AUDIOGAR: BRIDGING RECONSTRUCTION AND GENERATION IN LATENT AUDIO GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent audio generative models are typically trained in two stages: an audio codec is learned first, followed by a latent generative model. This decomposition leads to a decoder train-generation mismatch: the codec decoder is trained on encoder-induced latents but deployed on generator-produced latents at inference time. Across diverse datasets and latent generative models, we observe clear reconstruction-generation gaps under both FD and FAD, showing that strong reconstruction quality does not necessarily translate into strong end-to-end generation quality. A natural remedy is to adapt the decoder on generation-produced latents, but generated latents lack correspondence with source audio and therefore cannot directly provide the paired supervision used for decoder fine-tuning. We introduce **AudioGAR**, which constructs intermediate latents by perturbing encoder latents and denoising them through the frozen latent diffusion model. These latents form a trajectory from reconstruction toward generation, with lower-noise latents retaining source correspondence and supporting paired decoder fine-tuning. We fine-tune only the codec decoder on these latents, while keeping the codec encoder and latent generative model frozen. When applied to AudioX, AudioGAR substantially improves generative performance. It requires only 1.5% of the original training audio hours and 0.26% of the original training cost.

1 INTRODUCTION

Recent audio and music generation models span autoregressive, masked-token, and diffusion-based approaches (Kreuk et al., 2023; Copet et al., 2023; García et al., 2023; Ziv et al., 2024; Liu et al., 2023; 2024). Although these models generate audio differently, they are typically trained in two stages. An audio codec first learns to compress waveforms into a lower-dimensional latent space and reconstruct them through its decoder. A latent generative model is then trained to generate within this learned latent space. This decomposition avoids the computational burden of generating directly in raw waveform space, and has been adopted by a wide range of audio and music generation systems (Liu et al., 2023; 2024; Ghosal et al., 2023; Evans et al., 2025; Tian et al., 2026).

This two-stage training exposes the decoder to different latent distributions during training and generation. During codec training the decoder is optimized primarily on encoder-induced latents from raw audio, whereas at inference it is deployed on latents produced by the latent generative model. If these two latent distributions differ, strong reconstruction performance on encoder-induced latents need not carry over to generator-produced latents, a phenomenon we refer to as the *decoder train-generation mismatch*. Recent work has begun to design audio codecs with downstream generative modeling in mind (Bralios et al., 2025; Seetharaman & Kumar, 2026; Parker et al., 2026), but the distribution shift at the decoder input has received little attention. The analogous mismatch has recently been studied in visual latent generative models (Fang et al., 2026), but its presence and impact in latent audio generation have not been systematically examined.

We find clear evidence of this mismatch in latent audio generative models. Encoder-induced and generator-produced latents remain distinguishable in latent space on representative models and datasets (see Section 2.2). Across a broader set of models and datasets, reconstruction and generation also exhibit substantial gaps under both Fréchet Distance (FD) and Fréchet Audio Distance (FAD) (see Table 2 in Section 2.3). These results show that the mismatch is consequential for generation performance, motivating a dedicated approach to bridge this gap.

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.

Table 1: **AudioGAR improves AudioX with less training cost and data.** Relative cost and data are computed w.r.t. the original AudioX training. Training cost is reported as single-H100-equivalent wall-clock hours.

Methods	Training Cost		Training Data		AudioCaps		MusicCaps	
	Wall-Clock (Hours)	Rel. Cost	Audio Hours	Rel. Data	FD ↓	FAD ↓	FD ↓	FAD ↓
AudioX	4000	–	33052.3	–	11.83	1.61	9.56	1.60
AudioX + AudioGAR	**10.3**	**0.26%**	**508.3**	**1.5%**	**11.62**	**1.34**	**8.33**	**1.10**

A natural way to address the decoder train-generation mismatch is to fine-tune the decoder on generation-relevant latents. However, fully generated latents lack correspondence with any source audio and therefore cannot directly provide the paired reconstruction supervision used for decoder fine-tuning. This raises a practical question:

Can we obtain generation-relevant latents that remain paired with the source audio for decoder fine-tuning?

We introduce **AudioGAR** for latent audio diffusion models. AudioGAR constructs intermediate latents between reconstruction and generation by perturbing encoder-induced latents with noise and denoising them through the frozen latent diffusion model. Increasing the noise level makes these latents more generation-relevant but gradually weakens their correspondence with the source audio. At lower noise levels, they retain correspondence with the source audio while already incorporating information from the generative process. We fine-tune only the codec decoder on these latents, keeping the encoder and latent generative model frozen. We term this fine-tuning strategy *generation-aware codec decoder adaptation*.

We evaluate AudioGAR on AudioX (Tian et al., 2026) across both audio and music generation tasks. On AudioCaps (Kim et al., 2019), AudioGAR reduces FAD by 16.8% and FD by 1.8%. On MusicCaps (Agostinelli et al., 2023), the gains are more pronounced, with FD reduced by 12.9% and FAD by 31.3%. These improvements are achieved by fine-tuning only the codec decoder, using just 1.5% of the audio hours and 0.26% of the training cost of the original AudioX training (see Table 1). AudioGAR also generalizes to TangoMusic (Kong et al., 2024a), reducing FD by 6.3% and FAD by 20.5% on music generation (see Table 3).

Contributions. Our contributions are summarized as follows:

1. We identify the **decoder train-generation mismatch** in latent audio generative models, showing that encoder-induced and generator-produced latents are distinguishable and that reconstruction and generation exhibit substantial performance gaps in music generation task.
2. We introduce **AudioGAR**, which constructs intermediate latents between reconstruction and generation distribution, with lower-noise latents retaining correspondence with the source audio.
3. We use AudioGAR latents for **generation-aware codec decoder adaptation**, fine-tuning only the codec decoder while keeping the encoder and latent generative model frozen. AudioGAR consistently improves FD and FAD across models and datasets while requiring only a small fraction of the original training audio and computational cost.

2 Reconstruction and Generation in Latent Audio Models

Latent audio generative models share a single codec decoder across two distinct pathways, reconstruction and generation, yet supply it with encoder-induced and generator-produced latents respectively. We examine this asymmetry from the perspective of the decoder's latent inputs by characterizing the two distributions, verifying their divergence empirically, and measuring the corresponding reconstruction-generation performance gap.

2.1 Latent Distributions in Reconstruction and Generation

Reconstruction and generation share the same codec decoder but use input latents from different sources. We formalize these two pathways and describe the corresponding evaluation metrics.

Reconstruction. Given a waveform $x \sim p_{\text{data}}$, the codec encoder $\mathcal{E}_\theta$ produces a latent $z_e = \mathcal{E}_\theta(x)$, which the decoder $\mathcal{D}_\phi$ maps back to a waveform $\hat{x} = \mathcal{D}_\phi(z_e)$. We denote the encoder-induced latent distribution by $\mathcal{P}_e$. During codec training, the decoder is optimized to reconstruct source audio from encoder-induced latents.

Generation. Given conditioning information c, the latent generative model $\mathcal{G}_\varphi$ produces a latent $z_g \sim p_\varphi(z|c)$, which is decoded by the same codec decoder as $\tilde{x} = \mathcal{D}_\phi(z_g)$. We denote the resulting generator-produced latent distribution by $\mathcal{P}_g$. The decoder is thus optimized on encoder-induced latents but deployed on generator-produced latents during generation. When $\mathcal{P}_e \neq \mathcal{P}_g$, its input distribution differs between these two pathways. We refer to this distribution shift as the **decoder train-generation mismatch**.

