FLORE-AE: RETHINKING AUDIO AUTOENCODING WITH WAVEFORM DIFFUSION DECODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Audio autoencoders for latent generation aim to combine compact representations with faithful and perceptually realistic waveform reconstruction. Existing spectral–adversarial decoders use learned discriminators, introducing minimax optimization. In this work, we instead formulate high-compression audio autoencoding as latent-conditioned waveform diffusion, where compact latents encode source structure and the diffusion decoder models remaining acoustic detail. **FLORE-AE** combines **carrier–amplitude factorization** for locally normalized synthesis, **FLORE-DiT** for time-aligned latent guidance and multiresolution context, and **Endpoint-Aware MeanFlow** for two-evaluation decoding of encoded and generated latents. Matched controls show that local factorization improves all reconstruction criteria, alignment drives paired fidelity, multiresolution computation matches full-resolution quality at lower inner-DiT cost, and endpoint-aware adaptation improves generated-latent decoding. Across speech, music, and sound, FLORE-AE improves reconstruction over the evaluated spectral–adversarial and generative autoencoders while supporting generation and editing at high compression. These results demonstrate a practical non-adversarial waveform decoder for compact audio representations. Audio examples are available on our demo page.

1 INTRODUCTION

Latent audio generation relies on autoencoders to combine efficient generative modeling with faithful waveform reconstruction (Liu et al., 2023; Evans et al., 2024). These systems aim to balance three requirements: **compactness**, limiting storage and processing cost; **generative tractability**, making its distribution amenable to learning and sampling; and **reconstruction quality**, preserving source fidelity and perceptual realism (Parker et al., 2026; Heek et al., 2026). The central challenge is to improve reconstruction without shifting an excessive modeling burden to the downstream generator.

The prevailing spectral–adversarial paradigm couples reconstruction learning to a jointly optimized discriminator (Evans et al., 2024; Wang et al., 2025; Parker et al., 2026). Spectral losses supervise source correspondence, while discriminator feedback encourages perceptual realism. This feedback, however, introduces an adversarial game whose updates can be unstable (Mescheder et al., 2018; Pasini et al., 2024); representation learning therefore couples spectral fidelity and latent regularization to an evolving real/fake objective. Increasing the latent rate can preserve more detail, but makes downstream generation more expensive (Heek et al., 2026; Parker et al., 2026). We ask whether a compact latent can instead support source-faithful waveform synthesis without adversarial decoder training.

Conditional generative decoding offers a different formulation. Rather than requiring a compact latent to specify every waveform sample, it lets the representation encode source structure while the decoder models residual acoustic variation (Song et al., 2021; Preechakul et al., 2022). Diffusion learns this conditional distribution by regression along prescribed noise–data paths without a learned critic (Lipman et al., 2023); a continuous interface also avoids committing the downstream model to quantized codec tokens (Défossez et al., 2023; Kumar et al., 2023). High compression nevertheless creates three challenges: a shared noise schedule exposes quiet and loud regions to different local signal-to-noise ratios; a latent sequence thousands of times shorter than the waveform needs to control dense synthesis without placing all context computation on the finest grid; and few-step decoding should remain effective on generated latents.

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.

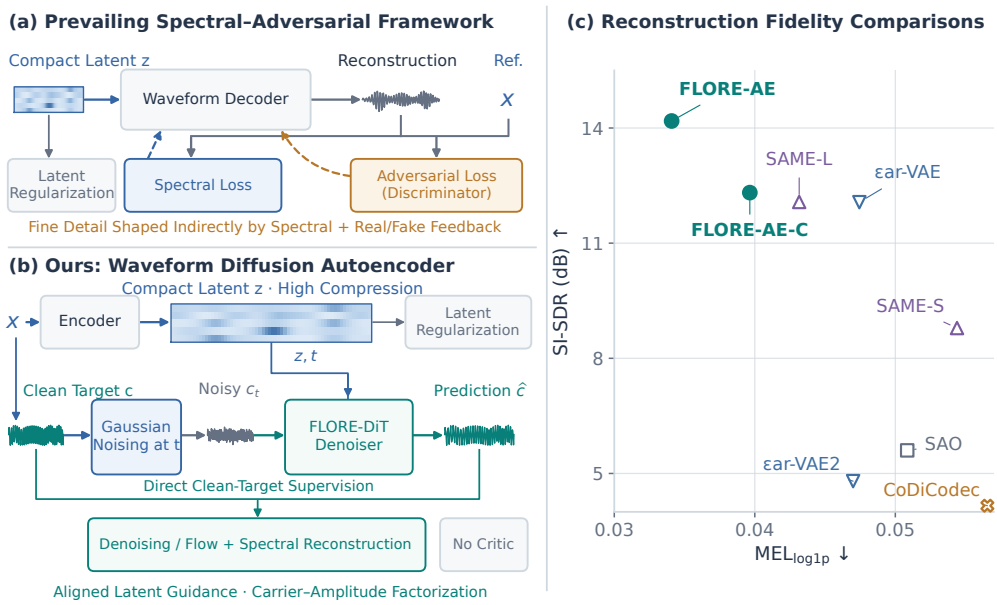


Figure 1: **Rethinking audio autoencoding with waveform diffusion.** Both frameworks regularize compact latents with paired spectral reconstruction. **(a)** The prevailing framework also uses a discriminator. **(b)** FLORE-AE combines latent-conditioned waveform flow regression, aligned guidance, and carrier–amplitude factorization without a learned critic. **(c)** Six-test-set reconstruction fidelity (Table 1); FLORE-AE and FLORE-AE-C use $2048\times$ and $4096\times$ temporal compression, and CoDiCodec is indicated off-axis.

In this work, we introduce **FLORE-AE**, a high-compression audio autoencoder that jointly learns compact continuous representations and a conditional waveform diffusion decoder through clean-target flow matching (Li & He, 2025). Its $2048\times$ and $4096\times$ operating points use one latent step per 2,048 or 4,096 waveform samples. Local amplitude variation changes corruption strength under a shared noise schedule (Zhou et al., 2026); normalization improves denoising conditioning but removes amplitude dynamics. Because the encoder still observes the natural-amplitude waveform, the representation remains responsible for the scale information excluded from the diffusion target. **Carrier–amplitude factorization** therefore pairs normalized carrier synthesis with latent-predicted global gain and local dynamics.

Beyond signal parameterization, faithful reconstruction requires local synthesis to follow the latent timeline while using broader context. Waveform U-Nets pass conditions through intermediate features (Schneider et al., 2023), whereas fixed-resolution DiTs handle alignment and context on one fine grid (Peebles & Xie, 2023; Zhou et al., 2026). Our **FLORE-DiT** instead expands ordered latents into position-specific conditions, reinjects them at every resolution, and moves broader context modeling to coarser sequences. Stage-wise latent access keeps source guidance direct, while the coarser sequences broaden context at lower cost without enlarging the representation.

Few-step decoding creates a further mismatch because clean-latent reconstruction need not prepare a decoder for generated latents. **Endpoint-Aware MeanFlow (EA-MF)** combines finite-interval learning (Geng et al., 2025a;b) with spectral supervision near the data endpoint. It reduces four decoder evaluations to two, maintains comparable paired reconstruction, improves generated-latent FAD over matched iMF, and reaches 0.008 RTF.

Across VCTK, EXPRESSO, Song Describer, MusicCaps, AudioCaps, and VGGSound, FLORE-AE leads the reconstruction baselines with 14.11 dB SI-SDR, 0.041 rFAD, and 4.173 ± 0.253 MOS. At $4096\times$, FLORE-AE-C improves SI-SDR (12.31 vs. 12.07 dB) and rFAD (0.061 vs. 0.088) over SAME-L with 61% fewer parameters; both variants use two evaluations and reach 0.008 RTF. FLORE-AE also leads the evaluated AudioCaps and Song Describer systems in FAD and CLAP and improves editing fidelity. Controls attribute these gains to local scale control, alignment, and lower-cost multiresolution computation.

Our contributions are as follows:

- We formulate high-compression audio autoencoding as conditional waveform diffusion, jointly learning continuous latents and a decoder without adversarial waveform training.
- We introduce carrier–amplitude factorization and FLORE-DiT; controls show that local scale control improves all reconstruction metrics, alignment produces the principal fidelity gain, and the hierarchy matches full-resolution quality at lower inner-DiT cost.
- We introduce EA-MF for two-evaluation decoding and demonstrate a quality–latency operating point that leads the evaluated reconstruction baselines, supports generation and editing, and approaches single-pass spectral–adversarial latency.

2 Related Work

Audio autoencoders learn compact representations for waveform reconstruction and downstream generative modeling (Liu et al., 2023; Evans et al., 2024). Continuous systems commonly combine spectral reconstruction losses with adversarial supervision, including phase-sensitive objectives and semantic or generative latent alignment (Evans et al., 2024; Wang et al., 2025; Parker et al., 2026). Diffusion and consistency-based alternatives instead learn generative reconstruction without an adversarial waveform critic (Preechakul et al., 2022; Schneider et al., 2023; Pasini et al., 2024; 2025). These works establish generative decoding as a viable alternative, while FLORE-AE focuses on how a highly compressed, time-indexed latent sequence can guide waveform synthesis with both local precision and broader context.

