

layers:

$$\mathbf{v}^{(k)} = \frac{\mathbf{P}^{(k+1)} - \mathbf{P}^{(k)}}{\|\mathbf{P}^{(k+1)} - \mathbf{P}^{(k)}\|}, \qquad \boldsymbol{\kappa}^{(k)} = \mathbf{v}^{(k+1)} - \mathbf{v}^{(k)}, \tag{1}$$

where $\mathbf{P}^{(k)}$ is the projected k-th state. We concatenate the mean and standard deviation of the projected states, the mean direction $\mathbf{v}^{(k)}$, and the mean absolute curvature $\boldsymbol{\kappa}^{(k)}$, then project them to a trajectory code $\mathbf{T}$. Starting from $\mathbf{Z}_0 = \tilde{\mathbf{Z}}$, a single weight-shared cell then refines the conditioned tokens for R iterations,

$$\mathbf{Z}_{r+1} = \mathbf{Z}_r + s_r \cdot \sigma(\mathbf{G}([\mathbf{Z}_r; \mathbf{T}; \mathbf{M}])) \odot \mathbf{U}([\mathbf{Z}_r; \mathbf{T}; \mathbf{M}]), \tag{2}$$

where $\mathbf{U}$ and $\mathbf{G}$ are the shared update and gate networks and $s_r \in (0, 1)$ is a learned step size initialized small. All iterations share the same update and gate networks. Small initial step sizes limit changes to the input tokens at the start of training.

Sparse evidence pooling. We use sparse pooling to select and aggregate local cues from the refined audio tokens (Martins & Astudillo, 2016). The module first measures how each token differs from its temporal neighbors, then scores these differences and pools the selected cues.

Let $\mathbf{Q} = \mathrm{LN}(\mathbf{Z}_R)$ denote the refined tokens after layer normalization. We standardize each feature channel over time to obtain $\hat{\mathbf{Q}}$. At each temporal scale k, we subtract a local temporal mean from $\hat{\mathbf{Q}}_n$ to obtain the residual $\mathbf{r}_n^{(k)}$. We also compute a local deviation $\mathbf{d}_n$ from $\mathbf{Q}_n$, scaled by its per-channel temporal standard deviation. Both quantities measure local differences in token features. The window sizes and exact definitions are given in Appendix A.5.

To combine information across temporal scales, we score each residual using a learned function w and its energy. The resulting softmax weights determine how much each scale contributes:

$$s_n^{(k)} = w\big(\mathbf{r}_n^{(k)}\big) + \tfrac{1}{2} \log \big\|\mathbf{r}_n^{(k)}\big\|^2, \qquad \mathbf{r}_n = \sum_k \beta_n^{(k)}\, \mathbf{r}_n^{(k)}, \tag{3}$$

where $\beta_n^{(k)} = \mathrm{softmax}_k(s_n^{(k)})$ and $\|\cdot\|^2$ denotes the mean of squared entries. A second scoring network g uses the magnitudes of $\mathbf{r}_n$ and $\mathbf{d}_n$, together with their combined energy, to score each token:

$$s_n = g\big([\,|\mathbf{r}_n|;\, |\mathbf{d}_n|\,]\big) + \log\big(\|\mathbf{r}_n\|^2 + \|\mathbf{d}_n\|^2\big). \tag{4}$$

We retain the $\lceil \rho N \rceil$ highest-scoring tokens, where ρ is the retained fraction, and set the remaining scores to $-\infty$. A temperature-scaled softmax gives attention weights α over the selected tokens.

Finally, we use these weights to aggregate signed residuals, local deviations, and variation across scales. We also summarize large residual magnitudes separately in each feature channel. These four evidence statistics complement the temporal mean and standard deviation of $\mathbf{Q}$ in the clip embedding:

$$\mathbf{z} = \mathrm{LN}\big(\bar{\mathbf{Q}} + \lambda_s\, \mathbf{F}(\bar{\mathbf{Q}}, \mathrm{sd}(\mathbf{Q}), \mathbf{a}_{\mathrm{sig}}, \mathbf{a}_{\mathrm{tail}}, \mathbf{a}_{\mathrm{mel}}, \mathbf{a}_{\mathrm{dis}})\big), \tag{5}$$

where $\mathbf{F}$ is a fusion network and $\lambda_s \in (0, 1)$ is a learned weight. The four statistics summarize signed residuals, large residual magnitudes, local token deviations, and variation across scales, respectively. The detailed computations are in Appendix A.5. We use a simple linear classification head to map $\mathbf{z}$ to two class logits.

3.3 Spectral Relation Modeling

We now describe how the spectral artifact features used in Section 3.2 are constructed. Motivated by the local differences observed in Figure 1, we complement the audio encoder representations with two branches: one extracts spectro-temporal residuals, and the other captures local time-frequency geometry using structure tensors.

Spectrogram residuals. We standardize the input spectrogram $\mathbf{X}$ over time and frequency to obtain $\hat{\mathbf{X}}$. The first branch constructs five maps from $\hat{\mathbf{X}}$: a spectral residual $\hat{\mathbf{X}} - \mu_{9\times1}(\hat{\mathbf{X}})$, a short and a long temporal residual $\hat{\mathbf{X}} - \mu_{1\times5}(\hat{\mathbf{X}})$ and $\hat{\mathbf{X}} - \mu_{1\times17}(\hat{\mathbf{X}})$, a joint time-frequency transport term $\partial_t \partial_f \hat{\mathbf{X}}$, and a temporal curvature term $\partial_t^2 \hat{\mathbf{X}}$, where $\mu_{a\times b}$ is a local mean over a bins and b frames. Standardization and local differences reduce the influence of overall signal level. A small convolutional encoder reduces the five maps to $\mathbf{M}_{\mathrm{res}} \in \mathbb{R}^{N\times D}$ aligned with the encoder tokens.

Table 1: Information about the EnvSDD dataset. 'Seen SD' and 'Seen GM' denote seen source datasets and seen generation models, respectively.

Split	Abbreviation	Real	Fake	Seen SD	Seen GM
Valid_TTA	VT	7,942	23,826	✓	✓
Test_TTA_01	TT1	3,971	11,913	✓	✓
Test_TTA_02	TT2	3,971	7,942	✓	✗
Test_TTA_03	TT3	1,000	3,000	✗	✓
Test_TTA_04	TT4	1,000	2,000	✗	✗
Valid_ATA	VA	7,942	7,942	✓	✓
Test_ATA_01	AT1	3,971	3,971	✓	✓
Test_ATA_02	AT2	3,971	3,971	✓	✗
Test_ATA_03	AT3	1,000	1,000	✗	✓
Test_ATA_04	AT4	1,000	1,000	✗	✗

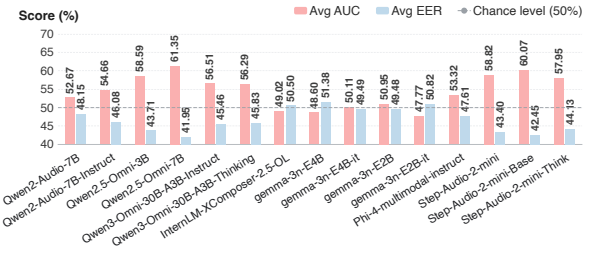


Figure 3: Average zero-shot AUC (%) and EER (%) of the 15 ALLMs over the eight EnvSDD test subsets. The dashed line marks the chance level of 50%.

Structure-tensor artifact cues. Real and AI-generated audio can have similar overall band energy but differ in the strength and direction of local spectrogram changes (Figure 1). We use structure tensors to describe these local patterns. Following Harris & Stephens (1988), we treat frequency and time as the two axes and average the gradient products over a local window Ω of the standardized spectrogram,

$$\mathbf{S} = \frac{1}{|\Omega|} \sum_{\Omega} \begin{bmatrix} (\partial_f \hat{\mathbf{X}})^2 & \partial_f \hat{\mathbf{X}}\, \partial_t \hat{\mathbf{X}} \\ \partial_f \hat{\mathbf{X}}\, \partial_t \hat{\mathbf{X}} & (\partial_t \hat{\mathbf{X}})^2 \end{bmatrix}, \tag{6}$$

and read out three rotation-invariant descriptors,

$$e = \log(1 + \operatorname{tr} \mathbf{S}), \quad c = \frac{\sqrt{(\mathbf{S}_{11} - \mathbf{S}_{22})^2 + 4\mathbf{S}_{12}^2}}{\operatorname{tr} \mathbf{S}}, \quad i = \frac{\sqrt{\det \mathbf{S}}}{\operatorname{tr} \mathbf{S}}, \tag{7}$$

where $\operatorname{tr}(\mathbf{S})$ and $\det(\mathbf{S})$ denote the trace and determinant of $\mathbf{S}$, respectively; e measures local activity, c measures gradient orientation consistency, and i measures patch isotropy. Descriptors are computed using short and long windows. We concatenate them with local gradient and curvature maps, encode the resulting maps with a small convolutional network, and use adaptive pooling and a projection to obtain $\mathbf{M}_{\text{geo}} \in \mathbb{R}^{N \times D}$ aligned with the audio tokens. Appendix A.5 provides the map definitions and encoder settings. A learnable scalar gate $\gamma = \gamma_{\max} \tanh(g)$ controls this branch, where $\gamma_{\max}$ bounds the gate magnitude. We initialize g to zero so that the branch initially contributes no features. We combine the residual and geometric artifact features as $\mathbf{M} = \mathbf{M}_{\text{res}} + \gamma \mathbf{M}_{\text{geo}}$. The resulting $\mathbf{M}$ is used in the token conditioning and refinement described above.

3.4 Training Objective

We adopt a simple classification objective to train SRAM end-to-end, using only label-smoothed cross-entropy. Given the predicted class probabilities $\mathbf{p}$ and the ground-truth label y, the loss is

$$\mathcal{L}_{\text{cls}} = -\sum_{c=0}^{1} \tilde{y}_c \log p_c, \tag{8}$$

where $\tilde{y}_c = (1 - \varepsilon)\mathbf{1}[y = c] + \varepsilon/2$ is the smoothed target with $\varepsilon = 0.05$, and $\mathbf{1}[y = c]$ equals 1 if $y = c$ and 0 otherwise. The loss is averaged over each mini-batch.

4 Experiments

4.1 Settings

Metrics. Following ALLM4ADD (Gu et al., 2025) and EnvSDD (Yin et al., 2025), we report the Area Under the Curve (AUC) and the Equal Error Rate (EER) as the main evaluation metrics. A higher AUC and a lower EER both indicate a better detector. Unless otherwise specified, model performance is evaluated using AUC (%) and EER (%).

Dataset. We follow the official EnvSDD data partitioning protocol (Yin et al., 2025). The dataset contains 45.25 hours of real audio and 316.74 hours of synthetic audio. Its test sets cover both

Table 2: Post-training evaluation of generative methods and task-specific discriminative approaches on the EnvSDD test sets, in AUC (%) and EER (%). For ALLMs, SFT applies LoRA to the LLM component; italicized audio encoders are trainable, whereas non-italicized ones are frozen. The best result in each column is shown in bold.