Evaluation metrics. We evaluate reconstructed and generated audio using Fréchet Distance (FD) with PANNs embeddings (Kong et al., 2020b) and Fréchet Audio Distance (FAD; Kilgour et al., 2019) with VGGish embeddings (Hershey et al., 2017). We use **rFD** and **rFAD** for reconstructed audio $\hat{x}$, and **gFD** and **gFAD** for generated audio $\tilde{x}$. Reconstruction metrics evaluate the codec's reconstruction pathway, with the decoder operating on encoder-induced latents $z_e \sim \mathcal{P}_e$. Generation metrics evaluate the full generative system, with the same decoder operating on generator-produced latents $z_g \sim \mathcal{P}_g$.

2.2 Empirical Evidence of Distribution Mismatch

To determine whether $\mathcal{P}_e$ and $\mathcal{P}_g$ differ in practice, we examine these two latent distributions in AudioX (Tian et al., 2026) on AudioCaps (Kim et al., 2019), MusicCaps (Agostinelli et al., 2023), and VGGSound-Omni (Dai et al., 2026). For each dataset, we obtain encoder-induced latents z_e by encoding the real waveforms and generate z_g using the corresponding text captions, so that the two latent collections are constructed from matched audio-caption examples. We then apply Linear Discriminant Analysis (LDA) to identify the direction that maximally separates z_e and z_g, and visualize their one-dimensional projections using Gaussian-smoothed densities.

As shown in Figure 1, encoder-induced and generator-produced latents are distinguishable in AudioX across all three datasets. Latent-space separation is strongest on AudioCaps, with an LDA AUC of 0.962, compared with 0.862 on MusicCaps and 0.845 on VGGSound-Omni. This distributional difference remains visible after the latents enter the decoder. Clear separation persists in the Conv In, Block 2, and Block 4 feature spaces, with particularly clear separation at Conv In and Block 2 for all three datasets. (see Appendix B for feature extraction details). Figure 8 in Appendix C shows the same qualitative pattern for Stable Audio Open (Evans et al., 2025) across all three datasets, with even clearer separation in the latent space. These results provide empirical evidence of a distribution mismatch, i.e., $\mathcal{P}_e \neq \mathcal{P}_g$, that occurs across the evaluated models and datasets and persists in intermediate decoder representations. One possible explanation is that the learned latent generator $\mathcal{G}_\varphi$ cannot perfectly reproduce the encoder-induced latent distribution, resulting in the observed shift between encoder-induced and generator-produced latents.

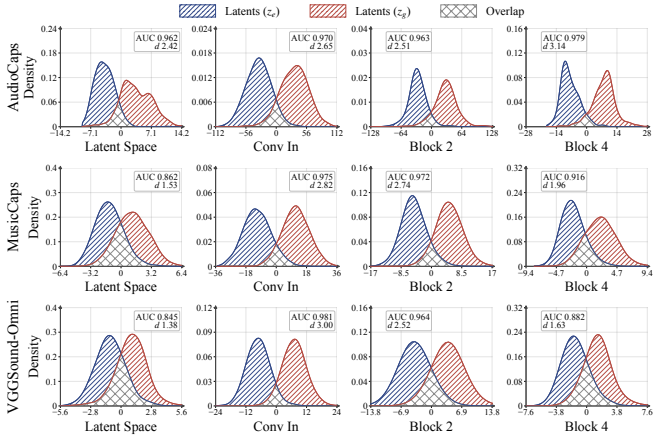


Figure 1: LDA projections of z_e and z_g in AudioX. Rows correspond to AudioCaps, MusicCaps, and VGGSound-Omni datasets; columns show latent space and decoder features at Conv In, Block 2, and Block 4. Curves show Gaussian-smoothed, normalized histogram densities of the one-dimensional LDA scores.

2.3 Reconstruction-Generation Performance Gap

We next examine whether the latent distribution mismatch identified in Section 2.2 translates into a measurable performance gap at the decoder output. Table 2 compares distribution-level reconstruction performance, measured by rFD and rFAD, with distribution-level generation performance, measured by gFD and gFAD, across 10 latent generative models on AudioCaps (Kim et al., 2019), 8 on MusicCaps (Agostinelli et al., 2023), and 11 on VGGSound-Omni (Dai et al., 2026). In every

Table 2: **Reconstruction-generation performance gap across latent audio generative models.** We report rFD and rFAD for reconstructed audio and gFD and gFAD for generated audio across multiple models on AudioCaps, MusicCaps, and VGGSound-Omni datasets. Reconstruction metrics are generally substantially lower than their generation counterparts. Lower values are better for all metrics.

Dataset	Models	rFD (↓)	gFD (↓)	rFAD (↓)	gFAD (↓)
AudioCaps	AudioLDM-2-Base (Liu et al., 2024)	3.19	11.54	1.31	2.04
	AudioLDM-2-Large (Liu et al., 2024)	3.20	11.53	1.31	1.86
	AudioLDM-S-Full (Liu et al., 2023)	3.05	21.85	1.27	4.79
	AudioX (Tian et al., 2026)	7.53	11.83	3.49	1.61
	Stable Audio Open (Evans et al., 2025)	7.53	30.36	3.49	3.41
	Tango (Ghosal et al., 2023)	2.97	14.47	1.14	1.52
	Tango Full (Ghosal et al., 2023)	2.97	22.23	1.14	4.10
	Tango Full (FT AudioCaps + MusicCaps) (Ghosal et al., 2023)	2.97	9.77	1.14	2.59
	Tango Full (FT AudioCaps) (Ghosal et al., 2023)	2.97	10.42	1.14	2.39
	Tango (AF-AC, FT AudioCaps) (Kong et al., 2024a)	2.97	12.08	1.14	2.55
MusicCaps	AudioLDM-2-Large (Liu et al., 2024)	2.20	18.84	0.66	2.51
	AudioLDM-2-Base (Liu et al., 2024)	2.20	21.71	0.66	3.82
	AudioX (Tian et al., 2026)	4.31	9.56	0.95	1.60
	Stable Audio Open (Evans et al., 2025)	4.31	37.72	0.95	3.42
	Tango (Ghosal et al., 2023)	2.18	46.80	0.56	5.64
	Tango Full (Ghosal et al., 2023)	2.18	35.67	0.56	5.78
	TangoMusic (Kong et al., 2024a)	2.04	15.20	0.42	1.85
	Tango Full (FT AudioCaps + MusicCaps) (Ghosal et al., 2023)	2.18	20.43	0.56	3.61
VGGSound-Omni	AudioLDM-2-Base (Liu et al., 2024)	1.64	10.03	0.49	1.72
	AudioLDM-2-Large (Liu et al., 2024)	1.64	9.93	0.49	1.49
	AudioLDM-S-Full (Liu et al., 2023)	1.63	18.78	0.50	3.44
	AudioX (Tian et al., 2026)	3.66	9.67	1.91	1.74
	Stable Audio Open (Evans et al., 2025)	3.66	26.07	1.91	2.68
	Tango (Ghosal et al., 2023)	1.62	24.03	0.50	2.88
	Tango Full (Ghosal et al., 2023)	1.62	19.01	0.50	3.69
	TangoMusic (Kong et al., 2024a)	1.62	13.43	0.50	2.36
	Tango (AF-AC, FT AudioCaps) (Kong et al., 2024a)	1.62	11.72	0.50	1.46
	Tango Full (FT AudioCaps + MusicCaps) (Ghosal et al., 2023)	1.62	13.26	0.50	2.28
	Tango Full (FT AudioCaps) (Ghosal et al., 2023)	1.62	19.12	0.50	3.06

case, rFD is substantially lower than gFD, and rFAD is also lower than gFAD in 26 of the 29 evaluated model-dataset combinations, indicating that strong decoder performance on encoder-induced latents does not reliably transfer when the same decoder receives generator-produced latents. The gap can be substantial: on MusicCaps, for example, Tango yields rFD = 2.18 against gFD = 46.80.