Waveform diffusion models typically condition on spectrograms or low-rate waveform observations (Kong et al., 2020; Chen et al., 2020b; Lee & Han, 2021). DiTs provide contextual interactions at a fixed token resolution, while hierarchical transformers move part of the computation to coarser resolutions (Peebles & Xie, 2023; Crowson et al., 2024). Audio autoencoding adds a distinct requirement: the jointly learned compact condition must preserve source timing throughout waveform reconstruction. FLORE-DiT addresses this requirement through position-specific latent expansion and resolution-aligned conditioning, while local carrier–envelope factorization separates waveform structure from time-varying amplitude. Appendix A provides an extended discussion of the waveform-generation lineage.

3 FLORE-AE

FLORE-AE jointly learns compact audio representations with a waveform diffusion decoder (Figure 2), formulating waveform reconstruction as conditional generation (Section 3.1). Carrier–amplitude factorization enables locally normalized synthesis with latent-predicted amplitude recovery (Section 3.2), while FLORE-DiT aligns latent guidance across fine-scale synthesis and lower-resolution context modeling (Section 3.3). EA-MF couples finite-interval learning with data-endpoint reconstruction supervision for two-step decoding (Section 3.4). Finally, Section 3.5 presents the two-stage training procedure and shared decoding interface for reconstruction and generation. Appendices B and C give the implementation and formal motivation: factorization makes the carrier approximately gain-equivariant, while expansion changes position-wise access, not latent information content. These arguments do not establish empirical superiority by themselves.

3.1 Waveform Diffusion Autoencoding

High-compression audio autoencoding requires reconstructing detailed waveforms from compact latent representations. The retained information need not uniquely determine fine acoustic detail. We therefore formulate decoding as conditional generation, using the latent representation to guide source structure while modeling the remaining acoustic variability (Preechakul et al., 2022; Schneider et al., 2023).

Given a waveform $\boldsymbol{x} \in \mathbb{R}^T$, the encoder defines a conditional distribution $q_\phi(\boldsymbol{z}_c \mid \boldsymbol{x})$ over continuous latent representations $\boldsymbol{z}_c \in \mathbb{R}^{N \times D}$, where $N = \lceil T/R \rceil$, R is temporal downsampling, and D is latent width. During training, a sample $\boldsymbol{z}_c \sim q_\phi(\cdot \mid \boldsymbol{x})$ conditions the decoder.

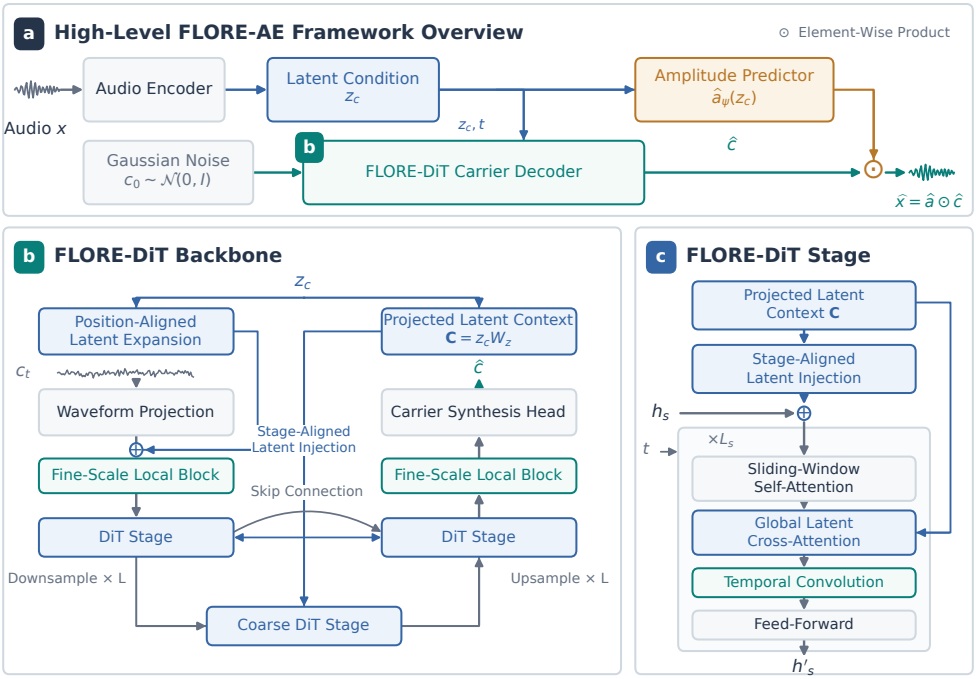


Figure 2: **FLORE-AE and FLORE-DiT. (a)** FLORE-AE overview. **(b)** Detail of the FLORE-DiT carrier decoder in (a), showing aligned latent conditioning and multiresolution waveform modeling. **(c)** Detail of one FLORE-DiT stage in (b), showing stage-aligned injection, latent cross-attention, and local waveform modeling.

We train the decoder using flow matching with clean-waveform prediction (Lipman et al., 2023; Li & He, 2025). For a clean target waveform $\boldsymbol{s}$, we sample a time $t \in [0, 1)$ and independent Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. The decoder predicts $\boldsymbol{s}$ from its noisy version and the latent condition:

$$\boldsymbol{s}_t = (1-t)\boldsymbol{\epsilon} + t\boldsymbol{s}, \qquad \widehat{\boldsymbol{s}}_t = f_\theta(\boldsymbol{s}_t, t, \boldsymbol{z}_c). \tag{1}$$

The clean-target regression objective is

$$\mathcal{L}_{\text{pred}} = \mathbb{E}\left[T^{-1}\left\|f_\theta(\boldsymbol{s}_t, t, \boldsymbol{z}_c) - \boldsymbol{s}\right\|_2^2\right]. \tag{2}$$

The clean prediction parameterizes the velocity field as

$$\boldsymbol{v}_\theta(\boldsymbol{s}_t, t, \boldsymbol{z}_c) = \frac{f_\theta(\boldsymbol{s}_t, t, \boldsymbol{z}_c) - \boldsymbol{s}_t}{1-t}. \tag{3}$$

With the encoder-induced condition distribution held fixed, the optimal clean-target predictor induces the corresponding conditional flow-matching field. Conditional modeling alone, however, does not ensure source-faithful reconstruction: fidelity depends on both the information retained in $\boldsymbol{z}_c$ and how the decoder uses it (Appendix C.1). During joint training, denoising gradients shape this representation, while latent regularization constrains it.

3.2 Waveform Normalization with Amplitude Recovery

Clip-level normalization (Zhou et al., 2026) controls the overall signal scale in waveform diffusion. With an unnormalized waveform target $\boldsymbol{s} = \boldsymbol{x}$, a region of mean-square energy e has effective SNR $t^2 e/(1-t)^2$. Thus, a shared noise schedule produces different local SNRs in quiet and loud passages, even after clip-level scaling. Local normalization can reduce this amplitude dependence, but the resulting carrier no longer contains the amplitude dynamics needed to reconstruct the natural waveform. We therefore introduce **carrier–amplitude factorization**.

Let $\boldsymbol{a}(\boldsymbol{x})$ be a positive, smoothed local-RMS envelope. We separate the waveform into a normalized carrier and its amplitude envelope, and reconstruct it as

$$\boldsymbol{c} = \boldsymbol{x} \oslash \boldsymbol{a}(\boldsymbol{x}), \qquad \widehat{\boldsymbol{x}} = \widehat{\boldsymbol{a}}_\psi(\boldsymbol{z}_c) \odot \widehat{\boldsymbol{c}}. \tag{4}$$

Setting $\boldsymbol{s} = \boldsymbol{c}$ in Equation 1 makes the normalized carrier the diffusion target; we write $\widehat{\boldsymbol{c}}_t = f_\theta(\boldsymbol{c}_t, t, \boldsymbol{z}_c)$ for its random-time prediction. The encoder processes the natural-amplitude waveform, while $\widehat{\boldsymbol{a}}_\psi$ predicts global gain and local amplitude dynamics from the same latent representation. Carrier synthesis and amplitude recovery are therefore coupled through the shared representation.

3.3 FLORE-DiT

High-compression waveform autoencoding poses a demanding conditional reconstruction problem: source-faithful waveform recovery relies on a compact latent sequence, while the diffusion state remains substantially denser even after patchification. Consequently, each latent interval serves multiple synthesis positions, especially at high noise levels where the waveform state carries little source-specific structure.

Waveform modeling also requires fine temporal coordinates for local phase, transients, and event timing, yet broader context for long-range structure. A fixed-resolution backbone assigns both roles to one token granularity: coarser patches place more local structure within each token, whereas finer patches lengthen contextual interactions. We therefore introduce **FLORE-DiT**, a temporally aligned diffusion transformer that combines position-specific latent conditioning with multiresolution waveform modeling (Figure 2(b–c)).

3.3.1 Temporally Aligned Conditioning

An ordered query resampler (Parker et al., 2026) expands each latent interval into position-specific conditions on the fine waveform grid. Let $\boldsymbol{c}_t$ denote the noisy carrier, $\boldsymbol{h}^{(s)}$ the waveform features at stage s, and $\mathbf{C} = \boldsymbol{z}_c W_z$ the latent context under a learned projection W_z. FLORE-DiT combines the initial expansion with direct latent injection at every internal stage:

$$\begin{aligned} \boldsymbol{h}^{(0)} &= \mathcal{P}_\theta(\boldsymbol{c}_t) + \mathcal{S}_\theta(\boldsymbol{z}_c), \\ \boldsymbol{h}^{(s)} &\leftarrow \boldsymbol{h}^{(s)} + \boldsymbol{\gamma}_s \odot \mathcal{I}_s(\mathbf{C}). \end{aligned} \tag{5}$$

Here $\mathcal{P}_\theta$ projects overlapping waveform patches, $\mathcal{S}_\theta$ produces the expanded conditions, $\mathcal{I}_s$ aligns the latent context with the temporal grid of stage s, and $\boldsymbol{\gamma}_s$ gates its contribution. The expansion gives synthesis positions within one latent interval distinct conditions. Stage-wise injection gives each resolution direct access to the original time-indexed latents rather than routing them only through intermediate waveform features. Global latent cross-attention complements these aligned paths with content-dependent access to the full latent sequence.