Model	AT1		AT2		AT3		AT4		TT1		TT2		TT3		TT4		**Avg**	
	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓
Qwen2-Audio-7B	99.82	01.96	62.97	40.64	79.24	28.80	44.81	52.90	99.81	02.12	99.56	03.23	75.20	31.40	59.65	43.90	77.63	25.62
Qwen2-Audio-7B-Instruct	99.83	01.84	61.80	41.22	81.19	26.60	43.99	53.70	99.83	02.03	99.59	02.98	76.15	30.80	61.62	42.20	78.00	25.17
Qwen2.5-Omni-3B	99.90	01.26	60.97	42.79	77.26	29.93	42.41	55.80	99.93	01.25	99.83	02.00	74.72	31.07	67.63	37.50	77.83	25.20
Qwen2.5-Omni-7B	99.97	00.63	63.40	40.81	83.38	24.54	43.14	55.10	99.95	00.93	99.82	01.99	79.07	29.01	66.91	38.85	79.45	23.98
Qwen3-Omni-30B-A3B-Instruct	**100.0**	00.15	98.62	06.02	95.53	10.60	81.85	26.00	**100.0**	00.20	99.95	01.06	93.77	13.00	83.20	25.90	94.12	10.37
Qwen3-Omni-30B-A3B-Thinking	**100.0**	00.20	98.30	07.07	96.00	11.00	81.69	25.70	**100.0**	00.25	99.93	00.96	93.97	12.75	84.14	23.70	94.25	10.20
InternLM-XComposer-2.5-OL	99.37	03.49	57.46	45.16	76.43	29.70	45.63	53.00	99.23	03.84	98.44	06.82	69.68	35.37	57.53	44.53	75.47	27.74
gemma-3n-E4B	99.39	03.80	78.03	28.84	79.60	28.40	57.01	44.50	99.64	02.80	99.43	03.92	79.85	28.00	80.40	28.00	84.17	21.03
gemma-3n-E4B-it	99.69	02.61	79.31	28.03	81.47	27.40	55.89	46.40	99.80	01.96	99.56	03.23	81.74	27.17	81.65	27.30	84.89	20.51
gemma-3n-E2B	99.00	04.96	73.12	33.27	74.94	33.16	52.51	48.70	99.23	04.03	98.98	04.96	75.44	31.74	73.97	34.71	80.90	24.44
gemma-3n-E2B-it	99.40	03.78	73.09	33.42	81.22	26.90	54.34	46.57	99.68	02.76	99.39	03.86	82.08	25.30	81.97	26.40	83.90	21.12
Phi-4-multimodal-instruct	94.89	12.09	64.92	38.63	60.31	43.45	48.90	51.05	96.80	09.02	96.46	09.36	64.37	40.13	61.30	41.85	73.49	30.70
Step-Audio-2-mini	98.30	06.47	60.72	42.37	74.61	32.85	45.22	55.00	97.38	08.61	96.30	09.85	69.77	36.65	65.02	39.50	75.91	28.91
Step-Audio-2-mini-Base	97.93	07.37	61.17	42.21	74.60	31.75	46.32	52.80	97.54	08.11	96.41	09.08	66.94	37.87	62.88	41.05	75.47	28.78
Step-Audio-2-mini-Think	97.36	08.83	58.48	43.79	72.48	34.15	46.09	52.50	96.82	09.43	94.86	11.60	67.71	37.63	63.07	40.70	74.61	29.83
Qwen2-Audio-7B-Instruct	99.99	00.33	90.76	17.28	93.51	14.65	66.16	37.75	**100.00**	00.29	99.94	01.06	93.24	14.90	84.70	21.93	91.04	13.52
Qwen2.5-Omni-3B	99.97	00.71	66.38	38.48	87.79	21.50	46.77	53.75	99.97	00.63	99.87	01.69	87.95	20.20	83.73	24.90	84.05	20.23
Qwen2.5-Omni-7B	99.97	00.67	59.87	43.47	85.68	23.30	44.16	55.40	99.97	00.73	99.86	01.63	85.15	23.38	76.81	31.07	81.43	22.46
AASIST	92.24	16.12	84.69	23.27	65.86	37.25	58.93	43.30	95.48	11.92	91.98	16.03	64.95	38.67	55.23	46.25	76.17	29.10
W2V2+AASIST	99.99	00.25	99.48	03.34	97.85	06.25	91.95	15.60	99.99	00.28	99.37	03.46	97.53	06.93	83.53	24.90	96.21	07.63
BEATs+AASIST	**100.00**	00.05	**100.00**	00.18	98.14	03.20	98.03	04.35	**100.00**	00.07	99.94	01.01	98.02	04.38	91.16	15.88	98.16	03.64
RAIS	**100.00**	00.18	98.95	04.62	94.81	09.05	85.55	21.65	99.99	00.45	99.35	03.96	94.73	09.25	79.63	26.98	94.12	09.52
DSSL	99.99	00.40	98.98	04.90	96.96	08.20	87.55	20.30	**100.00**	00.23	**99.97**	**00.76**	96.98	07.95	95.03	11.03	96.93	06.72
SRAM (ours)	**100.00**	**00.00**	**100.00**	**00.03**	**99.48**	**01.55**	**99.45**	**01.40**	99.99	**00.03**	99.20	02.81	**99.59**	**01.40**	**96.71**	**09.10**	**99.30**	**02.04**

in-domain and out-of-domain conditions, allowing us to evaluate detection performance and generalization. Table 1 summarizes the data splits and their abbreviated names.

Baselines. For task-specific discriminative models, we compare against AASIST (Jung et al., 2022), W2V2+AASIST (Tak et al., 2022), BEATs+AASIST (Yin et al., 2025), RAIS (Febrinanto et al., 2025), and DSSL (Fan et al., 2026). For generative models, we compare against Qwen2-Audio-7B, Qwen2-Audio-7B-Instruct (Chu et al., 2024), Qwen2.5-Omni-3B, Qwen2.5-Omni-7B (Xu et al., 2025), Qwen3-Omni-30B-A3B-Instruct, Qwen3-Omni-30B-A3B-Thinking (Team, 2025b), InternLM-XComposer-2.5-OL (Zhang et al., 2024), Gemma-3n-E4B, Gemma-3n-E4B-it, Gemma-3n-E2B, Gemma-3n-E2B-it (Team, 2025a), Phi-4-multimodal-instruct (Abouelenin et al., 2025), Step-Audio-2-mini, Step-Audio-2-mini-Base, and Step-Audio-2-mini-Think (Team, 2025c).

4.2 Zero-shot Evaluation of ALLMs

Figure 3 reports the average zero-shot AUC and EER of the 15 ALLMs across the eight test subsets. The per-subset results are given in Table 7. The models show limited zero-shot ability to distinguish real from synthetic environmental audio: mean AUC ranges from 47.77% to 61.35%, while mean EER remains at or above 41.95%. Activating more parameters at inference does not always guarantee better performance: Qwen3-Omni-30B-A3B-Instruct achieves 56.51% AUC, compared with 54.66% for Qwen2-Audio-7B-Instruct. These results indicate that current general-purpose ALLMs cannot effectively perform the ESDD task without task-specific adaptation.

4.3 Evaluation after Supervised Fine-Tuning

Table 2 reports the results of the generative ALLMs and discriminative detectors after SFT. Following ALLM4ADD (Gu et al., 2025), all ALLMs retain generative prediction and adapt the LLM component using LoRA (Hu et al., 2022); the audio encoder is frozen by default, while the three italicized variants additionally fine-tune the entire audio encoder. Compared with the zero-shot results in Section 4.2, supervised fine-tuning substantially improves every ALLM, increasing mean AUC by 15–38 percentage points. The Qwen3-Omni variants perform best among the ALLMs, achieving mean AUCs of 94.12% and 94.25%. Although these mixture-of-experts models activate only a fraction of their parameters during inference, they clearly outperform Qwen2.5-Omni-7B, which activates more parameters but reaches only 79.45% mean AUC. Moreover, although fully fine-tuning the audio encoder improves average performance across the models, the improvement comes at the cost of substantially higher computational overhead, including increased training time. Despite being pretrained with broad, general-purpose audio knowledge, ALLMs still suffer larger

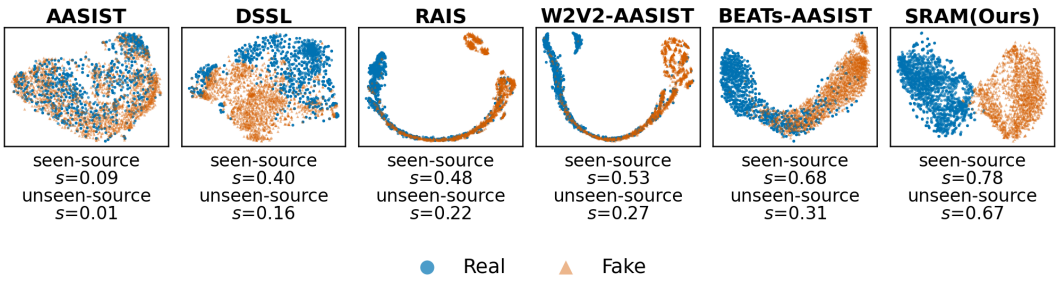


Figure 4: t-SNE visualizations of the clip embeddings fed into the classifier heads of six detectors under a shared configuration. The "seen-source" and "unseen-source" indicate whether the source dataset is included in the training data; s is the mean Real/Fake silhouette score over AT1, AT2, TT1, TT2 (seen) or AT3, AT4, TT3, TT4 (unseen).

performance drops on subsets from unseen datasets after fine-tuning, suggesting that task-specific fine-tuning does not fully address their generalization limitations.

Among the discriminative models, SRAM reaches 99.30% mean AUC and 2.04% mean EER, against 98.16% and 3.64% for the previous best system BEATs+AASIST, reducing the mean EER by approximately 44%. SRAM is built on the same pretrained audio encoder as Qwen3-Omni-30B-A3B-Instruct but lowers the mean EER. These results suggest that combining audio representations with local spectral artifact cues improves detection performance. The gains are most evident on the cross-domain subsets. On AT1 and TT1, where both the source datasets and generators are seen, strong models already achieve near-saturated AUCs of 99.99–100.00%. In contrast, AT3, AT4, TT3, and TT4 are more challenging because their source datasets are unseen, while AT4 and TT4 also contain unseen generators. SRAM achieves the best AUC and EER on the four subsets, demonstrating stronger generalization to unseen datasets and generators. The ablation study in Table 4 further validates the effectiveness of our proposed design.

4.4 Representation Geometry Visualization

To evaluate the discriminability of the learned representations, we focus on discriminative models, which generally outperform generative models in Table 2. We also compare with AASIST because both W2V2+AASIST and BEATs+AASIST use it as their backbone. Figure 4 presents t-SNE visualizations of the classifier-input embeddings of these six detectors. We quantify Real/Fake separation using silhouette scores in the projected space, computed within each test subset and averaged separately over seen and unseen sources. AASIST shows substantial overlap between real and synthesized samples. W2V2+AASIST, RAIS, and BEATs+AASIST arrange the two classes along an arc with overlap near the bottom, while DSSL forms two spread-out, overlapping groups. In contrast, SRAM forms two more compact groups with a clearer gap and limited overlap. It also achieves the highest silhouette scores on both seen and unseen sources, suggesting that our framework learns more discriminative representations that generalize to unseen synthesis domains.

4.5 Robustness to Unseen Audio Perturbations

In real-world scenarios, environmental audio may undergo noise interference and signal processing during recording, transmission, and playback, making deepfake detection more difficult. We evaluate five detectors under Gaussian noise, resampling, high-pass filtering, and same-label mixing, each at two severity levels, with detailed settings provided in Appendix A.4. These perturbations can mask local synthesis cues or remove frequency information that helps distinguish real from generated audio. As shown in Table 3, stronger Gaussian noise, more aggressive resampling, and higher high-pass cutoffs increase the EER of all evaluated detectors. Despite these challenges, SRAM achieves the lowest EER under both levels of Gaussian noise, high-pass filtering, and same-label mixing, while BEATs+AASIST performs best under resampling. The strong performance across these conditions suggests that combining audio representations with local spectral and geometric artifact cues helps SRAM retain useful detection cues when the input is distorted. Overall,

Table 3: Mean EER (%, ↓) for each perturbation type and severity level, computed separately on the eight EnvSDD test subsets and then averaged across subsets. We evaluate the five models with the lowest mean EER on the original test audio in Table 2. Each perturbation has two severity levels; Avg averages all eight conditions.