Figure 2 plots the empirical quantile curves of rFD and gFD across models on AudioCaps; the two curves remain separated throughout, with a median gap of 8.95. The same separation holds on MusicCaps and VGGSound-Omni (Figures 9a-9b, Appendix D), where the median gaps are larger still, reaching 18.88 and 11.81 respectively. This consistent separation across the full quantile range on all three datasets shows that the reconstruction-generation gap is a broad phenomenon across the evaluated models, rather than being driven by a few individual cases.

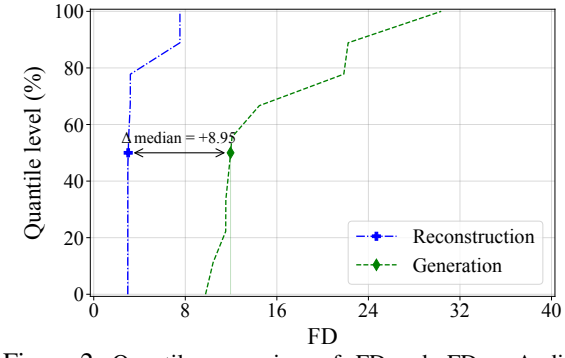


Figure 2: Quantile comparison of rFD and gFD on AudioCaps. The two curves remain separated throughout.

Reconstruction and generation are evaluated on the same datasets and under the same evaluation protocol, and both pathways use the same codec decoder. However, generation additionally depends on the latent generative model, so the observed reconstruction–generation gap does not by itself isolate the effect of the decoder input-distribution shift. Together with the clear separation between $\mathcal{P}_e$ and $\mathcal{P}_g$ observed in Section 2.2, these results motivate a direct empirical question: can adapting the decoder to generation-relevant latents improve end-to-end generation performance? We study this question in the next section.

3 Generation-Aware Codec Decoder Adaptation

Standard neural audio codecs are trained to decode encoder-induced latents back to the source audio, with reconstruction-based supervision (Zeghidour et al., 2021; Défossez et al., 2022; Kumar et al.,

2023; Evans et al., 2025). This supervision is naturally available for encoder-induced latents but not for generator-produced latents, and naive adaptation on either comes with a fundamental drawback.

Adapting on encoder-induced latents $z_e \sim \mathcal{P}_e$ preserves correspondence with the source audio, but continues to train the decoder on the same latent distribution seen during codec training and therefore does not directly address the decoder train-generation mismatch. Adapting on generator-produced latents $z_g \sim \mathcal{P}_g$, in contrast, exposes the decoder to its generation-time inputs, but without a paired source waveform, reconstruction supervision is unavailable.

To resolve this, we introduce **AudioGAR**, which constructs intermediate latents between reconstruction and generation. These latents become progressively more generation-relevant while retaining correspondence with the source audio at lower noise levels, making them suitable for decoder adaptation. We first describe how AudioGAR latents are constructed, then show how they bridge $\mathcal{P}_e$ and $\mathcal{P}_g$, and finally present generation-aware codec decoder adaptation.

3.1 AudioGAR Latent Construction

AudioGAR constructs intermediate latents by perturbing an encoder-induced latent and then denoising it with the frozen latent diffusion model, as illustrated in Figure 3. Given a source waveform x and its encoder-induced latent $z_e = \mathcal{E}_\theta(x)$, we first add Gaussian noise according to the diffusion schedule, yielding $z^t_{\text{noisy}} = a_t z_e + b_t \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ and $t \in [0, 1]$ denotes the normalized diffusion time. We characterize the perturbation strength by the relative noise level $\eta_t = b_t/(a_t + b_t)$. For the cosine parameterization used in our experiments, $a_t = \cos(\pi t/2)$ and $b_t = \sin(\pi t/2)$, so η_t increases monotonically from 0 to 1 as t goes from 0 to 1.

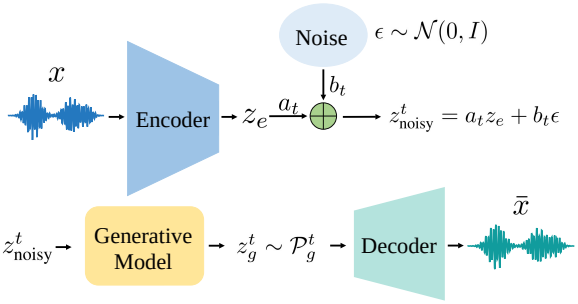


Figure 3: Illustration of the AudioGAR process.

Starting from z^t_{noisy}, the frozen latent diffusion model $\mathcal{G}_\varphi$ denoises the latent from diffusion time t toward the clean latent space using $\lfloor N_{\max} t \rceil$ denoising steps. We denote the resulting latent by $z^t_g = \mathcal{G}_\varphi(z^t_{\text{noisy}}, t, c) \sim \mathcal{P}^t_g$, where c is the conditioning information associated with the source audio. The codec encoder and latent diffusion model remain frozen throughout this construction.

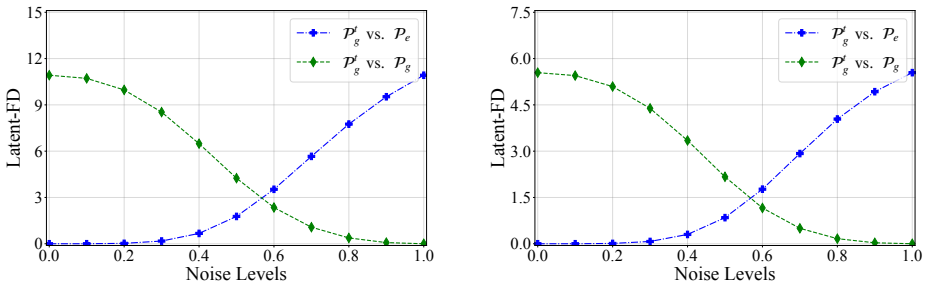
The noise level η_t controls the position between reconstruction and generation. At $t = 0$, we have $\eta_t = 0$, so no perturbation or denoising is applied and $z^t_g = z_e$, recovering the reconstruction endpoint. At $t = 1$, we have $\eta_t = 1$, so the encoder latent is fully replaced by Gaussian noise and the full $N_{\max}$-step generative process is applied, recovering the generation endpoint. Intermediate values of t therefore produce a family of latents between these two endpoints. Because these latents originate from a particular encoder latent z_e, lower-noise AudioGAR latents retain correspondence with the source waveform while incorporating information from the generative process.

3.2 Bridging Reconstruction and Generation

We next examine whether these intermediate latents indeed trace a trajectory between the encoder-induced and generator-produced latent distributions. Using AudioX (Tian et al., 2026) on AudioCaps (Kim et al., 2019) and MusicCaps (Agostinelli et al., 2023), we extract encoder-induced latents $z_e \sim \mathcal{P}_e$, generator-produced latents $z_g \sim \mathcal{P}_g$, and AudioGAR latents $z^t_g \sim \mathcal{P}^t_g$ across different noise levels, and compare each intermediate distribution $\mathcal{P}^t_g$ with both endpoints.

Directly characterizing the discrepancy between high-dimensional latent distributions is difficult, so we use Latent Fréchet Distance (Latent-FD) as a tractable approximation. Latent-FD computes the Fréchet distance from the empirical means and covariances of two latent sets under a Gaussian approximation, following the same general principle as widely used Fréchet metrics for image and audio generation. We provide the full definition in Appendix A.

Figures 4a and 4b show the resulting distances on AudioCaps and MusicCaps. As the noise level increases, the Latent-FD between $\mathcal{P}^t_g$ and $\mathcal{P}_e$ increases monotonically, while the Latent-FD between



(a) AudioCaps (b) MusicCaps

Figure 4: **AudioGAR bridges encoder-induced and generator-produced latent distributions.** Latent-FD between the intermediate AudioGAR distribution $\mathcal{P}_g^t$ and the two endpoints, $\mathcal{P}_e$ and $\mathcal{P}_g$, across noise levels on (a) AudioCaps and (b) MusicCaps. As the noise level increases, $\mathcal{P}_g^t$ moves farther from $\mathcal{P}_e$ and closer to $\mathcal{P}_g$, tracing a trajectory from reconstruction toward generation.