3.3.2 Multiresolution Waveform Modeling

The conditioned waveform features pass through a downsampling–upsampling hierarchy with L factor-two reductions, based on hierarchical diffusion transformers (Crowson et al., 2024); stage s contains L_s DiT blocks. Fine-resolution local blocks operate before the first downsampling and after the final upsampling, while skip connections preserve features across the contextual hierarchy.

Within stage s, sliding-window self-attention restricts token i to a local neighborhood. For window radius w, waveform projection stride P, and cumulative reduction factor d_s,

$$\mathcal{N}_s(i) = \{j : |j - i| \le w\}, \qquad \mathrm{span}_{\mathrm{wav}}\big(\mathcal{N}_s(i)\big) \approx (2w+1)Pd_s. \tag{6}$$

The same token-space window therefore covers progressively broader waveform context at coarser stages. This organization assigns fine synthesis and contextual integration to appropriate temporal scales, while aligned latent injection maintains a common source reference throughout the hierarchy.

3.4 Endpoint-Aware MeanFlow

Compact latent modeling reduces downstream generation burden, but iterative waveform decoding can offset this benefit. Few-step decoding also changes the task: the pretrained decoder parameterizes instantaneous motion, whereas Improved MeanFlow (iMF) learns an average velocity from start time t to interval endpoint r, where $0 \le t \le r \le 1$ and $t < 1$ (Geng et al., 2025a;b). Because r need not equal the clean-data endpoint 1, **Endpoint-Aware MeanFlow (EA-MF)** retains general interval learning while supervising instantaneous predictions and intervals ending at or near 1.

We initialize an interval-conditioned predictor from the pretrained decoder and retain its clean-prediction-to-velocity parameterization from Equation 3. Let $u = u_\vartheta(\boldsymbol{c}_t, t, r, \boldsymbol{z}_c)$. The predicted interval state and its corresponding clean-carrier estimate are

$$\widehat{\boldsymbol{c}}_r = \boldsymbol{c}_t + (r-t)u, \qquad \widehat{\boldsymbol{c}}^{\,\mathrm{aux}}_{t,r} = \boldsymbol{c}_t + (1-t)u. \tag{7}$$

When $r = t$, the second expression recovers the instantaneous clean prediction. When $r = 1$, it coincides with the predicted interval endpoint. For $t < r < 1$, it extrapolates the average motion by $(1-r)u$ and is used only for reconstruction regularization.

Near-endpoint intervals are sampled independently of the main iMF pairs, limiting extrapolation without changing their interval distribution or velocity targets. After amplitude restoration (Equation 4), both clean-carrier estimates are compared with the source using a multi-resolution short-time Fourier transform (MRSTFT) loss (Yamamoto et al., 2020).

3.5 Training and Inference

Stage 1: Joint autoencoder training. We jointly train the encoder, carrier predictor, and amplitude predictor with sampled latent conditions $\boldsymbol{z}_c \sim q_\phi(\cdot \mid \boldsymbol{x})$. Writing $\widehat{\boldsymbol{x}}_t = \widehat{\boldsymbol{a}}_\psi(\boldsymbol{z}_c) \odot \widehat{\boldsymbol{c}}_t$ for the amplitude-restored random-time prediction, we optimize

$$\mathcal{L}_{\mathrm{AE}} = \mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{amp}}\mathcal{L}_{\mathrm{amp}} + \lambda_{\mathrm{spec}}\mathcal{L}_{\mathrm{MRSTFT}}(\widehat{\boldsymbol{x}}_t, \boldsymbol{x}) + \mathcal{R}_{\mathrm{lat}}. \tag{8}$$

$\mathcal{L}_{\mathrm{amp}}$ supervises amplitude recovery, while $\mathcal{L}_{\mathrm{MRSTFT}}$ compares the amplitude-restored random-time prediction with the source without a sampling rollout. All reconstruction terms update the encoder. $\mathcal{R}_{\mathrm{lat}}$ is a standard-normal KL bottleneck (Kingma & Welling, 2014) with coefficient $\lambda_{\mathrm{KL}} = 10^{-5}$.

Stage 2: Endpoint-aware decoder training. EA-MF adapts the waveform predictor and its interval-conditioning branch, with the encoder and amplitude predictor fixed. Using deterministic encoder outputs $\boldsymbol{z}_c = E_\phi(\boldsymbol{x})$ to match the deployed latent interface, it minimizes

$$\mathcal{L}_{\text{EA-MF}} = \mathcal{L}_{\mathrm{MF}} + \lambda_{\mathrm{inst}}\mathcal{L}^{\mathrm{inst}}_{\mathrm{spec}} + \lambda_{\mathrm{near}}\mathcal{L}^{\mathrm{near}}_{\mathrm{spec}}, \tag{9}$$

where $\mathcal{L}_{\mathrm{MF}}$ is the iMF velocity objective (Geng et al., 2025b), and the spectral terms supervise the amplitude-restored instantaneous and near-endpoint estimates in Equation 7, respectively.

Inference. Reconstruction uses clean encoder outputs, whereas generation uses latents from an independently trained downstream generator. In both cases, FLORE-AE starts from Gaussian carrier noise and applies two EA-MF updates on the grid $(0, 0.5, 1)$. The amplitude predictor then restores the waveform scale from the same latents (Equation 4); no reference amplitude envelope is supplied.

4 Experiments

4.1 Experimental Setup

Implementation details. We train on 131,072-sample, 44.1-kHz mono crops (approximately 2.97 s) from frozen FMA, Freesound, and WavCaps (Mei et al., 2024) manifests. FLORE-AE and FLORE-AE-C use $(R, D) = (2048, 192)$ and $(4096, 256)$, respectively, and their base checkpoints use 900,000/300,000 updates on 8 NVIDIA A800 GPUs. EA-MF fine-tunes only the waveform decoder for 60,000 updates and uses the grid $(0, 0.5, 1)$. Appendix B.6.2 gives data and optimization details.

Evaluation protocols. Reconstruction uses VCTK/EXPRESSO (Yamagishi et al., 2019; Nguyen et al., 2023), Song Describer/MusicCaps (Manco et al., 2023; Agostinelli et al., 2023), and AudioCaps/VGGSound (Kim et al., 2019; Chen et al., 2020a) for speech, music, and sound, respectively. Metrics first average recordings per dataset; SI-SDR and spectral scores then average all six datasets equally, whereas rFAD averages AudioCaps and Song Describer. Generation evaluates sound on AudioCaps and music on Song Describer using the designated text-conditioned generator for each frozen autoencoder interface. Within each dataset, prompts, evaluation population, sampling, and metrics are fixed; because generator architectures are not the controlled factor, these are system-level comparisons. Editing reuses the AudioCaps generator with training-free masked inference following Stable Audio 3 (Evans et al., 2026); FLORE-AE and FLORE-AE-C each use four decoder evaluations for editing. RTF uses 30-s, batch-one FP32 decoding on 1 NVIDIA A800 GPU.

Table 1: **Audio reconstruction across speech, music, and sound.** SI-SDR, STFT, and MEL are equal-weight means over the six test sets; rFAD averages AudioCaps and Song Describer, and MOS is reported as mean ± standard deviation over 60 item-level MOS values, each averaging five listener scores. SI-SDR is in dB; † gives CoDiCodec's equivalent sequence compression; bold marks the best result.

Model	Temporal compression	NFE	Params. (M)	Reconstruction quality					Efficiency
				SI-SDR ↑	$\text{STFT}_{\log 1p}$ ↓	$\text{MEL}_{\log 1p}$ ↓	rFAD ↓	MOS ↑	RTF ↓
Single-pass continuous autoencoders									
ϵar-VAE	1024×	1	147.8	12.06	0.055	0.047	0.054	3.921 ± 0.321	0.010
ϵar-VAE2	1920×	1	178.4	4.79	0.059	0.047	0.048	4.021 ± 0.248	0.029
Stable Audio Open	2048×	1	156.1	5.60	0.063	0.051	0.101	3.837 ± 0.330	0.009
SAME-S	4096×	1	108.2	8.78	0.065	0.054	0.136	3.868 ± 0.241	**0.002**
SAME-L	4096×	1	852.1	12.07	0.058	0.043	0.088	3.921 ± 0.336	0.010
Generative continuous autoencoders									
Moûsai (v2)	1024×	20	185.1	-27.71	0.088	0.103	0.341	2.675 ± 0.140	0.058
Music2Latent	4096×	1	58.2	-2.42	0.071	0.066	0.089	3.238 ± 0.421	0.026
CoDiCodec	4096 × †	2	148.5	0.37	0.069	0.065	0.074	3.579 ± 0.413	0.092
Ours									
FLORE-AE	2048×	2	259.7	**14.11**	**0.052**	**0.034**	**0.041**	$\mathbf{4.173 \pm 0.253}$	0.008
FLORE-AE-C	4096×	2	335.3	12.31	0.059	0.043	0.061	4.086 ± 0.280	0.0080

Baselines and metrics. Baselines span spectral–adversarial Stable Audio Open, ϵar-VAE/2, and SAME-S/L (Evans et al., 2024; Wang et al., 2025; 2026; Parker et al., 2026), and generative Moûsai, Music2Latent, and CoDiCodec decoders (Schneider et al., 2023; Pasini et al., 2024; 2025). We use released continuous latents and native decoders, with CoDiCodec in continuous mode. Reconstruction reports SI-SDR, log1p multiresolution STFT/mel, LAION-CLAP FAD (Kilgour et al., 2019; Wu et al., 2022), and five-point MOS; we write rFAD for reconstruction and FAD for generation. Generation also reports CLAP; editing adds full/edited-region FAD, context SI-SDR/STFT, text alignment, and generated–original consistency. Appendix B.6.4 details all metrics.