Model	Gaussian noise		Resampling		High-pass filtering		Same-label mix		**Avg**
	30 dB	20 dB	0.75×	0.50×	500 Hz	1333 Hz	30 dB	20 dB	
BEATs+AASIST	11.24	14.78	**09.08**	**10.51**	15.50	23.10	04.05	03.71	11.50
W2V2+AASIST	16.64	21.31	14.30	28.78	17.09	21.82	07.81	08.32	17.01
DSSL	13.78	21.30	12.97	14.54	03.48	04.48	02.30	02.09	09.37
RAIS	17.39	22.43	17.17	27.30	17.61	25.21	08.65	09.16	18.12
SRAM	**09.59**	**13.64**	16.27	22.49	**02.83**	**03.73**	**01.60**	**01.48**	**08.95**

Table 4: Component removal on the EnvSDD test subsets, in AUC (%, ↑) and EER (%, ↓). Avg is the mean over the eight subsets. Full Model is the reported run of Table 2.

Variant	AT1		AT2		AT3		AT4		TT1		TT2		TT3		TT4		**Avg**	
	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓
Full Model	100.00	00.00	100.00	00.03	99.48	01.55	99.45	01.40	99.99	00.03	99.20	02.81	99.59	01.40	96.71	09.10	99.30	02.04
w/o spectrogram-residual branch	99.98	00.08	99.96	00.23	99.47	01.60	99.44	02.20	99.94	00.33	98.28	04.16	99.01	04.80	95.06	11.60	98.89	03.12
w/o structure-tensor branch	100.00	00.08	100.00	00.15	99.47	01.90	99.46	02.10	99.96	00.18	99.13	02.80	99.47	02.70	96.40	10.15	99.24	02.51
w/o Spectral Relation Modeling	99.98	00.13	99.97	00.43	99.10	03.10	98.81	04.60	99.97	00.47	99.07	04.41	98.56	05.28	89.04	20.50	98.06	04.86
w/o trajectory refiner	100.00	00.05	99.97	00.25	98.92	02.80	98.86	02.80	99.79	00.36	96.02	05.51	98.57	04.60	92.40	14.33	98.07	03.84
dense pooling	100.00	00.08	100.00	00.20	99.54	01.70	99.49	01.90	100.00	00.10	99.51	02.64	99.39	02.90	96.16	11.00	99.26	02.56

SRAM achieves the lowest average EER across all eight perturbation conditions, demonstrating the strongest overall performance in this robustness evaluation.

4.6 Ablation Study

SRAM combines audio representation learning and spectral relation modeling to capture synthesis artifacts. Cross-layer trajectory refinement and sparse evidence pooling refine and aggregate the audio representations, while two spectral branches provide local artifact cues. We investigate the role of each component through ablations in Table 4, with additional analyses in Appendix A.6.

Audio representation learning. Removing the cross-layer trajectory refiner clearly reduces detection performance, especially on subsets with unseen source datasets. This suggests that information from multiple encoder depths helps refine the audio representations for detecting synthesis artifacts. Replacing sparse evidence pooling with dense attention also leads to worse average performance, with larger errors on TT3 and TT4. These results support selecting informative temporal tokens when aggregating local artifact cues. Together, trajectory refinement and sparse pooling improve the use of pretrained audio representations for detection.

Spectral relation modeling. Removing either spectral branch degrades average performance. Without the spectrogram-residual branch, the model loses explicit local spectral and temporal differences that may be less apparent in the audio encoder representations. Removing the structure-tensor branch also reduces performance despite retaining residual cues, suggesting that gradient activity and directional coherence provide complementary information. Removing both branches causes the largest degradation, particularly on TT4, highlighting the value of explicit spectral cues beyond the learned audio representations for generalization.

5 Conclusion

In this paper, we evaluate 15 ALLMs from five model families on EnvSDD under zero-shot and supervised fine-tuning settings. Their limited zero-shot performance and their gap behind strong task-specific discriminative detectors after fine-tuning highlight the challenges of applying general-purpose ALLMs to ESDD. We propose SRAM, which combines pretrained audio representations with local spectro-temporal residuals and time-frequency geometry. Cross-layer trajectory refinement and sparse evidence pooling integrate and aggregate these cues for detection. Extensive experiments demonstrate SRAM's strong detection performance, generalization to unseen domains, and robustness to unseen perturbations. We hope our work can facilitate further research on model design and generalization in ESDD and other deepfake detection scenarios.

REFERENCES

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Jun-Kun Chen, Weizhu Chen, Yen-Chun Chen, Yi-ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, Mei Gao, Min Gao, Amit Garg, Abhishek Goswami, Junheng Hao, Amr Hendy, Yuxuan Hu, Xin Jin, Mahmoud Khademi, Dongwoo Kim, Young Jin Kim, Gina Lee, Jinyu Li, Yunsheng Li, Chen Liang, Xihui Lin, Zeqi Lin, Mengchen Liu, Yang Liu, Gilsinia Lopez, Chong Luo, Piyush Madan, Vadim Mazalov, Arindam Mitra, Ali Mousavi, Anh Nguyen, Jing Pan, Daniel Perez-Becker, Jacob Platin, Thomas Portet, Kai Qiu, Bo Ren, Liliang Ren, Sambuddha Roy, Ning Shang, Yelong Shen, Saksham Singhal, Subhojit Som, Xia Song, Tetyana Sych, Praneetha Vaddamanu, Shuohang Wang, Yiming Wang, Zhenghao Wang, Haibin Wu, Haoran Xu, Weijian Xu, Yifan Yang, Ziyi Yang, Donghan Yu, Ishmam Zabir, Jianwen Zhang, Li Lyna Zhang, Yunan Zhang, and Xiren Zhou. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025.

Boson AI. Higgs Audio V2: Redefining Expressiveness in Audio Generation. https://github.com/boson-ai/higgs-audio, 2025.

Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, Wanxiang Che, Xiangzhan Yu, and Furu Wei. Beats: Audio pre-training with acoustic tokenizers. In *ICML*, 2023.

Xiaohuan Chen, Wenhuan Lu, Ruiteng Zhang, Junhai Xu, Xugang Lu, Lin Zhang, and Jianguo Wei. Continual unsupervised domain adaptation for audio deepfake detection. In *ICASSP*, 2025a.

Yujie Chen, Jiangyan Yi, Cunhang Fan, Jianhua Tao, Yong Ren, Siding Zeng, Chu Yuan Zhang, Xinrui Yan, Hao Gu, Jun Xue, Chenglong Wang, Zhao Lv, and Xiaohui Zhang. Region-based optimization in continual learning for audio deepfake detection. In *AAAI*, 2025b.

Haonan Cheng, Kangyue Li, Long Ye, and Jingling Wang. Envfake: An initial environmental-fake audio dataset for scene-consistency detection. In *ISCSLP*, 2024.

Keunwoo Choi, Jaekwon Im, Laurie M. Heller, Brian McFee, Keisuke Imoto, Yuki Okamoto, Mathieu Lagrange, and Shinnosuke Takamichi. Foley sound synthesis at the DCASE 2023 challenge. *arXiv preprint arXiv:2304.12521*, 2023.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*, 2023.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

Thien-Phuc Doan, Long Nguyen-Vu, Souhwan Jung, and Kihun Hong. BTS-E: audio deepfake detection using breathing-talking-silence encoder. In *ICASSP*, 2023.

Cunhang Fan, Junqin Cao, Tian Gao, Zhipeng Xie, Jun Xue, Zhao Lv, and Xin Fang. Beyond speech: Dual-domain ssl fusion for unified all-type audio deepfake detection. In *ACM MM 2026*, 2026.

Falih Gozi Febrinanto, Kristen Moore, Chandra Thapa, Jiangang Ma, Vidya Saikrishna, and Feng Xia. Rehearsal with auxiliary-informed sampling for audio deepfake detection. In *Interspeech*, 2025.

Hao Gu, Jiangyan Yi, Chenglong Wang, Jianhua Tao, Zheng Lian, Jiayi He, Yong Ren, Yujie Chen, and Zhengqi Wen. ALLM4ADD: unlocking the capabilities of audio large language models for audio deepfake detection. In *ACM MM*, 2025.

Tamara Gurevich, Peter R. Herman, Farid Toubal, and Yoto V. Yotov. A dataset on linguistic connectivity across and within countries. *Scientific Data*, 2025.

Yunqi Hao, Minqiang Xu, Yihao Chen, Yanyan Liu, Liang He, Lei Fang, and Lin Liu. Integrating spectro-temporal cross aggregation and multi-scale dynamic learning for audio deepfake detection. In *ICASSP*, 2025.

Christopher G. Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference, AVC*, 1988.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

Zehui Jin, Linlong Lang, and Biao Leng. Wave-spectrogram cross-modal aggregation for audio deepfake detection. In *ICASSP*, 2025.

Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas W. D. Evans. AASIST: audio anti-spoofing using integrated spectro-temporal graph attention networks. In *ICASSP*, 2022.

Siva Rajesh Kasa, Karan Gupta, Sumegh Roychowdhury, Ashutosh Kumar, Yaswanth Biruduraju, Santhosh Kumar Kasa, Nikhil Priyatam Pattisapu, Arindam Bhattacharya, Shailendra Agarwal, and Vijay Huddar. Generative or discriminative? revisiting text classification in the era of transformers. In *EMNLP*, 2025.

Gautam Siddharth Kashyap, Zohaib Hasan Siddiqui, Mohammad Anas Azeez, Rafiq Ali, Shantanu Kumar, Navin Kamuni, and Jiechao Gao. Fooling the forgers: A multi-stage framework for audio deepfake detection. In *ICASSP*, 2025.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *NAACL-HLT*, 2019.

Linxi Li, Yuncong Yu, Qianwei Guo, Liwei Jin, Yechen Wang, and Carsten Maple. Synsfx: Multi-model sound effects synthesis dataset for deepfake detection and evaluation. *arXiv preprint arXiv:2607.04848*, 2026.

Chu-Hsuan Abraham Lin, Chen-Yu Liu, Samuel Yen-Chi Chen, and Kuan-Cheng Chen. Quantum-trained convolutional neural network for deepfake audio detection. In *ICASSP*, 2025.

Frederick Liu, Siamak Shakeri, Hongkun Yu, and Jing Li. Enct5: Fine-tuning T5 encoder for non-autoregressive tasks. *arxiv preprint arXiv:2110.08426*, 2021.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo P. Mandic, Wenwu Wang, and Mark D. Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. In *ICML*, 2023.

Huadai Liu, Rongjie Huang, Yang Liu, Hengyuan Cao, Jialei Wang, Xize Cheng, Siqi Zheng, and Zhou Zhao. Audiolcm: Efficient and high-quality text-to-audio generation with minimal inference steps. In *ACM MM*, 2024.

Huadai Liu, Wen Wang, Kaicheng Luo, Qian Chen, Xiangang Li, and Wei Xue. Star-vae: Structured topology-aware regularization for audio reconstruction and generation. In *ICML*, 2026.

André F. T. Martins and Ramón Fernandez Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *ICML*, 2016.

Sharmistha Modak, Arnab Kumar Das, and Ruchira Naskar. Specvit: A custom vision-transformer based approach for audio deepfake detection. In *ICASSP*, 2025.