$\mathcal{P}_g^t$ and $\mathcal{P}_g$ decreases monotonically, consistently across both datasets. AudioGAR thus provides a smooth distributional trajectory from reconstruction toward generation, rather than an abrupt transition, supporting its use for generation-aware decoder adaptation.

3.3 Decoder Adaptation with AudioGAR Latents

We fine-tune the pretrained codec decoder on AudioGAR latents, keeping the same decoder-side training objective as the original codec. Throughout adaptation, the codec encoder $\mathcal{E}_\theta$ and latent generative model $\mathcal{G}_\varphi$ are frozen, while the pretrained decoder $\mathcal{D}_\phi$ is updated. The key difference from standard codec training is the decoder input: instead of encoder-induced latents z_e, the decoder receives AudioGAR latents z_g^t, which retain correspondence with the source audio while shifting toward the generator-produced latent distribution.

For a source waveform x with AudioGAR latent z_g^t, the reconstructed waveform $\bar{x} = \mathcal{D}_\phi(z_g^t)$ is optimized under $\mathcal{L}_{\text{AudioGAR}} = \mathcal{L}_{\text{codec}}(x, \bar{x})$, where $\mathcal{L}_{\text{codec}}$ denotes the decoder-side reconstruction and adversarial objectives of the corresponding pretrained codec. For EnCodec-style codec training, these include time- and frequency-domain reconstruction losses, adversarial losses, and feature-matching losses (Défossez et al., 2022); for the AudioLDM VAE (Liu et al., 2023), mel-spectrogram reconstruction and PatchGAN-based adversarial supervision (Isola et al., 2017). All loss weighting hyperparameters are kept identical to those used in the original codec training. We refer to this decoder adaptation procedure using AudioGAR latents as **AudioGAR**.

AudioGAR relies on a generative process that can start from a partially perturbed encoder latent and progressively denoise it toward the generated latent distribution. It is therefore designed for latent diffusion and flow-matching models (Rombach et al., 2022; Peebles & Xie, 2023; Geng et al., 2026; Liu et al., 2023; Ghosal et al., 2023; Tian et al., 2026), and does not directly apply to autoregressive latent generators such as AudioGen (Kreuk et al., 2023) and MusicGen (Copet et al., 2023), or to other generators without an analogous intermediate denoising trajectory.

4 Experiments

We evaluate AudioGAR on pretrained latent audio generative models to study its effectiveness and behavior during codec decoder adaptation. We first describe the experimental setup and then report the main results. Finally, we present sensitivity analysis, ablation studies, and training dynamics. Code and checkpoints will be publicly released to ensure reproducibility.

4.1 Experimental Setup

Models and Training Setup. We primarily evaluate AudioGAR on AudioX (Tian et al., 2026) and additionally apply it to TangoMusic (Kong et al., 2024a). We initialize from released pretrained checkpoints and freeze the codec encoder and latent generative model in both cases. For AudioX, we fine-tune only the codec decoder; for TangoMusic, we jointly fine-tune the VAE decoder and HiFi-GAN vocoder. AudioGAR latents z_g^t are constructed with $\eta_t = 0.2$. Since the released checkpoints

Table 3: **Performance evaluation across tasks and datasets.** T2A and T2M denote Text-to-Audio and Text-to-Music, respectively. † denotes results reported in AudioX (Tian et al., 2026), while ‡ denotes results from our experiments, including reproduced baselines and AudioGAR evaluations. The best result for each metric within each task is shown in bold.

Dataset	Method	Task	KL (↓)	IS (↑)	FD (↓)	FAD (↓)	PC (↑)	PQ (↑)
AudioCaps	AudioGen†	T2A	1.39	10.22	13.29	1.72	3.26	5.25
	AudioLDM-L-Full†	T2A	2.00	6.51	37.27	8.37	2.82	5.67
	AudioLDM-S-Full‡	T2A	1.52	7.59	21.85	4.79	3.21	5.81
	Tango 2†	T2A	**1.11**	10.37	12.22	3.20	**3.63**	5.82
	Tango (AF-AC, FT AudioCaps)‡	T2A	1.24	10.66	12.08	2.55	3.44	6.13
	MAGNET-large†	T2A	1.62	7.46	24.88	2.99	3.25	5.15
	MMAudio†	T2A	1.35	12.03	12.63	4.71	3.06	5.64
	AudioLDM-2-Base‡	T2A	1.28	10.56	11.54	2.04	3.12	5.89
	AudioLDM-2-Large†	T2A	1.49	8.46	26.34	1.97	2.86	5.77
	AudioLDM-2-Large‡	T2A	1.22	10.15	11.53	1.86	3.14	5.80
	Stable Audio Open†	T2A	2.01	10.37	29.01	3.15	2.77	**6.16**
	Stable Audio Open‡	T2A	2.04	10.12	30.36	3.41	2.69	6.08
	AudioX†	T2A	1.27	**12.48**	**11.51**	1.59	3.32	5.80
	AudioX‡	T2A	1.31	12.47	11.83	1.61	3.16	5.73
	AudioX + AudioGAR (Ours)‡	T2A	1.29	12.06	11.62	**1.34**	3.11	5.73
MusicCaps	MusicGen†	T2M	1.43	2.24	25.40	4.55	5.19	7.16
	AudioLDM-L-Full†	T2M	1.45	2.49	34.44	6.34	4.72	6.10
	MAGNET-large†	T2M	1.32	1.98	23.88	4.24	5.84	6.71
	AudioLDM-2-Base‡	T2M	1.20	2.51	21.71	3.82	5.35	**7.39**
	AudioLDM-2-Large†	T2M	1.26	2.84	15.61	2.80	5.22	6.70
	AudioLDM-2-Large‡	T2M	1.17	2.70	18.84	2.51	5.38	7.23
	Stable Audio Open†	T2M	1.51	2.94	36.33	3.23	3.91	7.18
	Stable Audio Open‡	T2M	1.56	2.98	37.72	3.42	3.75	7.26
	Tango‡	T2M	2.03	2.05	46.80	5.64	**5.86**	6.64
	Tango Full (FT AudioCaps + MusicCaps)‡	T2M	0.98	2.71	20.43	3.61	5.71	6.29
	TangoMusic†	T2M	1.13	2.86	15.00	1.88	5.57	7.06
	TangoMusic‡	T2M	1.09	2.85	15.20	1.85	5.62	7.22
	TangoMusic + AudioGAR (Ours)‡	T2M	1.09	2.83	14.25	1.47	5.61	7.16
	AudioX†	T2M	**0.96**	3.55	9.76	1.53	5.21	6.70
	AudioX‡	T2M	1.00	**3.65**	9.56	1.60	4.78	6.61
	AudioX + AudioGAR (Ours)‡	T2M	0.99	3.59	**8.33**	**1.10**	4.75	6.56

do not include the discriminators used for codec training, we initialize the discriminator with random weights and train them jointly with the decoder-side components following the respective codec training procedures (Evans et al., 2025; Kong et al., 2020a). AudioX adaptation runs for 100k steps using 508.3 hours of audio. As shown in Table 1, this amounts to 1.5% of the original training audio hours and 0.26% of the original training cost. Full training details are provided in Appendix E.

Evaluation Metrics. Following AudioX (Tian et al., 2026), we evaluate overall audio generation performance using several established metrics: Kullback-Leibler Divergence (KL), Inception Score (IS), Fréchet Distance (FD) computed with PANNs embeddings (Kong et al., 2020b), Fréchet Audio Distance (FAD) computed with VGGish embeddings (Hershey et al., 2017), and Production Complexity (PC) and Production Quality (PQ) (Tjandra et al., 2025). Additional details on these metrics are provided in Appendix F.