4.2 Main Results

4.2.1 Audio reconstruction

Table 1 supports three conclusions: (1) FLORE-AE gives the strongest objective and subjective point estimates across the evaluated speech, music, and sound tests, leading all evaluated baselines on four objective metrics and MOS. Its SI-SDR (14.11 vs. 12.07 dB), rFAD (0.041 vs. 0.048), and MOS (4.173 ± 0.253) provide complementary evidence of source fidelity, distributional quality, and perceptual quality; (2) at $4096\times$, FLORE-AE-C improves rFAD (0.061 vs. 0.088) and SI-SDR (12.31 vs. 12.07 dB) over SAME-L with 61% fewer parameters, although SAME-L retains lower log1p spectral errors; and (3) with two decoder evaluations, FLORE-AE and FLORE-AE-C both reach 0.008 RTF, comparable to single-pass Stable Audio Open (0.009), ϵar-VAE (0.010), and SAME-L (0.010), and faster than Moûsai (0.058) and CoDiCodec (0.092). This operating point makes waveform diffusion autoencoding a practical alternative to spectral–adversarial decoding. Appendix D.6 reports deployed parameter accounting and the timing scope; the reported 0.008 RTFs are 30-s A800 decoder-only measurements excluding latent generation and streaming latency.

4.2.2 Downstream tasks

Text-to-audio generation. Table 2 tests downstream sound and music generation. FLORE-AE leads AudioCaps at 0.086/0.321 FAD/CLAP and Song Describer at 0.2703/0.328; FLORE-AE-C reaches 0.115/0.312 and 0.293/0.298. Appendix B.2 notes that the designated generators directly model the native 192/256-channel interfaces, while the AudioCaps-only, single-seed trajectory in Appendix D.7.2 shows FLORE-AE improving through 200k updates and FLORE-AE-C peaking earlier. Thus wider-interface compatibility is contract-specific and rankings depend on generator-training maturity.

Audio editing. Table 3 tests whether the latent–decoder interface can generate a missing region without corrupting observed audio. Under the matched AudioCaps protocol, FLORE-AE gives the

Table 2: **Text-to-audio generation on AudioCaps (sound) and Song Describer (music).** Each row pairs a frozen autoencoder with its designated generator; prompts, evaluation sets, sampling, and metrics are fixed within each dataset.

	Sound (AudioCaps)		Music (Song Describer)	
Autoencoder	**FAD ↓**	**CLAP ↑**	**FAD ↓**	**CLAP ↑**
ϵar-VAE2	0.098	0.289	0.291	0.298
Stable Audio Open	0.160	0.318	0.327	0.305
SAME-L	0.205	0.284	0.357	0.269
CoDiCodec	0.116	0.290	0.312	0.308
FLORE-AE	**0.086**	**0.321**	**0.2703**	**0.328**
FLORE-AE-C	0.115	0.312	0.293	0.298

strongest joint context–generation operating point: relative to the strongest external results, it reduces full-output FAD by 19%/14% and edited-region FAD by 16%/7%, while improving context SI-SDR by 2.18/2.42 dB for inpainting/continuation. FLORE-AE-C retains strong fidelity at $4096\times$, reaching full-output FAD of 0.072/0.109 and context SI-SDR of 7.45/7.43 dB. External baselines retain the highest text-alignment point estimates, so the results support improved context preservation and overall fidelity rather than uniform dominance in prompt alignment. Complete metrics are reported in Appendix D.2. Appendix D also specifies the matched downstream contract and fixed-latent decoder controls: terminal reconstruction gives the strongest paired reconstruction among the adapted decoders, whereas EA-MF gives the best generated-latent FAD. The latter conclusion remains specific to the matched generator and adaptation setting.

Table 3: **Training-free audio editing on AudioCaps.** Inpainting reconstructs an internal gap; continuation extends an observed prefix. FAD is measured in CLAP space and Ctx. denotes the observed, unedited audio; complete metrics are in Appendix D.2.

	Single-region inpainting				Continuation			
Autoencoder	**FAD_{full} ↓**	**FAD_{edit} ↓**	**Ctx. SI-SDR↑**	**$\text{CLAP}_{\text{text}}$ ↑**	**FAD_{full} ↓**	**FAD_{edit} ↓**	**Ctx. SI-SDR↑**	**$\text{CLAP}_{\text{text}}$ ↑**
ϵar-VAE2	0.058	0.100	-1.00	**0.167**	0.087	0.107	-1.16	0.267
Stable Audio Open	0.181	0.141	0.20	0.140	0.194	0.195	-0.13	0.269
SAME-L	0.150	0.154	7.18	0.125	0.184	0.232	7.04	0.238
CoDiCodec	0.084	0.189	-6.33	0.062	0.156	0.209	-6.69	0.227
FLORE-AE	**0.046**	**0.084**	**9.36**	0.145	**0.075**	**0.099**	**9.46**	0.278
FLORE-AE-C	0.072	0.109	7.45	0.135	0.109	0.135	7.43	0.268

4.3 Analysis

Table 4 identifies three requirements for a practical high-compression waveform diffusion autoencoder: the denoising target must be well conditioned across local amplitude scales, compressed latents must retain direct time-indexed control over dense synthesis, and few-step decoding must remain effective on generated as well as clean latents.

Signal parameterization: local scale control. Panel (a) distinguishes local factorization from clip-level normalization. Predicting one global RMS lowers rFAD from 0.344 to 0.149, but reduces SI-SDR and increases mel error. By contrast, the local carrier–envelope coordinate is the only tested variant that improves all four metrics over fixed-scale prediction, raising SI-SDR from 12.32 to 13.45 dB and lowering rFAD from 0.344 to 0.110. An inverse-energy loss control remains near the fixed-scale baseline (Appendix D.5), so the gain is not reproduced by merely upweighting quiet regions. The evidence supports a general design principle: normalize the local variation that complicates denoising, but require the latent representation to reconstruct rather than discard source dynamics.

FLORE-DiT: alignment for fidelity, hierarchy for efficiency. Panel (b) separates two roles that standard backbones conflate. A waveform U-Net supplies useful local and multiscale structure,

yet source guidance still passes through intermediate features; adding FLORE alignment improves SI-SDR from 6.67 to 7.65 dB and lowers both spectral errors. A standard patchify DiT already obtains strong distributional similarity (0.108 rFAD), but only 10.07 dB SI-SDR: globally plausible audio is not necessarily a faithful reconstruction of the input. Explicit alignment raises SI-SDR to 13.33 dB and improves STFT/mel, while rFAD changes to 0.116. Its transfer across U-Net and DiT therefore identifies direct source-to-waveform correspondence, rather than backbone capacity alone, as the dominant paired-fidelity mechanism.

Once that correspondence is available, contextual computation need not remain on the finest waveform grid. The full-resolution FLORE-DiT control uses 0.274 inner-DiT TFLOPs, yet is nearly indistinguishable from the hierarchy in quality (13.42 vs. 13.45 dB SI-SDR); the hierarchy uses 0.098 TFLOPs, a 64% reduction under the stated counting scope. Multiresolution computation should therefore be interpreted as compute allocation rather than an accuracy-only module: keep source guidance and local synthesis temporally precise, while moving broader contextual reasoning to shorter, coarser sequences. Appendix D.4 supplies the path controls and TFLOP scope; its capacity sensitivity (Appendix D.4.4) finds only a 0.24-dB macro SI-SDR gain from a 3.6× larger decoder, supporting Base as the quality–efficiency point rather than a capacity maximum.

EA-MF: deployment-aware few-step decoding. Panel (c) shows that the two-evaluation iMF control improves generation FAD only modestly over four-evaluation Conditional FM (0.111 vs. 0.118). At the same two-evaluation budget, EA-MF lowers FAD further to 0.086. Together with the measured 0.008 RTF, this shows that the FAD gain does not require additional decoder evaluations. Decoder objectives should therefore be assessed on generated latents under the deployed interface, rather than inferred from clean-latent reconstruction alone. The conclusion is specific to the matched generator and adaptation setting; complete substitutions are in Appendix D.3.

Table 4: **Controlled analysis of FLORE-AE.** Panels (a–b) use matched training and four-evaluation decoding; panel (c) holds generated latents fixed across decoder objectives. Inner-DiT TFLOPs are per 3-s decoder call; complete control and counting details are in Appendix D.4.