Hafsa Ouajdi, Oussama Hadder, Modan Tailleur, Mathieu Lagrange, and Laurie M. Heller. Detection of deepfake environmental audio. In *EUSIPCO*, 2024.

Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. Layer-wise analysis of a self-supervised speech representation model. In *ASRU*, 2021.

Lekai Qian, Haoyu Gu, Jingwei Zhao, and Ziyu Wang. Beat: Tokenizing and generating symbolic music by uniform temporal steps. In *ICML*, 2026.

Muhammad A. Shah, David Solans Noguero, Mikko A. Heikkilä, Bhiksha Raj, and Nicolas Kourtellis. Speech robust bench: A robustness benchmark for speech recognition. In *ICLR*, 2025.

Ganesh Sivaraman, Hemlata Tak, and Elie Khoury. Investigating voiced and unvoiced regions of speech for audio deepfake detection. In *ICASSP*, 2025.

Hemlata Tak, Massimiliano Todisco, Xin Wang, Jee-weon Jung, Junichi Yamagishi, and Nicholas W. D. Evans. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. In *Odyssey 2022: The Speaker and Language Recognition Workshop*, 2022.

DeepL Team. Deepl. https://www.deepl.com/zh/translator, 2026.

Gemma Team. Gemma 3n. https://ai.google.dev/gemma/docs/gemma-3n, 2025a.

Qwen Team. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025b.

StepFun Audio Team. Step-audio 2 technical report. *arXiv preprint arXiv:2507.16632*, 2025c.

Jiarui Wang, Huiyu Duan, Juntong Wang, Ziheng Jia, Woo Yi Yang, Xiaorong Zhu, Yu Zhao, Jiaying Qian, Yuke Xing, Guangtao Zhai, and Xiongkuo Min. Dfbench: Benchmarking deepfake image detection capability of large multimodal models. In *ACM MM*, 2025.

Yuankun Xie, Xiaoxuan Guo, Jiayi Zhou, Tao Wang, Jian Liu, Ruibo Fu, Xiaopeng Wang, Haonan Cheng, and Long Ye. Interpretable all-type audio deepfake detection with audio llms via frequency-time reinforcement learning. *arxiv preprint arXiv:2601.02983*, 2026.

Zeyu Xie, Baihan Li, Xuenan Xu, Zheng Liang, Kai Yu, and Mengyue Wu. Fakesound: Deepfake general audio detection. In *Interspeech*, 2024.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025.

Jiangyan Yi, Chenglong Wang, Jianhua Tao, Chuyuan Zhang, Cunhang Fan, Zhengkun Tian, Haoxin Ma, and Ruibo Fu. Scenefake: An initial dataset and benchmarks for scene fake audio detection. *arXiv preprint arXiv:2211.06073*, 2024.

Han Yin, Yang Xiao, Rohan Kumar Das, Jisheng Bai, Haohe Liu, Wenwu Wang, and Mark D. Plumbley. Envsdd: Benchmarking environmental sound deepfake detection. In *Interspeech*, 2025.

Pan Zhang, Xiaoyi Dong, Yuhang Cao, Yuhang Zang, Rui Qian, Xilin Wei, Lin Chen, Yifei Li, Junbo Niu, Shuangrui Ding, Qipeng Guo, Haodong Duan, Xin Chen, Han Lv, Zheng Nie, Min Zhang, Bin Wang, Wenwei Zhang, Xinyue Zhang, Jiaye Ge, Wei Li, Jingwen Li, Zhongying Tu, Conghui He, Xingcheng Zhang, Kai Chen, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer2.5-omnilive: A comprehensive multimodal system for long-term streaming video and audio interactions. *arXiv preprint arXiv:2412.09596*, 2024.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. SWIFT: A scalable lightweight infrastructure for fine-tuning. In *AAAI*, 2025.

A Appendix

A.1 AI use statement

We use ChatGPT and Claude for minor grammatical adjustments to the manuscript. All resulting edits are carefully reviewed and finalized by the authors, who take full responsibility for the final content of this work.

A.2 Reproducibility statement

To facilitate reproducibility, we describe the model architecture and algorithms in Section 3 and provide implementation details in Appendix A.5. Experimental settings are presented in Section 4.1, with training configurations, hyperparameters, and the computing environment detailed in Appendix A.3. Upon acceptance, we will publicly release the source code and model weights.

A.3 Experimental Details

All experiments are conducted on a single NVIDIA Pro6000 GPU. For SRAM, all parameters are trained end-to-end, and the audio encoder is initialized from the weights of `Qwen3-Omni-30B-A3B-Instruct`, and the encoded tokens are projected to a feature dimension of $D = 512$. The cross-layer trajectory refiner taps four equally spaced encoder layers, at 25%, 50%, 75%, and 100% of the encoder depth, and refines the conditioned tokens for $R = 2$ iterations. The sparse evidence pool retains the top $\rho = 0.25$ of tokens by score. Training uses a dropout rate of 0.15 and a label-smoothing factor of 0.05 in the classification loss $\mathcal{L}_{\text{cls}}$. All ALLMs receive the same prompt: 'You are a sound forensics expert who can accurately determine whether a sound clip is real or fake. Analyze whether the `<audio>` is real or fake. If it is real, output 0; otherwise, output 1. Output only 0 or 1, with no additional text.'

For the generative methods, all 15 models are run with the `ms-swift` framework (Zhao et al., 2025), and prompts are written in English unless otherwise stated. For models that require fine-tuning, we use LoRA (Hu et al., 2022) with rank 64, $\alpha = 16$, and dropout 0.05. As indicated in Table 2, a trainable audio encoder means that all of its parameters are updated, whereas the LLM component is adapted using LoRA. This configuration follows ALLM4ADD (Gu et al., 2025). These models are trained in `bfloat16` precision with an effective batch size of 32 through gradient accumulation. We use AdamW with an initial learning rate of 5×10^{-5}, which decays to zero following a cosine schedule. Based on the experiments in Appendix A.6.12, we train each model for at most 10 epochs and evaluate the checkpoint with the best validation performance. The evaluation logits for generative models are computed following the protocol adopted in DFBench (Wang et al., 2025).

For the discriminative methods, training is likewise carried out in `bfloat16` precision, so that both tracks share the same numerical precision and the comparison remains fair. We use a batch size of 32 and optimize with AdamW at an initial learning rate of 5×10^{-5}, again annealed to zero on a cosine schedule. The discriminative models are trained for at most 50 epochs, with early stopping if validation performance does not improve for 10 consecutive epochs. The checkpoint with the best validation performance is used for evaluation. The EnvSDD study (Yin et al., 2025) does not specify whether audio perturbations are applied during benchmark construction and training. We therefore assume that no such perturbations are introduced and apply no additional perturbation-based data augmentation to the training audio in any experiment. This setup also facilitates the evaluation of robustness to test-time perturbations.

SynSFX (Li et al., 2026) introduces an approximately 178-hour dataset for ESDD, focusing on isolated sound effects generated by text-to-audio models. We intend to include SynSFX in our experimental evaluation; however, although the dataset is reported as publicly available, the provided dataset link is inaccessible throughout our experiments. Consequently, the dataset is not publicly obtainable at the time of our evaluation, preventing us from including it in this study.

For the representation visualizations in Section 4.4, we apply identical preprocessing and t-SNE settings across all detectors to improve comparability. For each of the eight test subsets, we randomly sample 200 real and 200 synthesized audio clips without replacement, yielding 3,200 clips per de-

tector. We compute silhouette scores using Euclidean distances between the two-dimensional t-SNE embeddings. For each sample i, let $a(i)$ be its average distance to the other samples of the same class (real or fake), and let $b(i)$ be its average distance to samples of the opposite class, both within the same test subset. The silhouette score of sample i is

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{9}$$

The score for a test subset $\mathcal{D}$ is the average over its samples:

$$S(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} s(i). \tag{10}$$

We then report the mean subset score separately for seen and unseen sources:

$$S_{\text{seen}} = \frac{1}{4} \sum_{\mathcal{D} \in \mathcal{T}_{\text{seen}}} S(\mathcal{D}), \qquad S_{\text{unseen}} = \frac{1}{4} \sum_{\mathcal{D} \in \mathcal{T}_{\text{unseen}}} S(\mathcal{D}), \tag{11}$$

where $\mathcal{T}_{\text{seen}} = \{\text{AT1}, \text{AT2}, \text{TT1}, \text{TT2}\}$ and $\mathcal{T}_{\text{unseen}} = \{\text{AT3}, \text{AT4}, \text{TT3}, \text{TT4}\}$. Scores range from -1 to 1, with higher values indicating better separation between real and synthesized samples in the projected space.

A.4 Robustness Evaluation Protocol

We apply four types of perturbations at two severity levels to the EnvSDD test clips. Gaussian noise, resampling, and high-pass filtering are implemented based on the open-source code of SRB (Shah et al., 2025). The original test audio serves as the reference for constructing these inputs. For each condition, we report the mean EER over the eight test subsets; the overall average gives equal weight to all eight perturbation conditions.

Gaussian noise. We add zero-mean Gaussian white noise to the waveform, scaling its amplitude according to the original signal power to obtain an SNR of 30 or 20 dB. The lower SNR introduces stronger interference.

Resampling. We downsample the original 16 kHz audio to 12 or 8 kHz, corresponding to ratios of 0.75 and 0.50, and then resample it back to 16 kHz. Restoring the sampling rate does not recover the frequency information lost during downsampling. This setting tests sensitivity to reduced bandwidth and resampling artifacts.

High-pass filtering. We apply a high-pass filter to the waveform with a cutoff frequency of 500 or 1333 Hz. The higher cutoff removes more low-frequency content. This setting tests whether detection remains effective when low-frequency cues are weakened and the model must rely more on the remaining frequency components.

Same-label mixing. For each test clip, we deterministically select a different clip with the same label from the same test set and mix it into the original waveform at an SNR of 30 or 20 dB. Real clips are mixed only with real clips, and generated clips only with generated clips, avoiding direct mixing of the two target classes. Labels are used only to construct the test inputs and are not provided to the detector. We implement this custom label-conditioned perturbation ourselves, following the mixing approach of SRB's `env_noise_musan` condition but using same-label test clips instead of MUSAN noise.

A.5 Model Details

This section gives the implementation details of the spectral branches, trajectory refiner, and sparse pooling in Section 3. Training settings are listed in Appendix A.3. Throughout the method, $\sigma(\cdot)$ denotes the sigmoid function, $\odot$ denotes elementwise multiplication, and $[\,;\,]$ denotes feature concatenation. FFN and LN denote a feed-forward network and layer normalization, respectively.

Spectral relation modeling. Both branches standardize each mel-spectrogram using its mean and variance over time and frequency, with 10^{-5} added to the variance before taking the square root. Local means use average pooling with replicated boundary values, and derivatives use first-order finite differences. The spectrogram-residual branch uses the five maps defined in Section 3.3. The structure-tensor branch computes e, c, and i at 5×5 and 13×17 windows (frequency bins $\times$ time frames). These six maps are concatenated with the frequency gradient, temporal gradient, sum of frequency and temporal second derivatives, and mixed time-frequency derivative, giving ten input channels. Tensor entries are local averages of gradient products; the denominators of c and i include 10^{-4} for numerical stability. Each branch uses two convolutional stages with 32 and 64 output channels, group normalization, and GELU. The kernel sizes are 7×5 and 5×5 for the residual branch, and 5×5 and 3×5 for the structure-tensor branch. Each stage has stride $(2, 1)$; the second stage uses depthwise convolution followed by a pointwise convolution. Adaptive average pooling produces eight frequency bins and N temporal positions. Flattening the frequency and channel dimensions and applying a pointwise projection gives $N \times D$ features, with $D = 512$. The geometry gate uses $\gamma_{\max} = 0.10$ and is initialized to zero.