Baselines. For the text-to-audio task on AudioCaps (Kim et al., 2019), we compare against AudioGen (Kreuk et al., 2023), AudioLDM-L-Full and AudioLDM-S-Full (Liu et al., 2023), Tango 2 (Majumder et al., 2024), Tango (AF-AC, FT AudioCaps) (Kong et al., 2024a), MAGNET-large (Ziv et al., 2024), MMAudio (Cheng et al., 2025), AudioLDM-2-Base and AudioLDM-2-Large (Liu et al., 2024), Stable Audio Open (Evans et al., 2025), and AudioX (Tian et al., 2026). For the text-to-music task on MusicCaps (Agostinelli et al., 2023), we compare against MusicGen (Copet et al., 2023), AudioLDM-L-Full (Liu et al., 2023), MAGNET-large (Ziv et al., 2024), AudioLDM-2-Base and AudioLDM-2-Large (Liu et al., 2024), Stable Audio Open (Evans et al., 2025), Tango and Tango Full (FT AudioCaps + MusicCaps) (Ghosal et al., 2023), TangoMusic (Kong et al., 2024a), and AudioX (Tian et al., 2026).

4.2 Main Results

AudioGAR on AudioX. Table 3 reports the main results on AudioCaps and MusicCaps. Compared with our reproduced AudioX baseline, AudioX + AudioGAR improves both FD and FAD on

Table 4: **Effect of latent choice during decoder adaptation.** We compare the pretrained AudioX baseline with decoder adaptation using either encoder-induced latents z_e or AudioGAR latents z_g^t on AudioCaps and MusicCaps. ✗ and ✔ indicate whether decoder adaptation is applied.

Dataset	Decoder Adaptation	Decoder Latents	KL (↓)	IS (↑)	FD (↓)	FAD (↓)	PC (↑)	PQ (↑)
AudioCaps	✗	-	1.31	**12.47**	11.83	1.61	**3.16**	**5.73**
	✔	z_e	**1.29**	12.19	11.72	1.55	3.14	5.70
	✔	z_g^t	**1.29**	12.06	**11.62**	**1.34**	3.11	**5.73**
MusicCaps	✗	-	1.00	**3.65**	9.56	1.60	**4.78**	**6.61**
	✔	z_e	**0.99**	3.62	8.87	1.38	4.76	6.50
	✔	z_g^t	**0.99**	3.59	**8.33**	**1.10**	4.75	6.56

the two datasets. On AudioCaps, FD decreases from 11.83 to 11.62 and FAD from 1.61 to 1.34, corresponding to relative reductions of 1.8% and 16.8%, respectively. On MusicCaps, the improvements are larger: FD drops from 9.56 to 8.33 and FAD from 1.60 to 1.10, reductions of 12.9% and 31.3%. Performance on KL, IS, PC, and PQ remains broadly matched with the pretrained AudioX baseline. Among the methods in Table 3, AudioX + AudioGAR achieves the best FAD on AudioCaps and the best FD and FAD on MusicCaps, with only marginal changes in the other evaluated metrics. Notably, decoder adaptation introduces no additional inference cost, as only the decoder weights are updated while the inference pipeline and generative model remain unchanged.

Generalization to TangoMusic. We further apply AudioGAR to TangoMusic (Kong et al., 2024a) to test whether the same adaptation strategy transfers to a different latent audio generative model and codec pipeline. Relative to our reproduced TangoMusic baseline, TangoMusic + AudioGAR reduces FD from 15.20 to 14.25 and FAD from 1.85 to 1.47 on MusicCaps, improvements of 6.3% and 20.5%, respectively. The remaining metrics stay closely matched, with KL unchanged and only minor differences in IS, PC, and PQ. These results confirm that the gains from AudioGAR are not specific to AudioX, and that generation-aware decoder adaptation transfers across different pretrained latent audio generative models and decoder architectures.

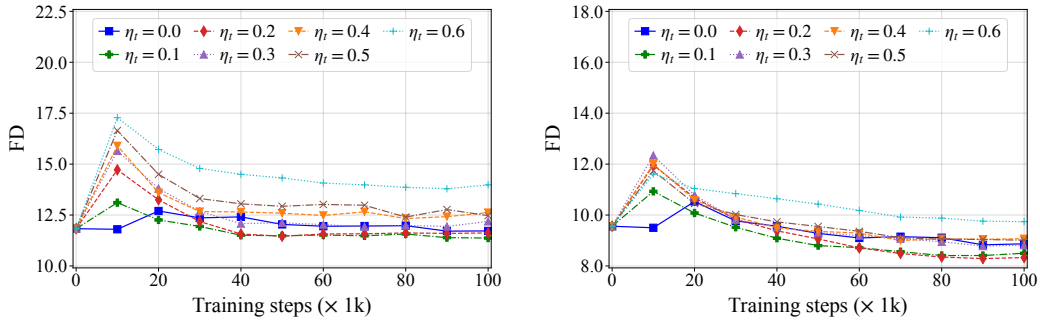
4.3 Ablation and Analysis

We further analyze AudioGAR from three perspectives: the role of AudioGAR latents in decoder adaptation, sensitivity to the noise level, and the training dynamics of generation performance.

Effect of AudioGAR Latents. We first isolate the effect of using AudioGAR latents during decoder adaptation. Table 4 compares the pretrained AudioX baseline with decoder adaptation using either encoder-induced latents z_e or AudioGAR latents z_g^t, while keeping the adaptation procedure otherwise unchanged. Adapting on z_e already provides modest improvements in FD and FAD, indicating that decoder fine-tuning itself can be beneficial. Replacing z_e with z_g^t, however, yields consistently larger gains on both datasets. On AudioCaps, FD/FAD improve from 11.72/1.55 with z_e to 11.62/1.34 with z_g^t; on MusicCaps, they improve from 8.87/1.38 to 8.33/1.10. This comparison isolates the benefit of exposing the decoder to generation-aware AudioGAR latents rather than simply continuing adaptation on the encoder-induced latent distribution.

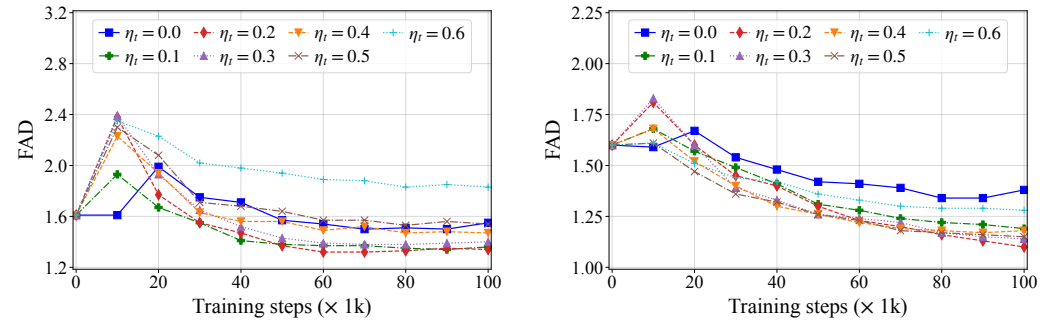
Sensitivity to the Noise Level. We next study how the AudioGAR noise level affects decoder adaptation. Figures 5 and 6 show FD and FAD throughout 100k adaptation steps for $\eta_t \in \{0.0, 0.1, \ldots, 0.6\}$ on AudioCaps and MusicCaps. Intermediate noise levels, particularly $\eta_t = 0.2$ and $\eta_t = 0.3$, generally achieve better generation performance than either endpoint of the evaluated range. When η_t is too small, such as $\eta_t = 0.0$, the decoder continues to receive latents close to the encoder-induced distribution and therefore does not sufficiently address the decoder train-generation mismatch. Increasing η_t makes the latents more generation-relevant, but excessively large noise levels, such as $\eta_t = 0.6$, weaken their correspondence with the source audio and make paired reconstruction supervision less informative. These results highlight the trade-off underlying AudioGAR: intermediate noise levels provide generation-relevant latents while retaining sufficient source correspondence for decoder adaptation.