(a) Signal parameterization

Variant	SI-SDR ↑	STFT ↓	MEL ↓	rFAD ↓
Fixed-scale waveform	12.32	0.056	0.041	0.344
Global RMS + predicted gain	11.95	0.055	0.044	0.149
Local carrier + predicted envelope	**13.45**	**0.052**	**0.036**	**0.110**

(b) FLORE-DiT architecture

Decoder	FLORE align.	Resolution hierarchy	Inner-DiT TFLOPs	SI-SDR ↑	STFT ↓	MEL ↓	rFAD ↓
Waveform U-Net	×	—	n/a	6.67	0.122	0.148	0.300
U-Net + FLORE alignment	✓	—	n/a	7.65	0.110	0.133	0.332
Standard patchify DiT	×	×	0.309	10.07	0.057	0.043	**0.108**
+ FLORE alignment	✓	×	0.309	13.33	0.053	0.037	0.116
FLORE-DiT, full resolution	✓	×	0.274	13.42	0.053	**0.036**	0.112
FLORE-DiT	✓	✓	**0.098**	**13.45**	**0.052**	**0.036**	0.110

(c) Few-step decoder objectives

Method	NFE	FAD ↓
Conditional FM	4	0.118
iMF terminal control	2	0.111
EA-MF	2	**0.086**

5 Conclusion

FLORE-AE reformulates high-compression audio autoencoding as conditional waveform generation without adversarial waveform training. Carrier–amplitude factorization controls local scale while recovering source dynamics, FLORE-DiT aligns latent guidance while moving context computation across resolutions, and EA-MF enables two-evaluation decoding. Across reconstruction, generation, and editing, the representations retain fidelity and downstream utility under high compression, establishing waveform diffusion as a practical decoder at competitive quality and latency. More broadly, the results suggest preserving source information and explicit temporal access, then evaluating fast decoders on the generated-latent distribution used at deployment. Appendix E details the present scope and future extensions.

AI USE STATEMENT

Generative AI tools were used for language polishing and LaTeX typesetting. The authors performed the research design, implementation, experiments, analysis, and verification, and retain responsibility for the final content.

REPRODUCIBILITY STATEMENT

Appendix B documents the model, objectives, and decoding procedures; Appendix B.6 fixes populations, preprocessing, interfaces, inference settings, and metrics; and Appendix D specifies downstream and fixed-latent controls. Input identifiers, seeds, checkpoints, and source fingerprints are retained with the archived evaluation records. We plan to release anonymized inference and evaluation code.

REFERENCES

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. MusicLM: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023. URL https://arxiv.org/abs/2301.11325.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. VGGSound: A large-scale audio-visual dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020a. URL https://www.robots.ox.ac.uk/~vgg/publications/2020/Chen20/.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. WaveGrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020b. URL https://arxiv.org/abs/2009.00713.

Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 9550–9575. PMLR, 2024. URL https://proceedings.mlr.press/v235/crowson24a.html.

Michael Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. FMA: A dataset for music analysis. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*, pp. 316–323, 2017. URL https://archives.ismir.net/ismir2017/paper/000075.pdf.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023.

Zach Evans, Julian D. Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. *arXiv preprint arXiv:2407.14358*, 2024.

Zach Evans, Julian D. Parker, Matthew Rice, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio 3. *arXiv preprint arXiv:2605.17991*, 2026. URL https://arxiv.org/abs/2605.17991.

Frederic Font, Gerard Roma, and Xavier Serra. Freesound technical demo. In *Proceedings of the 21st ACM International Conference on Multimedia*, pp. 411–412, 2013. doi: 10.1145/2502081.2502245.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025a.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.02012*, 2025b.

Jonathan Heek, Emiel Hoogeboom, Thomas Mensink, and Tim Salimans. Unified latents (UL): How to train your latents. *arXiv preprint arXiv:2602.17270*, 2026. URL https://arxiv.org/abs/2602.17270.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In *Proceedings of Interspeech*, pp. 2350–2354, 2019. doi: 10.21437/Interspeech.2019-2219.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. AudioCaps: Generating captions for audios in the wild. In *Proceedings of NAACL-HLT*, pp. 119–132, 2019. doi: 10.18653/v1/N19-1011. URL https://aclanthology.org/N19-1011/.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020. URL https://arxiv.org/abs/2009.09761.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in Neural Information Processing Systems*, 2023.

Junhyeok Lee and Seungu Han. NU-Wave: A diffusion probabilistic model for neural audio upsampling. In *Proceedings of Interspeech*, 2021. URL https://www.isca-archive.org/interspeech_2021/lee21c_interspeech.html.

Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. PriorGrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_BNiN4IjC5.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025. URL https://arxiv.org/abs/2511.13720.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D. Plumbley. AudioLDM: Text-to-audio generation with latent diffusion models. In *International Conference on Machine Learning*, 2023.

Ilaria Manco, Benno Weck, SeungHeon Doh, Minz Won, Yixiao Zhang, Dmitry Bogdanov, Yusong Wu, Ke Chen, Philip Tovstogan, Emmanouil Benetos, Elio Quinton, György Fazekas, and Juhan Nam. The Song Describer Dataset: A corpus of audio captions for music-and-language evaluation. *arXiv preprint arXiv:2311.10057*, 2023. URL https://arxiv.org/abs/2311.10057.

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*, 32:3339–3354, 2024. doi: 10.1109/TASLP.2024.3419446. URL https://arxiv.org/abs/2303.17395.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3481–3490. PMLR, 2018. URL https://proceedings.mlr.press/v80/mescheder18a.html.

Tu Anh Nguyen, Wei-Ning Hsu, Antony D'Avirro, Bowen Shi, Itai Gat, Maryam Fazel-Zarani, Tal Remez, Jade Copet, Gabriel Synnaeve, Michael Hassid, Felix Kreuk, Yossi Adi, and Emmanuel Dupoux. EXPRESSO: A benchmark and analysis of discrete expressive speech resynthesis. In *Interspeech 2023*, pp. 4823–4827, 2023. doi: 10.21437/Interspeech.2023-1905. URL https://www.isca-archive.org/interspeech_2023/nguyen23_interspeech.html.

Julian D. Parker, Zach Evans, CJ Carr, Zachary Zukowski, Josiah Taylor, Matthew Rice, and Jordi Pons. SAME: A semantically-aligned music autoencoder. *arXiv preprint arXiv:2605.18613*, 2026.

Marco Pasini, Stefan Lattner, and George Fazekas. Music2latent: Consistency autoencoders for latent audio compression. *arXiv preprint arXiv:2408.06500*, 2024.

Marco Pasini, Stefan Lattner, and George Fazekas. CoDiCodec: Unifying continuous and discrete compressed representations of audio. *arXiv preprint arXiv:2509.09836*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10619–10629, 2022. URL https://arxiv.org/abs/2111.15640.

Flavio Schneider, Ojasv Kamal, Zhijing Jin, and Bernhard Schölkopf. Moûsai: Text-to-music generation with long-context latent diffusion. *arXiv preprint arXiv:2301.11757*, 2023. URL https://arxiv.org/abs/2301.11757.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2011.13456.

Kangdi Wang, Zhiyue Wu, Dinghao Zhou, Rui Lin, Junyu Dai, and Tao Jiang. Back to Ear: Perceptually driven high fidelity music reconstruction. *arXiv preprint arXiv:2509.14912*, 2025.

Kangdi Wang, Yusheng Dai, and Jin Xu. Fourier is frontier: Frequency-aware autoencoding for high-fidelity music reconstruction. *arXiv preprint arXiv:2608.19843*, 2026. URL https://arxiv.org/abs/2608.19843.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. *arXiv preprint arXiv:2211.06687*, 2022. URL https://arxiv.org/abs/2211.06687.

Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92), 2019.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6199–6203, 2020. doi: 10.1109/ICASSP40776.2020.9053795. URL https://arxiv.org/abs/1910.11480.

Feiyan Zhou, Luyuan Wang, Shoufa Chen, Zhe Wang, Zhiheng Liu, Yuren Cong, Xiaohui Zhang, Fanny Yang, and Belinda Zeng. WavFlow: Audio generation in waveform space. *arXiv preprint arXiv:2605.18749*, 2026. URL https://arxiv.org/abs/2605.18749.

A Additional Related Work on Waveform Generation

Waveform generation synthesizes audio from acoustic or semantic conditions. Early diffusion models inject spectrogram guidance layerwise or through multiresolution fusion (Kong et al., 2020; Chen et al., 2020b), while audio upsampling conditions on low-rate waveform observations (Lee & Han, 2021). Audio autoencoding differs in that it reconstructs a long waveform from a compact latent sequence learned jointly with the decoder. This condition must retain source-specific timing rather than only specify a broad acoustic or semantic target.

In fixed-resolution waveform DiTs, finer tokenization lengthens the sequence throughout contextual processing (Peebles & Xie, 2023). Hierarchical architectures reduce this cost through lower-resolution computation (Crowson et al., 2024), but their image-generation formulations do not specify how compact, time-indexed source information should remain available across waveform resolutions. FLORE-DiT combines hierarchical computation with position-specific latent expansion and resolution-aligned conditioning, providing source guidance to both fine-scale synthesis and coarser contextual processing.

Signal scale presents a complementary challenge. Data-dependent priors adapt diffusion noise to conditional acoustic energy (Lee et al., 2022), while direct waveform generation uses clip-level RMS normalization and amplitude scaling (Zhou et al., 2026). FLORE-AE instead predicts a locally normalized carrier and an amplitude envelope from the shared compressed latents, allowing the waveform target to retain local energy variation while recovering source amplitude dynamics.

B Implementation and Evaluation Details

This section records the concrete configurations underlying the formulation in Section 3. FLORE-DiT denotes the clean-carrier predictor, including its latent-conditioning paths and waveform hierarchy. The encoder and amplitude predictor are separate components of FLORE-AE. A training-time KL bottleneck regularizes the latent distribution. Temporal downsampling, latent width, and decoder processing resolution are specified separately.