Token conditioning and refinement. The final audio encoder states pass through layer normalization, a linear projection to D dimensions, GELU, and dropout. A gate computed from these tokens and the spectral artifact features controls their fusion. The trajectory refiner uses four encoder depths and two refinement steps, as specified in Appendix A.3. At each temporal position, it concatenates the mean and standard deviation of the projected layer states, the mean normalized inter-layer difference, and the mean absolute change in that direction. A two-layer network maps these $4D$ statistics to the D-dimensional trajectory code. The update network has hidden width $2D$, and both refinement steps share its weights and the gate network. Each step size is a sigmoid-transformed scalar initialized to -1.5 before the sigmoid. Layer normalization is applied after the final refinement step.

Token residual scoring and pooling. Let $\mathbf{Q} = \mathrm{LN}(\mathbf{Z}_R)$ denote the tokens passed to the pooling module, and let $\mathbf{u}$ and $\mathbf{v}$ be their per-channel temporal mean and standard deviation. For $\hat{\mathbf{Q}}_n = (\mathbf{Q}_n - \mathbf{u})/\mathbf{v}$, the two inputs to the token scorer are constructed as

$$\mathbf{r}_n^{(k)} = \hat{\mathbf{Q}}_n - \mathrm{Avg}_{h_k}(\hat{\mathbf{Q}})_n, \qquad \mathbf{d}_n = \frac{\mathbf{Q}_n - \mathrm{Avg}_9(\mathbf{Q})_n}{\mathbf{v}}, \quad h_k \in \{3, 7, 15\}, \tag{12}$$

where Avg_h is a centered temporal moving average over h tokens with replicated boundaries. A window of 9 tokens includes the current token and four tokens on either side. Thus, $\mathbf{d}_n$ is computed from the refined token sequence. Both scoring networks use layer normalization, a hidden layer of width $D/4 = 128$, GELU, and a scalar output. Squared-energy terms are bounded below by 10^{-6} before taking logarithms. We retain the highest-scoring 25% of tokens (at least two when available) and apply softmax with temperature 0.5. Attention weights aggregate the signed residuals, absolute local deviations, and standard deviation of residuals across scales. The remaining statistic, $\mathbf{a}_{\mathrm{tail}}$, averages the largest $\lceil \rho N \rceil$ absolute residual values separately in each channel; these positions need not match the attention-selected tokens. The four statistics are concatenated with the token mean and standard deviation, then mapped from $6D$ to $2D$ to D with GELU and dropout between the linear layers. The resulting correction is scaled by a sigmoid gate initialized with logit -1.25, added to the token mean, and layer-normalized before classification.

A.6 Experimental Results

A.6.1 Qualitative motivation for spectrogram-residual artifact cues.

Figure 5 provides two additional real–generated pairs, following the visualization in Figure 1(I). The residual maps expose local spectral and temporal differences between each real audio clip and its AudioLDM1-generated counterpart, providing qualitative support for spectrogram-residual artifact cues. Black boxes mark corresponding regions selected by the largest smoothed difference in the joint maps.

A.6.2 Qualitative motivation for structure-tensor artifact cues.

Similar overall spectral energy can conceal differences in local time–frequency geometry. Figure 6 illustrates real–synthetic pairs with highly correlated band-energy profiles, measured by

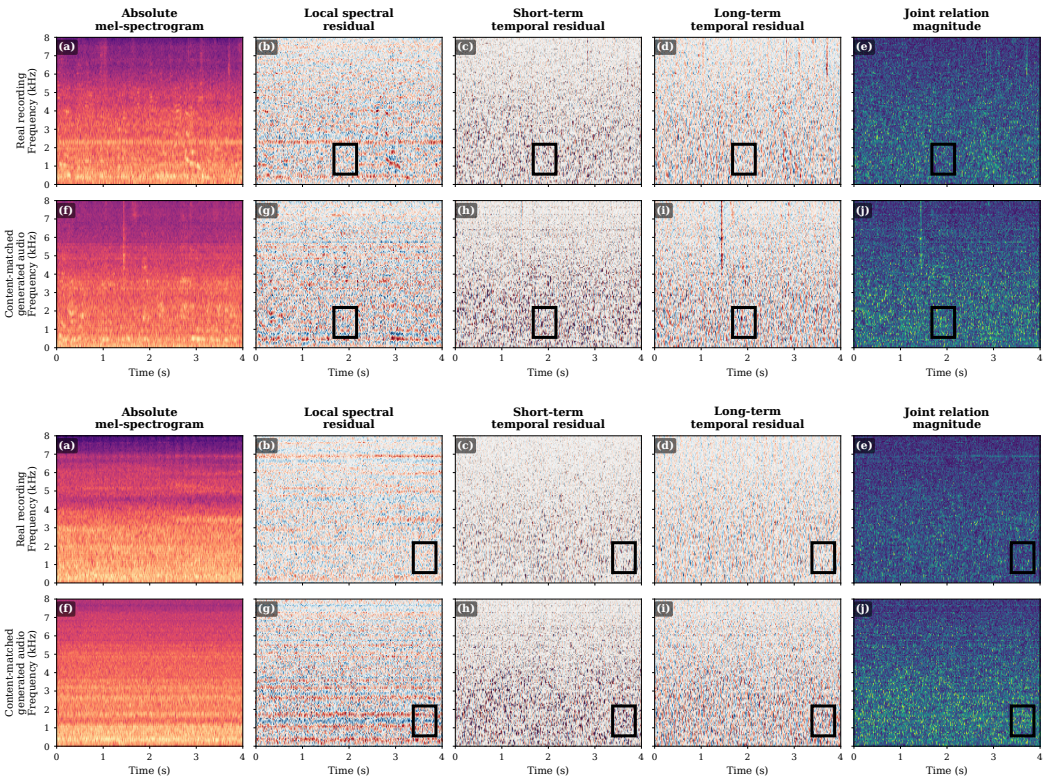


Figure 5: Additional examples of local spectral and temporal residuals: engine idling (upper pair) and jackhammer (lower pair). Each pair compares a real audio clip (top) with its content-matched ATA/AudioLDM1-generated counterpart (bottom). Columns show the mel-spectrogram, spectral residual, short- and long-term temporal residuals, and joint residual magnitude. Color scales are shared within each column of each pair.

$r_{\text{band}} = \text{corr}(\mathbf{p}^{\text{real}}, \mathbf{p}^{\text{syn}})$, where $p_f = \log(T^{-1} \sum_{t=1}^{T} \exp(X_{t,f}))$ for a mel-spectrogram $\mathbf{X}$ with T frames. To characterize local geometry, we compute gradients $\mathbf{g} = [\partial_f \hat{\mathbf{X}}, \partial_t \hat{\mathbf{X}}]^\top$ on the per-clip standardized spectrogram $\hat{\mathbf{X}}$ and define

$$\mathbf{S}_\Omega = \frac{1}{|\Omega|} \sum_{(t,f)\in\Omega} \mathbf{g}_{t,f} \mathbf{g}_{t,f}^\top, \qquad e = \log(1 + \text{tr}\, \mathbf{S}_\Omega), \qquad c = \frac{\sqrt{(S_{11} - S_{22})^2 + 4S_{12}^2}}{\text{tr}\, \mathbf{S}_\Omega + \epsilon}, \tag{13}$$

where Ω is a 13×17 local window (frequency bins $\times$ time frames), $|\Omega| = 221$ is the number of elements in the window, S_{ij} denotes an entry of the locally averaged tensor, and $\epsilon = 10^{-4}$ ensures numerical stability. Gradient activity e measures the strength of local variation, while directional coherence c measures its alignment. Despite similar band-energy profiles, the generated examples exhibit reduced activity (C), reduced coherence at nearly unchanged activity (D), or both (E). These observations suggest that local variation and directional organization provide complementary cues beyond marginal spectral energy, motivating the structure-tensor branch of SRAM as a starting point for capturing such synthesis artifact cues.

A.6.3 Contribution of Spectrogram Residuals.

Our spectrogram-residual branch captures local spectral and temporal differences through five residual maps. Here, we investigate the contribution of each map on the eight EnvSDD test subsets. Specifically, we keep the trained SRAM checkpoint fixed and evaluate all $2^5 = 32$ combinations of the maps. Maps excluded from a combination are replaced with zeros before the five-channel convolution, while the audio encoder, structure-tensor branch, and all remaining parameters are unchanged. To account for the dependence of a map's contribution on the other active maps, we

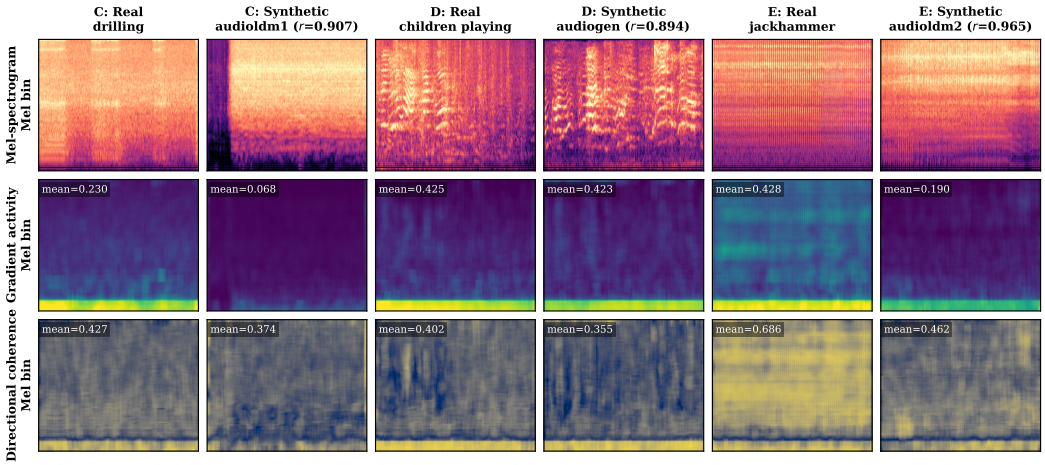


Figure 6: Additional matched real–synthetic cases motivating structure-tensor artifact cues. Columns form three pairs: drilling–AudioLDM1 (C), children playing–AudioGen (D), and jackhammer–AudioLDM2 (E). Rows show the mel-spectrogram, gradient activity e, and directional coherence c; real and synthetic panels within each pair use shared color limits. The title value r is r_{band} defined above, and each in-panel mean averages the displayed descriptor over time and frequency.

Table 5: Contributions of the five spectrogram-residual maps. The first five rows report exact Shapley values in pp; positive values indicate lower EER. The last two rows report EER (%) with all five maps zeroed or enabled. Zeroing the maps leaves the residual encoder and the other branches in place. Values are rounded to two decimal places; 00.00 may represent a small nonzero contribution.