Training Dynamics. Finally, we examine how generation performance evolves during AudioGAR adaptation. Figure 7 tracks FD and FAD on AudioCaps and MusicCaps from the pretrained



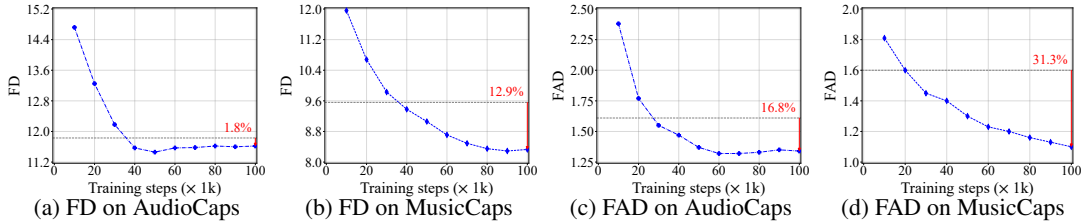
(a) AudioCaps (b) MusicCaps

Figure 5: **Sensitivity of FD to the AudioGAR noise level.** FD throughout decoder adaptation with AudioX for different noise levels η_t on (a) AudioCaps and (b) MusicCaps.



(a) AudioCaps (b) MusicCaps

Figure 6: **Sensitivity of FAD to the AudioGAR noise level.** FAD throughout decoder adaptation with AudioX for different noise levels η_t on (a) AudioCaps and (b) MusicCaps.



(a) FD on AudioCaps (b) FD on MusicCaps (c) FAD on AudioCaps (d) FAD on MusicCaps

Figure 7: **Training dynamics of AudioGAR on AudioX.** FD and FAD over 100k decoder-adaptation steps on AudioCaps and MusicCaps with $\eta_t = 0.2$.

AudioX checkpoint at 0k steps through 100k adaptation steps. All four metrics initially worsen before improving with continued training and eventually surpassing the pretrained baseline. A possible explanation for the early decline is the random initialization of the discriminator, which is trained jointly with the pretrained decoder and may require time to provide useful adversarial supervision. The curves therefore complement the final results in Section 4.2 by showing when the improvements emerge during adaptation.

5 Conclusion

We examined latent audio generation from the decoder's perspective and identified a train-generation mismatch between codec training and inference. This mismatch shows that strong reconstruction performance does not necessarily transfer to generation, making the decoder an important but underexplored target for improving latent audio generative models. Based on this observation, AudioGAR constructs intermediate latents between reconstruction and generation and uses them for generation-aware codec decoder adaptation, leaving the codec encoder and latent generative model unchanged. More broadly, our results suggest that improving two-stage generative systems does not always require modifying the generator itself: adapting the decoder to its generation-time inputs provides a complementary route that preserves the pretrained generative structure.

6 AI USE STATEMENT

Generative AI tools were used in this work solely for language refinement and editorial assistance. They did not contribute to the development of the research ideas, methodological design, theoretical derivations, experimental setup, data preparation, implementation, or interpretation of the scientific results. All AI-assisted text was carefully reviewed and edited by the authors, who remain fully responsible for the accuracy, integrity, and final presentation of the manuscript.

7 ETHICS STATEMENT

This research does not involve human-subject studies, user studies, crowdsourcing, participant interaction, or the collection of new personal or sensitive data. Therefore, Institutional Review Board (IRB) approval or an equivalent ethics review was not required.

Our experiments use publicly available audio datasets and pretrained model architectures, as described in the paper and supplementary material. We do not collect additional information about the audio recordings, attempt to identify individuals, or conduct analyses related to personal attributes.

This work does not introduce a new dataset or collect new human-subject data. AudioGAR adapts the codec decoder of existing latent audio generative models and does not change how training data are collected. Any ethical concerns related to the underlying datasets or generated audio are therefore inherited from the base models and data used in our experiments. All datasets and models are used for research purposes under their applicable terms and conditions.

8 REPRODUCIBILITY STATEMENT

To support reproducibility, we provide the implementation code used in our experiments in the supplementary material, and instructions for running the code and reproducing the main results.

We document the training data and preprocessing procedures, model-specific adaptation settings, optimization hyperparameters, and implementation details in Appendix E. Additional details on the evaluation metrics and protocols are provided in Appendix F. These details cover the settings needed to reproduce the reported experiments.

Upon publication, we will publicly release the complete AudioGAR implementation, configuration files, and the adapted model checkpoints produced in this work, together with execution instructions. We expect these resources to facilitate reproducibility and future research building on our results.

REFERENCES

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matthew Sharifi, Neil Zeghidour, and Christian Havnø Frank. MusicLM: Generating music from text. *ArXiv*, 2023.

Dimitrios Bralios, Jonah Casebeer, and Paris Smaragdis. Re-bottleneck: Latent re-structuring for neural audio autoencoders. In *MLSP*, 2025.

Ho Kei Cheng, Masato Ishii, Akio Hayakawa, Takashi Shibuya, Alexander Schwing, and Yuki Mitsufuji. MMAudio: Taming multimodal joint training for high-quality video-to-audio synthesis. In *CVPR*, 2025.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. In *NeurIPS*, 2023.

Yusheng Dai, Zehua Chen, Yuxuan Jiang, Qiuhong Ke, Jianfei Cai, and Jun Zhu. Omni2Sound: Towards unified video-text-to-audio generation. In *CVPR*, June 2026.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *ArXiv*, 2022.

Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. In *ICLR*, 2019.

Zach Evans, Julian Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *ICASSP*, 2025.

Xianghong Fang, Wenjie Shu, Tongda Xu, Wenlong Mou, Dehan Kong, and Tim G. J. Rudner. Bridging reconstruction and generation: A latent distribution perspective on evaluation and improvement. *ArXiv*, 2026.

Hugo Flores García, Prem Seetharaman, Rithesh Kumar, and Bryan Pardo. VampNet: Music generation via masked acoustic token modeling. In *ISMIR*, 2023.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. In *CVPR*, 2026.

Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria. Text-to-audio generation using instruction-tuned LLM and latent diffusion model. In *ACM Multimedia*, 2023.

Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin W. Wilson. CNN architectures for large-scale audio classification. In *ICASSP*, 2017.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash Equilibrium. In *NeurIPS*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In *Interspeech 2019*, pp. 2350–2354, 2019.

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

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

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020b.

Zhifeng Kong, Sang gil Lee, Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, Rafael Valle, Soujanya Poria, and Bryan Catanzaro. Improving text-to-audio models with synthetic captions. In *Synthetic Data's Transformative Role in Foundational Speech Models*, pp. 1–5, 2024a.

Zhifeng Kong, Arushi Goel, Rohan Badlani, Wei Ping, Rafael Valle, and Bryan Catanzaro. Audio flamingo: A novel audio language model with few-shot learning and dialogue abilities. In *ICML*, 2024b.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. AudioGen: Textually guided audio generation. In *ICLR*, 2023.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved RVQGAN. In *NeurIPS*, 2023.

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.

Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. AudioLDM 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

Navonil Majumder, Chia-Yu Hung, Deepanway Ghosal, Wei-Ning Hsu, Rada Mihalcea, and Soujanya Poria. Tango 2: Aligning diffusion-based text-to-audio generations through direct preference optimization. In *ACM Multimedia*, 2024.

Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D. Plumbley, Yuexian Zou, and Wenwu Wang. Wavcaps: A ChatGPT-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

Julian Parker, Zach Evans, CJ Carr, Zack Zukowski, Josiah Taylor, Matthew Rice, and Jordi Pons. SAME: A semantically-aligned music autoencoder. *ArXiv*, 2026.

William S. Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *NeurIPS*, 2016.

Prem Seetharaman and Rithesh Kumar. Taming audio VAEs via target-KL regularization. *ArXiv*, 2026.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Zeyue Tian, Zhaoyang Liu, Yizhu Jin, Ruibin Yuan, Liumeng Xue, Xu Tan, Qifeng Chen, Wei Xue, and Yike Guo. AudioX: A unified framework for anything-to-audio generation. In *ICLR*, 2026.

Andros Tjandra, Yi-Chiao Wu, Baishan Guo, John Hoffman, Brian Ellis, Apoorv Vyas, Bowen Shi, Sanyuan Chen, Matt Le, Nick Zacharov, Carleigh Wood, Ann Lee, and Wei-Ning Hsu. Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound. *ArXiv*, 2025.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP*, 2023.

Zeyu Xie, Xuenan Xu, Zhizheng Wu, and Mengyue Wu. AudioTime: A temporally-aligned audio-text benchmark dataset. In *ICASSP*, 2025.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

Alon Ziv, Itai Gat, Gaël Le Lan, Tal Remez, Felix Kreuk, Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. Masked audio generation using a single non-autoregressive transformer. In *ICLR*, 2024.

A Latent Fréchet Distance

To measure the distributional discrepancy in latent space, we adopt the Fréchet distance between two sets of latent variables, $z_i^a \sim \mathcal{P}_A$ and $z_i^b \sim \mathcal{P}_B$. Specifically, we estimate the empirical mean and covariance of the two latent distributions, denoted by (μ_1, Σ_1) and (μ_2, Σ_2), respectively. Based on these statistics, we define the latent Fréchet distance (Latent-FD) as

$$\mathrm{FD}(\mathcal{P}_A, \mathcal{P}_B) = \|\mu_1 - \mu_2\|_2^2 + \mathrm{Tr}\Big(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{1/2}\Big), \tag{1}$$

This metric provides a well-established measure of discrepancy between two distributions by capturing both first-order (mean) and second-order (covariance) statistics. It is closely related to the widely used Fréchet Inception Distance (FID) (Heusel et al., 2017) for image generation and Fréchet Distance (FD) and Fréchet Audio Distance (FAD) for audio generation. All of these metrics compute the Fréchet distance under a Gaussian assumption and differ primarily in the representation space used to estimate the corresponding statistics. Specifically, FID uses 2048-dimensional features extracted by a pretrained InceptionV3 (Szegedy et al., 2016) network, FD uses PANNs embeddings (Kong et al., 2020b), and FAD uses VGGish embeddings (Hershey et al., 2017). In contrast, our Latent-FD is computed directly in the model's latent space using latents obtained from the encoder or the generative process. This allows Latent-FD to quantify distributional discrepancies within the generative pipeline while retaining the same underlying geometric interpretation as FID, FD, and FAD.

B Decoder Feature Spaces for Latent Distribution Analysis

In Section 2.2, we analyze intermediate representations of AudioX (Tian et al., 2026) and Stable Audio Open (Evans et al., 2025), which use the same codec decoder (Evans et al., 2025). We extract features from three decoder stages: (i) *Conv In*, the initial convolutional projection of the latent input; (ii) *Block 2*, an early decoder stage; and (iii) *Block 4*, a later decoder stage.

For encoder-induced and generator-produced latents, we collect activations using forward hooks at `pretransform.model.decoder.layers.0`, `pretransform.model.decoder.layers.2.layers.4`, and `pretransform.model.decoder.layers.4.layers.4`, corresponding to *Conv In*, *Block 2*, and *Block 4*, respectively. The activations are flattened and projected onto a one-dimensional LDA direction fitted to distinguish encoder-induced from generator-produced representations.

C Additional LDA analyses on Stable Audio Open

In addition to the LDA analysis of AudioX in Section 2.2, we conduct the same analysis on Stable Audio Open (Evans et al., 2025) to examine whether the observed distribution mismatch extends to another latent audio generative model. As shown in Figure 8, encoder-induced and generator-produced latents remain clearly distinguishable across all three datasets. In the latent space, the LDA AUC is 0.977 on AudioCaps (Kim et al., 2019), 0.916 on MusicCaps (Agostinelli et al., 2023), and 0.898 on VGGSound-Omni (Dai et al., 2026). Clear separation between z_e and z_g also persists in the decoder feature spaces at Conv In, Block 2, and Block 4. These results provide additional empirical evidence that $\mathcal{P}_e \neq \mathcal{P}_g$ across different latent audio generative models and datasets.

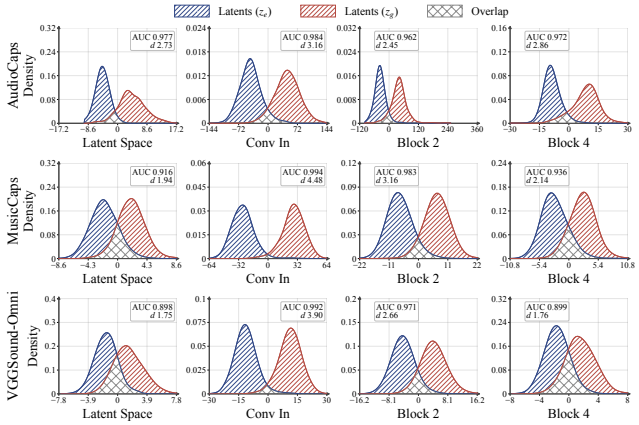
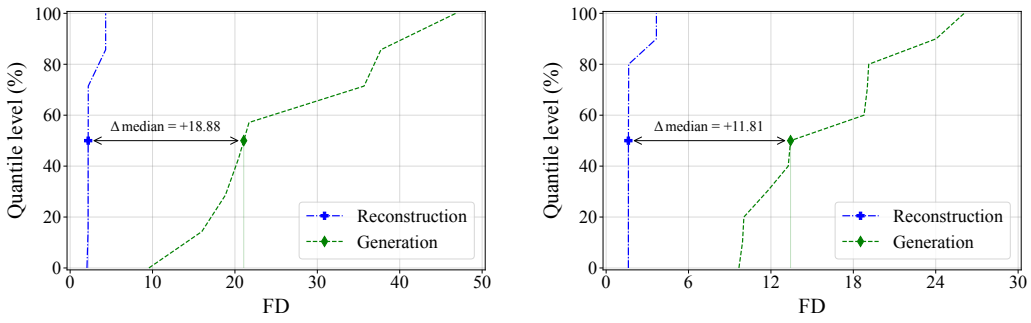


Figure 8: LDA projections of z_e and z_g in Stable Audio Open. Rows correspond to AudioCaps, MusicCaps, and VGGSound-Omni datasets; columns show latent space and decoder features at Conv In, Block 2, and Block 4. Curves show Gaussian-smoothed, normalized histogram densities of the one-dimensional LDA scores.



(a) MusicCaps (b) VGGSound-Omni

Figure 9: **Quantile comparison of rFD and gFD on MusicCaps and VGGSound-Omni.** The rFD and gFD curves are separated across the full quantile range on both datasets, with median gFD-rFD gaps of 18.88 on MusicCaps and 11.81 on VGGSound-Omni.

D Additional Quantile Analysis

In addition to the quantile analysis of AudioCaps in Section 2.3, we compare the empirical quantiles of rFD and gFD on MusicCaps (Agostinelli et al., 2023) and VGGSound-Omni (Dai et al., 2026). As shown in Figures 9, the rFD and gFD curves remain separated across the full quantile range on both datasets. The median gFD-rFD gap is 18.88 on MusicCaps and 11.81 on VGGSound-Omni. Together with the AudioCaps results, the quantile analysis shows a similar reconstruction-generation gap across all three datasets.