B.1 Signal factorization and amplitude recovery

The reference envelope uses a 100-ms local energy window, a 1,024-sample hop, and seven-frame smoothing in log space. Writing $\mathcal{E}_w$ for the local mean-square operator and $\mathcal{H}_k$ for the log-envelope smoother,

$$\boldsymbol{\ell} = \mathcal{H}_k\left[\tfrac{1}{2}\log\big(\mathcal{E}_w(\boldsymbol{x}^2) + a_{\min}^2\big)\right], \qquad \boldsymbol{a}(\boldsymbol{x}) = \exp(\mathcal{I}_T\boldsymbol{\ell}), \tag{10}$$

with $a_{\min} = 10^{-3}$. Local energy excludes padded samples from the pooling average; log smoothing uses replicated boundary values. Interpolation is linear with `align_corners=False`. Envelope extraction is detached from the training graph. This parameterization does not apply a clip-global crest multiplier or clip the carrier.

The global amplitude head receives per-channel latent means and log RMS statistics. The local head reads normalized latent features and predicts a residual envelope, whose temporal mean is subtracted. The sum of the global gain and residual is interpolated to waveform length, clipped in log space to $[\ell_{\min}, \ell_{\max}] = [-9, 1]$, and exponentiated:

$$\widehat{\boldsymbol{a}}_\psi(\boldsymbol{z}_c) = \exp\big[\mathrm{clip}\big(\mathcal{I}_T\big(g_\psi(\boldsymbol{z}_c) + \boldsymbol{e}_\psi(\boldsymbol{z}_c)\big), \ell_{\min}, \ell_{\max}\big)\big], \tag{11}$$

where g_ψ is the global log gain and $\boldsymbol{e}_\psi$ is the zero-mean local residual. The target log envelope is also resampled to the latent length for supervision. If μ denotes its temporal mean and $\boldsymbol{\ell}_N$ the resampled target,

$$\mathcal{L}_{\mathrm{amp}} = \mathrm{Huber}_{0.5}(g_\psi, \mu) + \mathrm{Huber}_{0.5}(\boldsymbol{e}_\psi, \boldsymbol{\ell}_N - \mu). \tag{12}$$

Both terms are averaged over their respective elements and carry equal weight before the overall amplitude-loss multiplier.

B.2 Encoder interfaces and FLORE-DiT configurations

The FLORE-AE encoder uses an Oobleck-style convolutional frontend at stride 512, followed by a query resampler that compresses four feature positions into one (Evans et al., 2024; Parker et al., 2026).

The FLORE-AE-C encoder instead uses a query stride of eight. A final projection sets the latent width. Query-based synthesis conditioning uses a learned resampler with 16 learned output positions per latent token and periodic feed-forward activations in its final four layers. These synthesis features are not additional stored latents. Encoder query readout includes an input-dependent aggregation residual: the mapped input is mean-pooled over each compression group and scaled by the square root of the group size before being added to the learned-query output. The query resamplers use Dynamic Tanh for feature and query/key normalization.

Table 5: **Latent and decoder configurations.** R is temporal downsampling, D is latent width, P is waveform projection stride, and L counts factor-two reductions. The rows specify FLORE-AE and FLORE-AE-C, respectively.

R	D	P	L	Input conditioning	Synthesis head
2048	192	128	3	Ordered query expansion	Overlapping
4096	256	256	2	Ordered query expansion	Blockwise Haar

Latent width and downstream generation. Early latent-diffusion systems often kept D small to simplify latent generation, whereas recent evidence indicates that wider latents remain tractable when their representation is suitably structured (Parker et al., 2026). We therefore train each downstream generator directly on its frozen native interface: $D = 192$ for FLORE-AE and $D = 256$ for FLORE-AE-C, without an additional width-reduction layer or auxiliary bottleneck. The sound and music results in Table 2 show that both interfaces support text-conditioned generation, including the 256-channel compact interface. This establishes that the larger width is manageable under the evaluated per-interface generator contract. It is not a width ablation: D changes together with temporal downsampling and autoencoder design, and each interface uses its designated generator, so the comparison does not imply that increasing width alone improves downstream generation.

The input waveform projection has stride P and kernel size $2P$. The fine-local hierarchy processes sequences of lengths $M \to M/2 \to M/4 \to M/8 \to M/4 \to M/2 \to M$, where M is the projected waveform length. Its outer fine-resolution blocks use pre-LayerNorm, a nine-token depthwise convolution, a gated feed-forward branch with hidden-width multiplier two, and a zero-initialized output projection. Figure 3 expands this block. After temporal convolution, the gated update is applied position-wise:

$$\begin{aligned} \widetilde{\boldsymbol{h}} &= \mathrm{DWConv}\big(\mathrm{LayerNorm}(\boldsymbol{h})\big), \\ \boldsymbol{h}'_i &= \boldsymbol{h}_i + W_o\Big[(W_c\widetilde{\boldsymbol{h}}_i) \odot \mathrm{GELU}(W_g\widetilde{\boldsymbol{h}}_i)\Big]. \end{aligned} \tag{13}$$

The two input projections implement GEGLU (Shazeer, 2020); W_o projects the gated features back to the block width. All projections are bias-free, and W_o is initialized to zero so the block starts as an identity. The outer hierarchy skip, distinct from this within-block residual, bypasses the entire downsample–process–upsample path and is fused after the final local block.

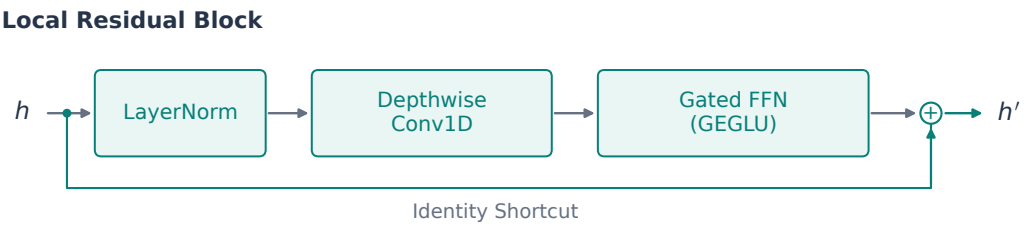


Figure 3: **Local residual block.** Normalized features pass through a depthwise temporal convolution and a gated feed-forward module. The GEGLU gating operation and output projection are grouped as one FFN; the residual adds the original features. This block operates on the finest waveform features on both sides of the outer hierarchy.

The inner hierarchy has learned stride-two convolutions and transposed convolutions, with concatenation-and-projection skip fusions. DiT blocks use per-block time modulation, sliding

self-attention with up to 64 neighbors on each side, global latent cross-attention, local convolution, and feed-forward processing. The attention neighborhood slides with each query; it is not a shifted-window partition. Aligned latent injections have learned channel-wise gates. Their inputs are linearly interpolated versions of the original projected latent context, rather than the query-expanded input features. In FLORE-AE-C, the five inner stages have independent gates, initialized to 0.02, and its 12 inner DiT blocks are allocated as $3/2/2/2/3$ across these stages. Rotary coordinates track the inner hierarchy's downsampling factors; this coordinate rescaling applies to self-attention, not to a separate physical-position encoding of latent cross-attention.

Within a DiT block, time-dependent scale, shift, and gate values are predicted independently for self-attention and the feed-forward branch. Both residual gates use $\mathrm{sigmoid}(1 - g(t))$. The modulation output is zero-initialized, giving an initial gate of $\mathrm{sigmoid}(1)$; the main attention and feed-forward output projections are not zero-initialized. Queries and keys use per-head RMSNorm. The local branch applies LayerNorm, a depthwise convolution, SiLU, and a pointwise projection, with a learned zero-initialized output gate. The main feed-forward branch uses SwiGLU, whereas the outer local block in Figure 3 uses GEGLU. These are implementation choices within the shared architecture, not separate proposed mechanisms.

FLORE-AE uses decoder width 768, eight inner DiT blocks across its three-reduction hierarchy, 12 attention heads, feed-forward multiplier four, and a four-layer synthesis-query resampler. FLORE-AE-C uses the 12-block, five-stage allocation specified above and the two-reduction hierarchy without the additional P128 fine-local wrapper. Its synthesis head predicts normalized blockwise Haar coefficients and applies a fixed four-level inverse transform. The transform changes the output parameterization, not the latent interface.

The capacity study in Appendix D.4 distinguishes two comparisons. Whole-system scaling changes multiple training components together with denoiser capacity while retaining the $R = 2048$, $D = 192$ latent interface. Decoder-only scaling keeps every non-decoder component and the denoiser depth fixed while widening the denoiser from 1024 to 1280 channels. The 768-channel, 8-block Base model is the configuration used by FLORE-AE. These controls use the four-evaluation conditional-flow decoder before EA-MF adaptation; complete configurations and results are in Table 12. Inputs are padded to synthesis alignment and cropped to their original length after decoding.

For projected waveform length $M \simeq T/P$, a stage with cumulative reduction d_s has $M_s \simeq M/d_s$ tokens. A sliding neighborhood of w tokens permits $O(M_s w)$ self-attention pairs, compared with $O(M_s^2)$ for global waveform attention. Its token centers cover approximately wPd_s waveform samples; projection and convolutional receptive fields add further support. Global cross-attention to N latent tokens contributes $O(M_s N)$ pairs. These are interaction counts, not a linear runtime guarantee for the full network; feature projections, channel widths, depths, and kernel implementations also determine cost.