Residual map / configuration	AT1	AT2	AT3	AT4	TT1	TT2	TT3	TT4
Spectral residual	00.00	00.00	+00.12	+00.11	00.00	−00.41	+00.08	+00.13
Short temporal residual	00.00	+00.01	−00.06	+00.09	+00.01	+00.05	+00.07	+00.27
Long temporal residual	00.00	00.00	−00.30	−00.42	+00.01	−00.07	−00.09	+00.95
Joint time-frequency transport	00.00	+00.01	+00.14	+00.17	00.00	+00.11	+00.06	+00.31
Temporal curvature	00.00	+00.01	00.00	+00.25	00.00	+00.33	+00.12	+00.54
All maps zeroed: EER (%)	00.00	00.05	01.45	01.60	00.05	02.81	01.63	11.30
All maps enabled: EER (%)	00.00	00.03	01.55	01.40	00.03	02.81	01.40	09.10

compute its exact Shapley value using negative EER as the utility:

$$\phi_i = \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{|S|!\,(4-|S|)!}{5!} \left[\mathrm{EER}(S) - \mathrm{EER}(S \cup \{i\})\right], \tag{14}$$

where $\mathcal{N}$ contains the five maps and S denotes a subset excluding map i. Positive values indicate a reduction in EER, measured in percentage points (pp). Each contribution is a weighted average over all 16 combinations of the other four maps. This experiment measures the trained model's reliance on residual inputs at inference time; it does not involve retraining after branch removal.

As shown in Table 5, enabling the residual maps clearly improves detection on the challenging TT4 subset, primarily through the long temporal residual and temporal curvature, while performance on AT1, AT2, and TT1 remains nearly perfect with or without these inputs. Shapley analysis identifies joint time-frequency transport as the most consistent contributor, improving or maintaining performance across all subsets, whereas other maps exhibit condition-dependent effects. Averaged equally over test subsets (Figure 7), temporal curvature and joint time-frequency transport provide the largest benefits. All five maps have higher mean contributions on unseen-source subsets than across the full suite, indicating that residual inputs provide more useful evidence under source shifts.

To explore whether the maps provide overlapping information, we also measure their pairwise interactions. Let $v(S) = -\frac{1}{8}\sum_{d=1}^{8} \mathrm{EER}_d(S)$, with EER expressed as a percentage. For two distinct

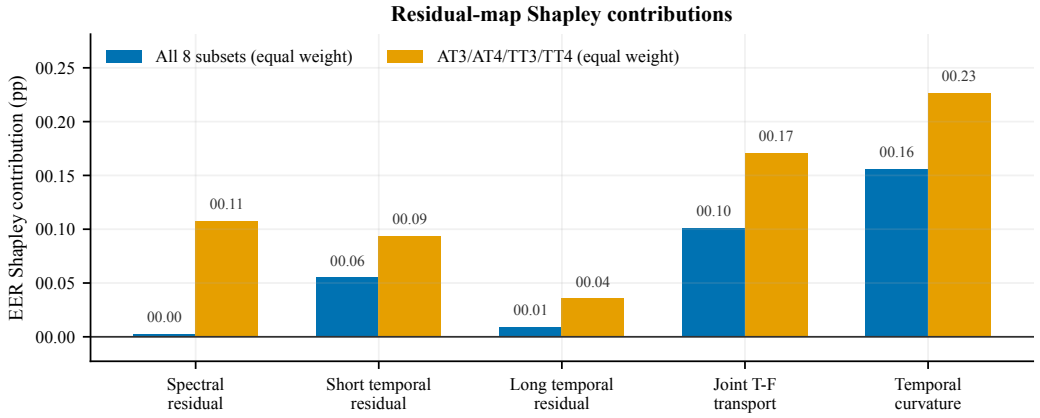


Figure 7: Mean EER Shapley contributions of the five residual maps under fixed-checkpoint intervention. Blue bars average over all eight test subsets; orange bars average over AT3, AT4, TT3, and TT4. Each subset receives equal weight. Contributions are measured in percentage points (pp), with positive values indicating an EER reduction.

maps i and j, the symmetric interaction matrix uses

$$I_{ij} = \sum_{S \subseteq \mathcal{N} \setminus \{i,j\}} \frac{|S|!\,(3-|S|)!}{2 \cdot 4!}\, \Delta_{ij} v(S), \qquad \Delta_{ij} v(S) = v(S \cup \{i,j\}) - v(S \cup \{i\}) - v(S \cup \{j\}) + v(S). \tag{15}$$

The factor $1/2$ assigns half of the weighted pair effect to each symmetric entry, $I_{ij} = I_{ji}$. Positive entries indicate synergy, while negative entries indicate redundancy or interference under this EER-based utility. As shown in Figure 8, joint time-frequency transport and temporal curvature exhibit the strongest redundancy or interference, despite having the largest individual contributions. This suggests that their benefits partly overlap. In comparison, the spectral and long temporal residuals show weak synergy, as do the long temporal residual and temporal curvature. Overall, the maps provide varying degrees of useful evidence across different conditions.

These overlapping and condition-dependent contributions motivate adaptive control over how residual information enters the audio tokens. In SRAM, the cross-gate in Section 3.2 conditions on both the audio tokens and the jointly encoded spectral features to regulate their contribution to each token. This provides feature-level control, although it does not explicitly select individual residual maps before encoding. We also explored such early map selection, but preliminary experiments did not yield consistent improvements over the current design.

A.6.4 Encoder vs. Full ALLM.

To test whether the full ALLM is necessary for ESDD, we compare three ALLMs with their audio-encoder-only counterparts in Figure 9. For each model, we retain the pretrained audio encoder, attach a two-layer MLP classifier, and jointly fine-tune the encoder and classifier. We compare this setup with full ALLMs using either a frozen or a fine-tuned audio encoder. Across the three models, Encoder + MLP performs comparably to or better than the corresponding Full ALLM, suggesting that the pretrained audio encoder already captures information useful for detecting low-level synthesis cues. The additional decoder does not consistently improve detection performance and may make such cues less directly accessible, as its training objectives primarily emphasize cross-modal alignment and generation. In contrast, a lightweight classifier can operate directly on the encoder representations for binary detection. Prior text-classification studies likewise report that encoder-based classification can be competitive with generation-based approaches (Liu et al., 2021; Kasa et al., 2025). More generally, environmental sound representations learned during pretraining may provide useful prior knowledge for ESDD, consistent with the findings of Yin et al. (2025).

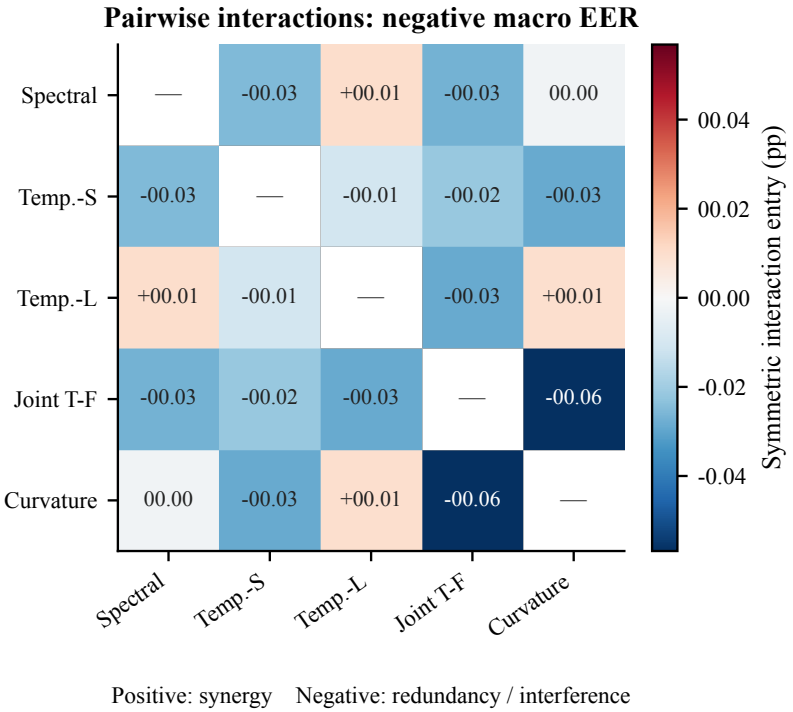


Figure 8: Pairwise residual-map interactions using negative EER averaged equally over the eight test subsets. Each off-diagonal cell reports I_{ij} from Equation 15 in pp, including the $1/2$ factor for symmetric entries. Red denotes synergy and blue denotes redundancy or interference. Values are rounded to two decimal places, so 00.00 may represent a small nonzero interaction. Diagonal entries are omitted; they do not represent the individual contributions in Figure 7. Temp.-S and Temp.-L denote short and long temporal residuals, respectively.

Table 6: Comparison of generative and sequence classification. AUC and EER are averaged over the eight EnvSDD test subsets.

Model	Avg AUC ↑	Avg EER ↓
Qwen2.5-Omni-3B (`seq_cls`)	75.13	27.59
Qwen2.5-Omni-3B (Generative)	77.83	25.20
Qwen2.5-Omni-7B (`seq_cls`)	76.81	27.10
Qwen2.5-Omni-7B (Generative)	79.45	23.98

A.6.5 Generative vs. Sequence Classification.

To evaluate how the prediction head affects ESDD performance, we compare generative and discriminative classification for ALLMs. In the generative configuration, the model generates class labels as text tokens using its language-model head. In the discriminative configuration, we use the `seq_cls` implementation in `ms-swift`, which replaces this head with a linear binary classification head to predict class probabilities directly. Here, sequence classification retains the LLM backbone and changes only the prediction head, whereas the task-specific discriminative detectors in our main comparison perform detection without an LLM backbone. In both settings, the audio encoder is frozen and the LLM backbone is adapted using LoRA. The newly introduced classification head is trained from scratch. As shown in Table 6, the generative configuration performs better overall. We therefore use the generative configuration for ALLMs in our main comparisons and do not include their discriminative variants as additional baselines.

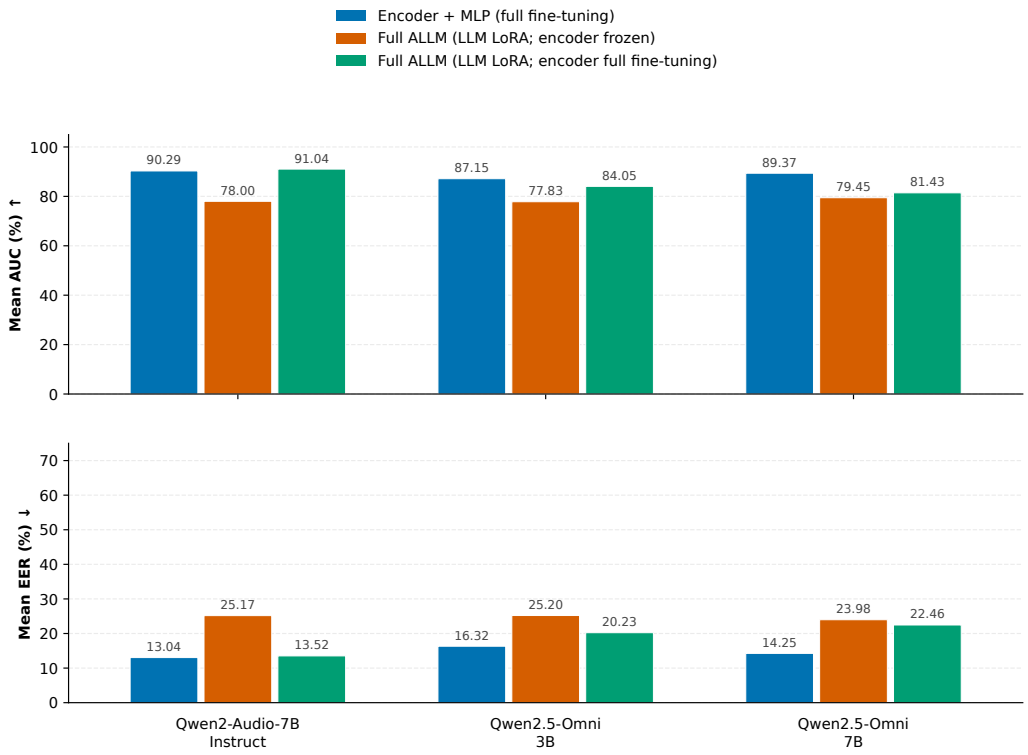


Figure 9: Comparison of three ALLMs under three configurations: Encoder + MLP with full fine-tuning, Full ALLM with LLM LoRA and a frozen audio encoder, and Full ALLM with LLM LoRA and full fine-tuning of the audio encoder. The panels report mean AUC (%, ↑) and EER (%, ↓) over the eight EnvSDD test subsets. Full ALLM results are from Table 2.