E Experimental Details

Training Data. AudioX is originally trained on 33,052.28 hours of audio collected from multiple datasets (Tian et al., 2026). In contrast, our decoder adaptation uses only 508.3 hours of training audio, corresponding to approximately 1.54% of the original AudioX training hours. As summarized in Table 5, we sample data from four text-to-audio datasets used by AudioX: AudioCaps (Kim et al., 2019), WavCaps (Mei et al., 2024), IF-caps (Tian et al., 2026), and AudioTime (Xie et al., 2025). The resulting subset contains 182,392 samples, with approximately ten seconds per example. All waveforms are resampled to 44.1 kHz and represented in stereo. Samples are drawn with replacement across the four datasets, and duplicated training examples may therefore occur.

Table 5: **Comparison of the original AudioX training data and the data used for our decoder adaptation.** AudioX training statistics are reported in Appendix A.1 of (Tian et al., 2026). Our adaptation uses only four text-to-audio datasets and 508.3 hours of audio in total.

Dataset	**AudioX**		**Ours**		
	Clips	**Hours**	**Clips**	**Share**	**Hours**
AudioCaps	45k	125.1	67554	36.9%	187.4
WavCaps	108.3k	300.8	46866	25.6%	130.0
IF-caps	1268k	3,524.4	58324	32.9%	167.1
AudioTime	20k	355.5	9648	4.7%	23.8
Private T2M dataset	175.2k	11679.3	0	0.0%	0.0
V2M	5685.7k	15793.58	0	0.0%	0.0
MUCaps	22k	1273.6	0	0.0%	0.0
Total	**7324.2k**	**33052.28**	**182392**	**100.0%**	**508.3**

TangoMusic. TangoMusic (Kong et al., 2024a) is originally trained on approximately 5,800 hours of AudioSet data (Gemmeke et al., 2017). In contrast, our decoder adaptation uses a substantially smaller subset containing 644 hours of ten-second recordings, corresponding to approximately 11.1% of the original training hours. Following TangoMusic's data preparation procedure (Kong et al., 2024a), Audio Flamingo (Kong et al., 2024b) generates candidate text captions from the source audio, which are then ranked and filtered using CLAP (Wu et al., 2023) audio-text similarity. Up to three captions per recording are retained with CLAP similarity scores of at least 0.35. All waveforms are resampled to 16 kHz and converted to mono.

Data and preprocessing. AudioX uses the codec from Stable Audio Open (Evans et al., 2025). Accordingly, we follow the preprocessing used for Stable Audio Open codec training. We retain approximately the first ten seconds of each recording and normalize each segment to -16 LUFS (Loudness Units relative to Full Scale), with a uniformly sampled loudness jitter of ± 2 dB. These segments are then used to construct the AudioGAR latents z_g^t for decoder adaptation. The latent cache is generated with classifier-free guidance (CFG) (Ho & Salimans, 2021) using a guidance scale of 7.0, matching the CFG setting used for evaluation in AudioX (Tian et al., 2026).

TangoMusic uses the codec from AudioLDM (Liu et al., 2023). We therefore follow the corresponding waveform preprocessing and acoustic frontend, together with the HiFi-GAN training procedure (Kong et al., 2020a). Each waveform is normalized by subtracting its mean and scaling its peak amplitude to half of the full-scale amplitude. We then use the first ten seconds of each recording to construct the AudioGAR latents z_g^t, which are subsequently decoded by the VAE decoder. The latent cache is generated with CFG using a guidance scale of 3.0, matching the setting used for TangoMusic evaluation.

Training Setup. For AudioX, we adapt only the codec decoder while keeping the codec encoder and latent generative model frozen. We follow the Stable Audio Open codec training recipe (Evans et al., 2025), using the same GAN training procedure, loss functions, and scalar loss weights. Training is performed for 100k steps with AdamW, using learning rates of 1×10^{-5} for the decoder and 3×10^{-4} for the discriminator, with an effective batch size of 16 clips. Following the original codec training setup, optimization uses aligned 1.5-second audio crops. Training is conducted on a single NVIDIA H100 80GB GPU and takes approximately 10.3 hours.

For TangoMusic, we jointly adapt the VAE decoder and HiFi-GAN vocoder while keeping the VAE encoder and latent generative model frozen. We follow the HiFi-GAN training procedure and objective (Kong et al., 2020a). Training is performed for 30k steps with AdamW, using a learning rate of 4×10^{-5} for both the VAE decoder and vocoder and 1×10^{-4} for the discriminator, with a batch size of eight clips. Vocoder optimization uses random 0.52-second segments aligned to latent frames, following the HiFi-GAN training procedure (Kong et al., 2020a). Training is conducted on a single NVIDIA H100 80GB GPU and takes approximately 4.8 hours.

F Additional Details on Evaluation Metrics

Following AudioX (Tian et al., 2026), we use the following metrics to evaluate audio and music generation performance.

Fréchet Audio Distance (FAD). FAD measures the distributional discrepancy between generated and real audio in a learned audio feature space, and is analogous to FID (Heusel et al., 2017) in image generation. Specifically, it compares the statistics of embedding distributions extracted from generated and reference audio, with a lower value indicating a smaller discrepancy between the two distributions. We compute FAD using features extracted by VGGish (Hershey et al., 2017).

Fréchet Distance (FD). FD follows the same distribution-level comparison as FAD but uses PANNs (Kong et al., 2020b) embeddings instead of VGGish, providing a complementary view in a different feature space.

Kullback-Leibler Divergence (KL). KL divergence measures the discrepancy between the acoustic-content distributions of generated and reference audio. Following prior work, we obtain multi-label class predictions from a pretrained PANNs model and compute the divergence between the corresponding prediction distributions. Lower KL indicates greater similarity between generated and reference audio in terms of the predicted acoustic content.

Inception Score (IS). IS evaluates generated samples based on both quality and diversity, favoring confident per-sample class predictions while encouraging variety across the generated set (Salimans et al., 2016). Following prior audio-generation work (Donahue et al., 2019; Majumder et al., 2024; Liu et al., 2023), we use a pretrained PANNs model (Kong et al., 2020b) to obtain the class-prediction distributions used to compute IS.

Production Complexity (PC) and Production Quality (PQ). PC and PQ are introduced in the Meta Audiobox Aesthetics framework (Tjandra et al., 2025) as no-reference measures of audio aes-

thetics. PQ captures technical production quality such as clarity and frequency balance; PC captures scene complexity, reflecting the presence of multiple sound components. Unlike distribution-level metrics such as FD and FAD, both metrics can be evaluated independently for each audio clip without requiring a reference sample.

G LIMITATIONS

AudioGAR relies on a denoising trajectory through the frozen latent diffusion model, and is therefore directly applicable to latent diffusion and flow-matching generative models. It does not directly extend to autoregressive latent generators such as AudioGen (Kreuk et al., 2023) and MusicGen (Copet et al., 2023), which do not provide the same intermediate denoising trajectory.

The noise level η_t controls the trade-off between source correspondence and generation relevance, and intermediate values generally work best in our experiments. However, the preferred η_t may depend on the generative model, codec architecture, and dataset. We currently select it empirically rather than with a principled rule.

Our AudioGAR adaptation experiments focus on AudioX and TangoMusic for text-to-audio and text-to-music generation, although we observe the decoder train-generation mismatch across a broader set of models and datasets. It remains to be tested whether the same adaptation strategy transfers to additional latent generative models, codec architectures, and conditioning settings such as video-to-audio and image-to-audio generation. We also evaluate with automatic metrics only and do not include perceptual listening studies.