B.3 Training and latent-condition protocols

The clean-carrier objective samples $t = \mathrm{sigmoid}(u)$, $u \sim \mathcal{N}(0, 1)$, clipped to $[\delta, 1-\delta]$ with $\delta = 10^{-3}$. Carrier MSE has uniform weight over these sampled times. Since $f_\theta - \boldsymbol{c} = (1-t)\big(\boldsymbol{v}_\theta - (\boldsymbol{c} - \boldsymbol{\epsilon})\big)$, this corresponds to a $(1-t)^2$ weight in velocity-regression coordinates, not uniform velocity MSE. The auxiliary weights are $\lambda_{\mathrm{spec}} = 0.05$ and $\lambda_{\mathrm{amp}} = 0.1$. The spectral term is evaluated after predicted amplitude restoration.

The encoder defines the latent distribution $q_\phi(\boldsymbol{z}_c \mid \boldsymbol{x})$ used in Section 3.1. We regularize it directly toward a standard normal reference with the KL bottleneck (Kingma & Welling, 2014)

$$\mathcal{R}_{\mathrm{lat}} = \lambda_{\mathrm{KL}}\, \mathbb{E}_{\boldsymbol{x}}[D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_c \mid \boldsymbol{x}) \,\|\, \mathcal{N}(0, I))]\,, \qquad \lambda_{\mathrm{KL}} = 10^{-5}. \tag{14}$$

The encoder receives both this KL term and the full reconstruction gradient. Joint autoencoder training samples $\boldsymbol{z}_c$ from the encoder distribution; reconstruction and subsequent EA-MF adaptation use the deterministic encoder output $\boldsymbol{z} = E_\phi(\boldsymbol{x})$. For downstream generation, per-channel standardization is inverted once, and the selected MeanFlow and compact decoders receive clean predicted latents. Source-derived waveform amplitude is never supplied at inference. The KL bottleneck is a training regularizer and is separate from the independently trained downstream latent generator.

B.4 MULTISTEP INTEGRATION

The unadapted four-evaluation multistep sampler uses the nominal time grid $(0, 0.36, 0.66, 0.94, 1)$ with endpoints clipped to $[\delta, 1-\delta]$. Starting from Gaussian carrier noise, update i uses $q_i = \min(i+1, 3)$ previous velocity evaluations:

$$\boldsymbol{c}_{i+1} = \boldsymbol{c}_i + \sum_{j=0}^{q_i-1} b_{ij} \boldsymbol{v}_\theta(\boldsymbol{c}_{i-j}, t_{i-j}, \boldsymbol{z}_c), \qquad b_{ij} = \int_{t_i}^{t_{i+1}} \ell_{ij}(s)\,\mathrm{d}s, \tag{15}$$

where ℓ_{ij} is the Lagrange basis on the backward nodes $t_i, \ldots, t_{i-q_i+1}$. Euler and second-order startup are followed by two third-order updates. The sampler therefore uses four new network evaluations and no terminal clean-prediction call. Amplitude restoration is applied once to the final integrated carrier. This is a classical variable-step solver with an empirical grid, not a learned solver or a distillation method. Uniform Euler comparisons use the same checkpoint and latent protocol.

B.5 ENDPOINT-AWARE MEANFLOW

The deployed two-evaluation decoder is initialized from the pretrained FLORE-AE decoder. It retains the original encoder and amplitude predictor; both are frozen. Only the waveform backbone and its added interval-conditioning branch are optimized. The adaptation uses clean encoder conditions and the original audio-training mixture.

For $0 \le t \le r \le 1$ with $t < 1$, let $h = r - t$. The interval-conditioned predictor retains the pretrained prediction-to-velocity conversion:

$$u_\vartheta(\boldsymbol{y}, t, r, \boldsymbol{z}_c) = \frac{f_\vartheta(\boldsymbol{y}, t, h, \boldsymbol{z}_c) - \boldsymbol{y}}{1-t}, \tag{16}$$

$$\boldsymbol{y}^+ = \boldsymbol{y} + (r-t) u_\vartheta(\boldsymbol{y}, t, r, \boldsymbol{z}_c). \tag{17}$$

The interval embedding is $A_\vartheta(e(12h)) - A_\vartheta(e(0))$, where e is sinusoidal and A_ϑ is a two-layer network whose final projection is zero-initialized. Its value vanishes exactly at $h = 0$. The shared instantaneous field can change during adaptation; no frozen teacher, trajectory rollout, or additional waveform residual head is used.

Field training uses analytically corrupted carriers from Equation 1, with $\boldsymbol{s} = \boldsymbol{c}$, and the data velocity $\boldsymbol{v}_{\mathrm{data}} = \boldsymbol{c} - \boldsymbol{\epsilon}$. Define $\boldsymbol{v}_{\mathrm{inst}} = u_\vartheta(\boldsymbol{c}_t, t, t, \boldsymbol{z}_c)$ and $\mathcal{D}_t u_\vartheta = \partial_t u_\vartheta + J_{\boldsymbol{y}} u_\vartheta\, \boldsymbol{v}_{\mathrm{inst}}$, holding r fixed. The iMF compound-velocity objective (Geng et al., 2025b), in our noise-to-data convention, is

$$\mathcal{L}_{\mathrm{MF}} = \mathbb{E}[\ell_{\mathrm{norm}}(u_\vartheta - (r-t)\,\mathrm{sg}(\mathcal{D}_t u_\vartheta) - \boldsymbol{v}_{\mathrm{data}})]\,, \tag{18}$$

where the field and its derivative are evaluated at $(\boldsymbol{c}_t, t, r, \boldsymbol{z}_c)$ and sg stops gradients through the derivative. No model-generated states are used for training.

Half of each training batch contains instantaneous pairs $r = t$, with t stratified uniformly over the interior of $[0, 1]$. The other half contains equal proportions of four interval types: general interior intervals, intervals starting at zero, intervals ending at one, and the full interval $[0, 1]$. For general intervals, the length is uniform and the start is uniform over the admissible range. Assignments to audio items are randomly permuted. Interior times and lengths are clipped to $[0.001, 0.999]$; explicit endpoints are retained. The two groups contribute equally to Equation 18.

For residual $\mathbf{q}$, the normalized residual loss is

$$\ell_{\mathrm{norm}}(\mathbf{q}) = \frac{\mathrm{mean}\,\mathbf{q}^2}{\mathrm{sg}(\mathrm{mean}\,\mathbf{q}^2 + 0.01)}. \tag{19}$$

The Jacobian–vector product uses the detached instantaneous prediction as its state direction, a unit start-time direction, and a zero end-time direction. Velocity conversion and differentiation use FP32 arithmetic even with BF16 network execution.

Spectral reconstruction is applied to the instantaneous clean estimate $\boldsymbol{c}_t + (1-t)u(\boldsymbol{c}_t, t, t, \boldsymbol{z}_c)$ and to an independently sampled near-clean estimate $\boldsymbol{c}_t + (1-t)u(\boldsymbol{c}_t, t, r, \boldsymbol{z}_c)$. For this auxiliary batch, starts are split equally between $t = 0.5$ and uniform $[0, 0.75]$, and endpoints between $r = 1$ and uniform $[0.90, 0.99]$, in a balanced cross-product. Inputs are analytically corrupted carriers. The

predicted envelope restores natural amplitude before spectral comparison. For $r < 1$, the clean estimate extrapolates beyond the interval endpoint; this is a perceptual regularizer, not an exact MeanFlow identity.

The auxiliary carrier estimate is defined in Equation 7. With π_{near} denoting the independent auxiliary pair distribution above, the spectral terms in Equation 9 are

$$\begin{aligned} \mathcal{L}_{\text{spec}}^{\text{inst}} &= \mathbb{E}_{r=t}\left[\mathcal{L}_{\text{MRSTFT}}\left(\widehat{\boldsymbol{a}}_\psi(\boldsymbol{z}_c) \odot \widehat{\boldsymbol{c}}_{t,t}^{\text{aux}}, \boldsymbol{x}\right)\right], \\ \mathcal{L}_{\text{spec}}^{\text{near}} &= \mathbb{E}_{(t,r)\sim\pi_{\text{near}}}\left[\mathcal{L}_{\text{MRSTFT}}\left(\widehat{\boldsymbol{a}}_\psi(\boldsymbol{z}_c) \odot \widehat{\boldsymbol{c}}_{t,r}^{\text{aux}}, \boldsymbol{x}\right)\right]. \end{aligned} \tag{20}$$

The near-clean branch uses an independent random stream and adds one auxiliary forward pass without changing the main field-training pairs. Its four equally sized start/end groups are averaged with equal weight. The instantaneous and near-clean spectral weights are $\lambda_{\text{inst}} = 0.016$ and $\lambda_{\text{near}} = 0.062$, respectively. They are fixed before performance evaluation from training-batch gradient calibration; they are not retuned per test metric. Adaptation uses AdamW with learning rate 5×10^{-6} and a global batch of 64. Reproducibility metadata records the base-training checkpoint separately from the selected 60,000-update EMA adaptation checkpoint for FLORE-AE. FLORE-AE-C instead retains its selected 300,000-update EMA checkpoint.

FLORE-AE inference uses $(0, 0.5, 1)$ for two evaluations. One- and four-evaluation sweeps use the same adapted weights with $(0, 1)$ and $(0, 0.25, 0.5, 0.75, 1)$. All comparisons retain the encoder, latent transform, amplitude recovery, and per-input noise seed. FLORE-AE uses its 60,000-update adapted EMA decoder and native two-step sampler. FLORE-AE-C does not use this 60,000-update adaptation; it uses exactly two model calls to its selected 300,000-update EMA checkpoint under the solver configuration specified for that checkpoint, matching Table 1.