A.6.6 Local Relations across Generation Paradigms.

Figure 10 compares a real audio clip with its generated counterparts under two generation paradigms: audio-to-audio (ATA) generation using AudioLDM1 and text-to-audio (TTA) generation using AudioGen, AudioLDM1, and AudioLDM2. The joint relation magnitude maps highlight differences in local spectral and temporal patterns between real and generated audio, with different patterns across the generation models. These examples extend the paired residual visualizations to both ATA and TTA generation, supporting the use of local relations as detection cues across different generation paradigms.

A.6.7 Zero-Shot Results by Subset.

Table 7 gives the per-subset zero-shot results of the 15 ALLMs on the eight test subsets of EnvSDD, of which Figure 3 only reports the average. The table shows that the near-chance behavior discussed in Section 4.2 is reflected in the overall average, with most models and subsets exhibiting similarly poor performance.

A.6.8 Backbone Generalization.

Table 10 evaluates whether the proposed SRAM design remains effective across different audio backbones, including Qwen3-Omni, Qwen2.5-Omni, Qwen2-Audio, and Gemma-3n, while keeping the SRAM modules unchanged. These encoders share a Transformer-based architecture, which reduces variation in the underlying architectural family and allows a more controlled assessment of backbone generalization. We also include a randomly initialized Qwen3-Omni audio encoder to explore the contribution of pretraining under an identical architecture.

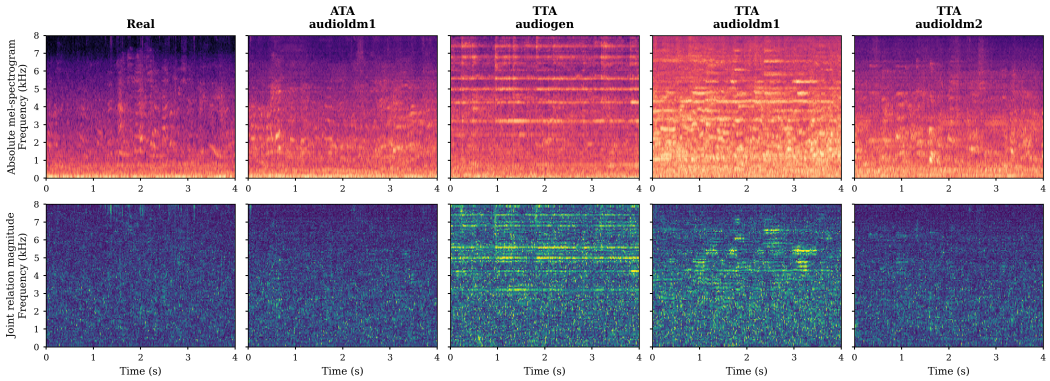


Figure 10: Mel-spectrograms (top) and joint relation magnitude maps (bottom) of a real audio clip and four generated counterparts under the ATA and TTA paradigms. Each row uses a shared color scale.

Table 7: Zero-shot evaluation of ALLMs on the EnvSDD test sets, in AUC (%) and EER (%).

Model	AT1		AT2		AT3		AT4		TT1		TT2		TT3		TT4		**Avg**	
	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓
Qwen2-Audio-7B	55.90	44.80	47.88	52.90	47.96	50.00	47.08	51.93	64.82	40.03	56.10	46.59	54.86	47.10	46.72	51.82	52.67	48.15
Qwen2-Audio-7B-Instruct	57.97	44.00	54.84	46.39	49.29	49.17	50.29	49.47	61.06	41.76	65.70	38.12	47.76	50.53	50.40	49.22	54.66	46.08
Qwen2.5-Omni-3B	48.60	51.46	51.59	48.97	49.16	50.21	53.51	47.62	61.37	42.04	77.28	28.93	59.67	43.89	67.56	36.55	58.59	43.71
Qwen2.5-Omni-7B	55.81	46.17	50.17	49.79	56.38	44.35	54.48	46.78	74.48	32.32	72.89	33.82	60.77	43.08	65.70	39.29	61.35	41.95
Qwen3-Omni-30B-A3B-Instruct	56.18	44.98	54.52	46.84	55.91	46.77	53.71	46.80	49.92	50.68	62.83	41.13	58.49	43.57	60.48	42.90	56.51	45.46
Qwen3-Omni-30B-A3B-Thinking	57.51	45.43	56.50	45.18	54.17	47.23	54.69	46.27	51.55	49.17	64.45	40.13	55.13	47.18	56.29	46.06	56.29	45.83
InternLM-XComposer-2.5-OL	47.22	51.43	50.00	50.00	47.83	51.11	50.00	50.00	47.45	51.31	49.69	50.16	50.00	50.00	50.00	50.00	49.02	50.50
gemma-3n-E4B	46.00	53.18	52.85	48.27	44.06	54.49	50.30	50.43	50.24	49.74	48.20	52.17	49.50	50.25	47.65	52.53	48.60	51.38
gemma-3n-E4B-it	47.29	52.17	47.55	51.73	57.10	44.80	46.27	52.55	43.09	54.72	51.80	47.63	55.68	44.40	52.12	47.95	50.11	49.49
gemma-3n-E2B	51.23	49.36	48.81	50.62	52.40	48.68	48.94	50.56	50.86	49.56	53.74	47.93	51.70	49.10	49.91	50.06	50.95	49.48
gemma-3n-E2B-it	48.37	50.02	54.82	46.69	45.64	52.95	54.86	46.32	43.84	53.10	49.40	47.80	41.91	55.23	43.34	54.45	47.77	50.82
Phi-4-multimodal-instruct	48.94	51.38	63.08	39.63	43.31	55.08	54.28	47.02	54.74	46.32	63.85	39.32	46.09	53.13	52.28	49.03	53.32	47.61
Step-Audio-2-mini	47.84	51.58	47.27	52.27	57.89	43.96	56.53	45.52	57.49	43.85	71.20	33.75	63.07	40.41	69.23	35.86	58.82	43.40
Step-Audio-2-mini-Base	56.09	44.94	44.54	53.60	59.92	42.84	53.74	47.32	63.03	40.54	71.22	34.32	65.05	38.26	66.94	37.81	60.07	42.45
Step-Audio-2-mini-Think	54.02	46.77	45.24	53.75	51.29	49.36	56.32	45.83	63.23	40.05	71.69	34.08	58.89	43.28	62.89	39.93	57.95	44.13

SRAM remains effective across the evaluated backbones, with pretrained Qwen3-Omni achieving the lowest mean EER. Its advantage may reflect the suitability of its pretraining data and specific encoder design for learning representations useful for audio forgery detection. These findings suggest that SRAM is applicable beyond a single backbone, while pretrained Qwen3-Omni provides the strongest foundation among the evaluated encoders.

A.6.9 Token retention ratio.

We vary the fraction ρ of tokens retained by sparse evidence pooling (Section 3.2). As shown in Table 8, $\rho = 0.25$ achieves the highest mean AUC (99.30%) and lowest mean EER (2.04%). Reducing the ratio to 0.10 raises the EER to 2.54%, whereas increasing it to 0.50 or 1.00 yields 3.12% or 2.56%, respectively. A small retained set may omit useful synthesis cues, while a larger set may include less informative tokens. The trend is not monotonic: dense pooling outperforms $\rho = 0.50$, but both remain worse than $\rho = 0.25$. These results favor retaining a moderate fraction of tokens among the tested settings.

A.6.10 Cross-layer trajectory refinement.

Table 9 explores the number of encoder layers used, the refinement steps, and direction normalization. Using two or three layers instead of four increases the mean EER from 2.04% to 2.57% and 2.50%, respectively, suggesting that sampling more encoder depths helps capture useful changes in the representations. Two refinement steps perform best among the tested settings: using one or four steps raises the mean EER to 2.56% and 2.63%, respectively. Thus, more refinement steps do not necessarily improve detection. Removing direction normalization increases the mean EER to 2.78%, the largest increase in this table. This supports normalizing inter-layer differences so that the trajectory code captures their directions without being affected by their magnitudes.

Table 8: Retention ratio ρ ablation on EnvSDD. AUC and EER (%) are averaged over the eight test subsets.

Retention ratio ρ	Avg AUC ↑	Avg EER ↓
0.10	99.05	02.54
0.25 (default)	**99.30**	**02.04**
0.50	99.01	03.12
1.00 (dense)	99.26	02.56

Table 9: Cross-layer trajectory refinement ablation on EnvSDD. The default configuration uses four encoder-layer taps and $R = 2$ refinement steps. AUC and EER (%) are averaged over the eight test subsets.

Variant	Avg AUC ↑	Avg EER ↓
4 taps, $R = 2$ (default)	99.30	02.04
2 taps	99.11	02.57
3 taps	99.18	02.50
$R = 1$	99.24	02.56
$R = 4$	99.16	02.63
w/o direction normalization	99.16	02.78

A.6.11 User Study.

To explore whether AI-generated environmental sounds can be distinguished from real recordings by human listeners, we conduct a small-scale perception study with 11 participants. All participants are graduate students, including six males and five females, and have passed a physical examination. Specifically, we use 80 audio clips from the eight EnvSDD test subsets, with five real and five generated clips from each subset. Every participant listens to the same clips and assigns a real or generated label to each, yielding 880 judgments in total. The study is conducted in an environment free from external sound interference, using a Mac computer and wired headphones. Confidence is recorded once at the end of the study: each participant selects low, medium, or high to express their overall confidence in their real/generated judgments across the listening task. Eight participants select low confidence, three select medium confidence, and none select high confidence. As shown in Table 11, overall accuracy is close to the chance level for this balanced task. Participants recognize real recordings more accurately than generated sounds and tend to judge the latter as real. This difficulty is most apparent on TT4, where generated sounds are the least frequently identified correctly. Table 12 reports the accuracy of every participant using anonymous identifiers. As shown in Figure 11, most participants report low confidence, and the medium-confidence group performs worse on average than the low-confidence group. Overall, these observations highlight the difficulty of distinguishing real from AI-generated environmental sounds by listening alone, underscoring the need for reliable ESDD methods.

A.6.12 Training epoch selection.

In this experiment, LoRA is applied only to the LLM component. Using Qwen2.5-Omni-3B and Qwen2.5-Omni-7B as controls, we find that performance does not increase monotonically with the epoch count and begins to degrade after roughly 8 epochs, as shown in Figure 12. We therefore cap all ALLM runs at 10 epochs and report the best checkpoint within that budget.

A.6.13 Performance with multilingual prompts.

We explore whether prompt language affects ALLM performance on ESDD. We choose English and Chinese (Mandarin) because they have the largest total numbers of speakers worldwide, including both native and non-native speakers (Gurevich et al., 2025). We train the models using prompts in each language. The Chinese prompts are translated from English using DeepL (Team, 2026), with special markers (e.g., `<audio>`) kept unchanged. As shown in Figure 13, prompt language has little effect on the performance of most models. However, switching between English and Chinese prompts changes the performance of gemma-3n-E2B by approximately 3%, suggesting that this

Table 10: AUC and EER (%) of different audio encoders on the eight EnvSDD test subsets and their average. 'Random' uses the Qwen3-Omni audio encoder architecture with randomly initialized weights.

Audio Encoder	AT1		AT2		AT3		AT4		TT1		TT2		TT3		TT4		**Avg**	
	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓	AUC ↑	EER ↓
Qwen3-Omni	100.00	00.00	100.00	00.03	99.48	01.55	99.45	01.40	99.99	00.03	99.20	02.81	99.59	01.40	96.71	09.10	99.30	02.04
Random	100.00	00.08	100.00	00.13	99.14	03.00	99.34	02.20	100.00	00.13	99.34	02.04	99.07	02.50	94.86	12.80	98.97	02.86
Qwen2.5-Omni	99.98	00.10	99.97	00.20	99.11	03.50	99.22	03.05	99.97	00.38	99.85	01.81	98.85	04.00	94.29	12.28	98.91	03.16
Gemma-3n	100.00	0.03	99.95	00.10	99.41	01.80	99.28	01.85	100.00	00.07	99.46	02.59	99.31	02.05	93.96	13.40	98.92	02.74
Qwen2-Audio	99.98	00.15	99.97	00.21	98.79	04.35	98.80	04.20	99.97	00.28	99.77	02.56	99.06	02.68	94.94	11.50	98.91	03.24

Table 11: Human detection performance across the eight EnvSDD test subsets, in accuracy (%) and class recall (%). Real and generated recall indicate the proportions of correctly classified judgments for each class.

Subset	Accuracy (%)	Real recall (%)	Generated recall (%)
AT1	49.1	54.5	43.6
AT2	58.2	67.3	49.1
AT3	52.7	49.1	56.4
AT4	49.1	58.2	40.0
TT1	52.7	65.5	40.0
TT2	48.2	60.0	36.4
TT3	54.5	58.2	50.9
TT4	40.0	54.5	25.5
Overall	50.6	58.4	42.7

model is more sensitive to prompt language. One possible explanation is that its training data may be mainly in English, making it harder for the model to connect Chinese instructions with the audio.

A.6.14 Performance gain after instruction tuning of ALLMs.

To investigate whether instruction tuning enhances the ESDD performance of ALLMs, we compare three non-instruction-tuned models—Qwen2-Audio-7B, Gemma-3n-E2B, and Gemma-3n-E4B—with their corresponding instruction-tuned variants. In this experiment, we apply LoRA fine-tuning only to the LLM component of each ALLM. As shown in Figure 14, instruction-tuned models consistently achieve higher AUC and lower EER. The largest improvement is observed for Gemma-3n-E2B-it. These gains may arise because instruction tuning helps models better understand and follow the task instructions for ESDD.

A.6.15 Model Efficiency.

Table 13 reports total parameter counts and per-sample inference latency for both tracks. For Qwen3-Omni-30B-A3B-Instruct, the language-model backbone contains approximately 30B parameters, but its mixture-of-experts architecture activates only approximately 3B parameters per token during inference (Team, 2025b). The models are measured with their trained checkpoints under `bfloat16` on a single NVIDIA RTX PRO 6000 with batch size 1, where latency is the mean of 100 forward passes after 20 warm-up runs and covers only the model's forward pass. Although SRAM is not the smallest model in terms of parameter count, its deployment cost remains modest, while it offers faster inference. Overall, SRAM provides a favorable trade-off between model size and inference efficiency. At 660.11M parameters it is comparable to DSSL at 629.89M, and its 10.53 ms latency is lower than both DSSL at 12.43 ms and BEATs+AASIST at 34.69 ms, enabling real-time detection in practical deployment scenarios. The contrast with the ALLMs is even more pronounced: SRAM is substantially smaller than every ALLM evaluated, while none of the ALLMs achieves a lower EER than SRAM. AASIST remains far smaller at 0.30M parameters, but its 29.10% mean EER indicates substantially weaker detection performance. These results highlight the potential of SRAM to provide real-time feedback in practical audio deepfake detection applications.

Table 12: Individual participant accuracy and overall self-reported confidence. Participants are identified anonymously. Each participant completes the same 80 judgments and reports confidence once after completing the study.

Participant	Confidence	Accuracy (%)
P01	Low	60.0
P02	Medium	40.0
P03	Low	52.5
P04	Low	50.0
P05	Medium	52.5
P06	Medium	35.0
P07	Low	65.0
P08	Low	50.0
P09	Low	45.0
P10	Low	58.8
P11	Low	47.5

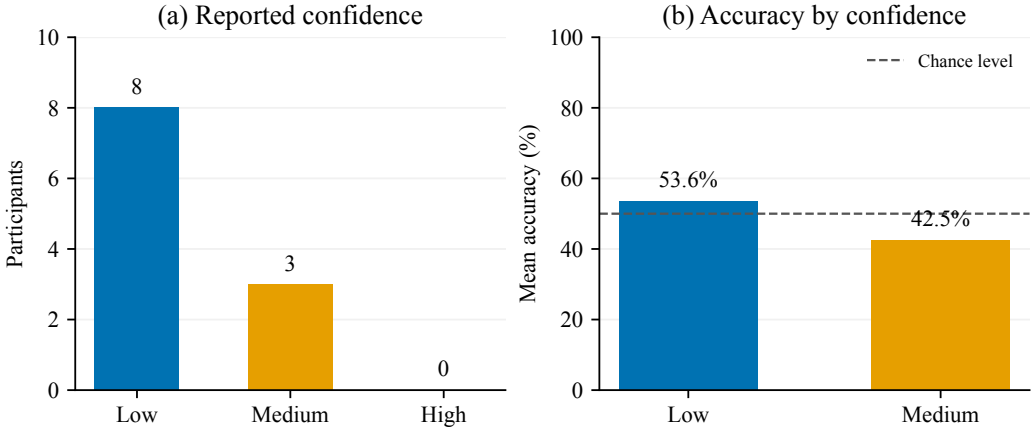


Figure 11: Overall confidence reported once after completing the listening task: (a) participant counts for the low, medium, and high confidence levels; (b) mean accuracy for the low- and medium-confidence groups. No participant selects high confidence. The dashed line shows the 50% accuracy expected from random guessing.

A.7 Future Work

Although individual residual maps provide useful cues, their benefits partly overlap. Our current design already provides input-adaptive control through a learned gate that regulates the injection of jointly encoded spectral features into the audio representations. However, it does not explicitly select

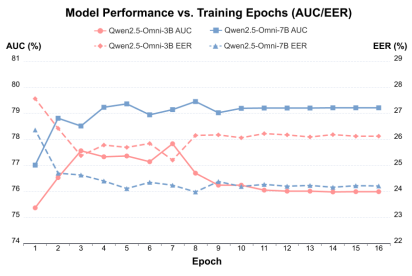


Figure 12: Analysis of the effect of the number of training epochs on model performance, in AUC (%) and EER (%).

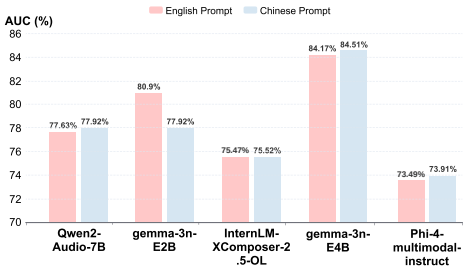


Figure 13: Analysis of the SFT performance of ALLMs across multilingual prompts, in AUC (%).

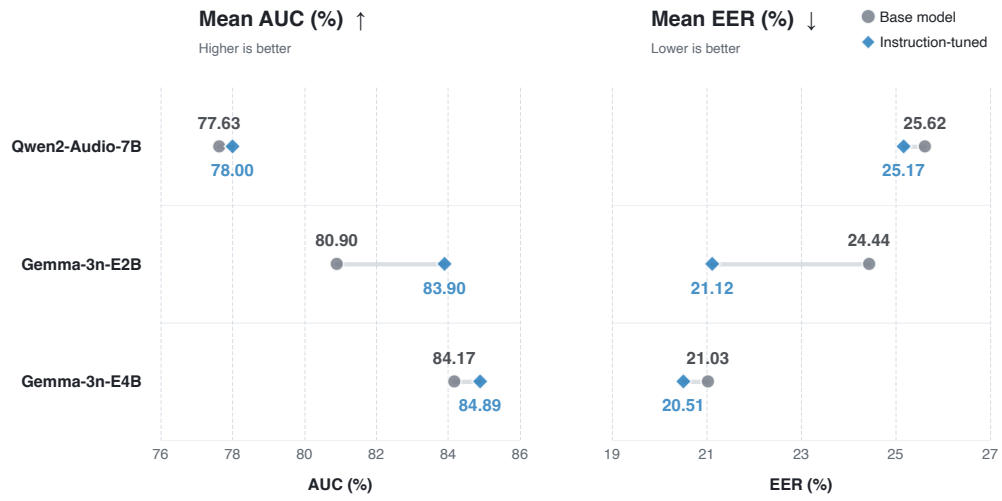


Figure 14: Analysis of the effect of instruction tuning on post-SFT performance, in AUC (%) and EER (%).

Table 13: Total parameter counts and per-sample inference latency of generative and discriminative models. Parameter counts are reported in billions (B) for generative models and millions (M) for discriminative models (1B = 1000M). Our method is highlighted in bold.

Model	Params	Latency (ms)
Generative models		
Qwen2-Audio-7B	8.40B	44.85
Qwen2-Audio-7B-Instruct	8.40B	45.37
Qwen2.5-Omni-3B	5.54B	19.33
Qwen2.5-Omni-7B	10.73B	27.92
Qwen3-Omni-30B-A3B-Instruct	35.26B	119.72
Qwen3-Omni-30B-A3B-Thinking	31.72B	115.64
gemma-3n-E4B	7.82B	47.62
gemma-3n-E4B-it	7.82B	48.31
gemma-3n-E2B	5.42B	42.06
gemma-3n-E2B-it	5.42B	42.24
Phi-4-multimodal-instruct	5.57B	38.67
internlm-xcomposer2d5-ol-7b	10.90B	39.74
Step-Audio-2-mini	8.36B	32.64
Step-Audio-2-mini-Think	8.36B	33.08
Step-Audio-2-mini-Base	8.36B	33.25
Discriminative models		
AASIST	0.30M	2.93
BEATs+AASIST	90.73M	34.69
W2V2+AASIST	315.88M	8.09
RAIS	315.92M	11.37
DSSL	629.89M	12.43
SRAM (ours)	**660.11M**	**10.53**

individual residual maps before encoding. We have explored such early selection, but preliminary experiments have not shown consistent improvements over the current design. We plan to investigate mixture-of-experts architectures to encourage specialization in different residual cues and enable more effective input-dependent selection.

A.8 Limitations

Room for improvement on the hardest subset. Although SRAM achieves the best performance on most subsets of EnvSDD, there is still room for further improvement under the most challenging condition. On TT4, where neither the source dataset nor the generator is seen during training, SRAM reaches 96.71% AUC and 9.10% EER, the best result reported on this subset so far. Even with this state-of-the-art performance, however, its cross-domain generalization can still be pushed further. How to generalize more robustly when both the data domain and the generator are unseen therefore remains a question worth studying, and we will explore it further in future work.

No in-depth analysis of why the fine-tuned ALLMs fall behind. Our experiments show that a discriminative classifier built on the audio encoder of an ALLM performs comparably to or better than the corresponding fine-tuned full ALLM, but we do not analyze its cause in depth. Analyzing the specific causes of these differences is beyond the scope of this paper and is left for future work.