B.6 Experimental Setup and Evaluation

B.6.1 Datasets and Evaluation Populations

FLORE-AE and FLORE-AE-C use temporal compression factors of $2048\times$ and $4096\times$, respectively: one latent time step represents 2,048 or 4,096 waveform samples. Both use random 131,072-sample crops (approximately 2.97 s) at 44.1 kHz mono from the same frozen three-source training pool: FMA (Defferrard et al., 2017), Freesound (Font et al., 2013), and WavCaps (Mei et al., 2024). The reconstruction test sets comprise VCTK (Yamagishi et al., 2019) and EXPRESSO (Nguyen et al., 2023) (speech), Song Describer (Manco et al., 2023) and MusicCaps (Agostinelli et al., 2023) (music), and AudioCaps (Kim et al., 2019) and VGGSound (Chen et al., 2020a) (sound). Each model uses the same selected checkpoint and inference configuration across all six datasets.

Table 6 lists the common evaluation populations after input-only quality checks and documented availability exclusions. VCTK uses a fixed speaker-balanced subset: 20 mic2 recordings from each of 108 speakers (2,160 total), selected by a fixed SHA-256 rule using only speaker and recording IDs. It is our reproducible evaluation cohort, not an official test split. The other five datasets retain their complete effective evaluation populations. All ten model rows use identical evaluated IDs in each domain. EXPRESSO follows the official test time windows; AudioCaps is reconstructed once per source recording, not once per caption. Existing model checkpoints are unchanged.

For each VCTK speaker, we hash the UTF-8 salt `FLORE-AE/VCTK-0.92/mic2/speaker-balanced/v1`, speaker ID and recording ID joined by newline separators. The recording ID is the original mic2 filename stem. We retain the 20 lexicographically smallest SHA-256 digests, breaking ties by recording ID. No model scores enter this selection. The frozen ID list and source-manifest hash are retained with the evaluation artifacts.

Input exclusions and source overlap. Before evaluation, all six reconstruction populations are compared with the union of the frozen FMA, Freesound, and WavCaps training manifests using the recorded source identifiers and hashes, including matches exposed across dataset boundaries. The evaluated counts in Table 6 are measured after excluding every match found under this manifest-level rule. The versioned manifest retains candidate IDs, exclusion reasons, and the identifiers used for matching. VGGSound also excludes videos with recorded source IDs shared with the complete AudioCaps test set or MusicCaps. This establishes disjointness under the frozen manifest identifiers; it does not claim exhaustive detection of perceptually similar audio lacking a shared identifier. The training sources need not be mutually disjoint because WavCaps includes Freesound.

Table 6: **Reconstruction datasets.** Candidate gives the initial evaluation population; evaluated gives the population scored in Table 1 after availability checks and removal of identified overlap with the training pool.

Category	Dataset	Reference	Candidate	Evaluated
Speech	VCTK	Yamagishi et al. (2019)	2160	2160
	EXPRESSO	Nguyen et al. (2023)	648	648
Music	Song Describer	Manco et al. (2023)	706	706
	MusicCaps	Agostinelli et al. (2023)	5521	5352
Sound	AudioCaps	Kim et al. (2019)	975	975
	VGGSound	Chen et al. (2020a)	15446	14483

B.6.2 Training and Implementation Details

The training corpus is defined by frozen FMA, Freesound, and WavCaps manifests. Within each pass through the pooled manifest, entries are shuffled and traversed without replacement; the pool is reshuffled after exhaustion, so recordings can recur across optimization while crops are resampled. Both autoencoders use random 131,072-sample crops (approximately 2.97 s); the selected FLORE-AE and FLORE-AE-C checkpoints are trained for 900,000 and 300,000 optimization steps, respectively, on 8 NVIDIA A800 GPUs with global batch size 64 and BF16 mixed precision. We use AdamW with $\beta_1 = 0.8$, $\beta_2 = 0.99$, weight decay 10^{-3}, peak learning rate 10^{-4}, 10,000 warm-up steps, and an inverse learning-rate schedule with decay scale 200,000 and power 0.5. EMA weights are used for evaluation.

FLORE-AE's EA-MF fine-tuning starts from its pretrained autoencoder and updates only the waveform decoder and interval-conditioning parameters; all other modules remain frozen. Fine-tuning uses 60,000 optimization steps, AdamW at 5×10^{-6}, the same crop length and global batch size, and EMA. The instantaneous and endpoint-neighborhood spectral weights are 0.016 and 0.062, respectively. FLORE-AE uses the two-evaluation grid $(0, 0.5, 1)$ at inference. FLORE-AE-C uses exactly two calls to its selected 300,000-update checkpoint under its recorded solver configuration and does not use the 60,000-update FLORE-AE adaptation. Each selected configuration is trained with a single seed.

Main downstream generators use 200,000 optimization steps. Equal update counts do not imply equal compute because latent sequence lengths, widths, and generator interfaces differ. Native interfaces and disclosed baseline training data are retained.

Table 7: **Latent interfaces.** R denotes temporal downsampling and D denotes latent width. Temporal rate and width jointly determine the scalar representation size.

Model	Sample rate (kHz)	R	D	Latent steps/s
Stable Audio Open	44.1	2048	64	21.53
ϵar-VAE	44.1	1024	64	43.07
ϵar-VAE2	48.0	1920	128	25.00
SAME-L	44.1	4096	256	10.77
Moûsai (v2)	48.0	1024	32	46.88
Music2Latent	44.1	4096	64	10.77
FLORE-AE	44.1	2048	192	21.53
FLORE-AE-C	44.1	4096	256	10.77

For each external baseline, the model specification records the released checkpoint, native sampling rate and channel count, latent structure, inference parameters, and disclosed training data. The ϵar-VAE2 row uses the official 48-kHz public-data release, with mono inputs passed through its native stereo interface and outputs returned to the common mono target. It is distinct from the 48-kHz variant of the original ϵar-VAE. The Moûsai comparator is the official ArchiSound `dmae1d-ATC64-v2` diffusion autoencoder, specified at 48 kHz with temporal downsampling 1024 and latent width 32.

We use hash-verified release weights and its native 20-step decoder. The repository's advertised compression factor of 64 counts scalar dimensions and should not be read as temporal downsampling. Music2Latent uses its official 44.1-kHz release with temporal downsampling 4096, latent width 64, and the native one-step consistency decoder. CoDiCodec uses its continuous interface (`discrete=False`, 64 output channels) and parallel consistency decoding configured for two model evaluations; no discrete token indices are used. At width 64, its 128 four-dimensional summary embeddings per 32-frame STFT chunk are reshaped into eight vectors. With hop 1024, this gives a nominal equivalent sequence compression of $4096\times$ and 11.72 vectors/s at the evaluated 48-kHz interface. Summary embeddings are not individually time-aligned; short-input padding changes the realized latent count. Its advertised $128\times$ scalar compression includes two waveform channels and is not the temporal ratio in Table 1. Native latent density is measured from the complete encoded representation, including summary tokens and any side information. For a regular sequence without side information, this is Df_s/R scalars per second. Temporal downsampling and scalar density are not entropy-coded bitrates.

B.6.3 Preprocessing and Reconstruction Inference

All systems receive the same mono source. Stereo-only baselines receive the mono waveform in both channels, and their reconstructed channels are averaged for evaluation. Resampling follows a fixed pipeline whenever a native model rate differs from 44.1 kHz. Padding is removed before scoring; any required chunking or fixed model delay is handled by a model-specific rule established on validation data. Output-dependent alignment or gain fitting is not applied to the spectral metrics. All source recordings are covered in full using 30-second outer windows with one-second linear crossfades; shorter recordings use one pass. Native model padding is removed before scoring. Empty, unreadable, nonfinite and degenerate mono inputs (peak at most 10^{-8}) are excluded independently of model outputs. Quiet, clipped and short but valid inputs are flagged rather than automatically removed; model failures do not remove test items.

FLORE-AE reconstruction uses the frozen base EMA encoder and its selected adapted EMA decoder with clean condition $\boldsymbol{z} = E_\phi(\boldsymbol{x})$. FLORE-AE-C uses its selected encoder–decoder checkpoint and two-evaluation inference. Amplitude is predicted from each variant's latent condition. Controlled objective and architecture comparisons use the matched four-evaluation protocol reported with those tables. Decoder noise is fixed per input under a common seed rule. A fixed SHA-256 function of dataset version, domain, recording ID and window start seeds native encoding and decoding. Only FLORE-AE uses the grid $(0, 0.5, 1)$. The seed rule is fixed across models and evaluation workers. Baselines use their native inference routes. Checkpoint identities and source fingerprints are retained with the result snapshot. Each selected FLORE-AE configuration is represented by one training seed, so the reported comparisons do not quantify training-run variation.

B.6.4 Evaluation Metrics

Objective metrics and statistical analysis. Let $S_q(\boldsymbol{x})$ be the magnitude short-time Fourier transform at resolution q, and let B_q be the fixed, scaled mel projection. The spectral metrics average elementwise absolute differences after a $\log(1+\cdot)$ transform:

$$d_{\mathrm{STFT}}(\boldsymbol{x}, \widehat{\boldsymbol{x}}) = \frac{1}{|\mathcal{Q}_s|} \sum_{q \in \mathcal{Q}_s} \operatorname{mean} \left|\log(1 + S_q(\boldsymbol{x})) - \log(1 + S_q(\widehat{\boldsymbol{x}}))\right|, \tag{21}$$

$$d_{\mathrm{MEL}}(\boldsymbol{x}, \widehat{\boldsymbol{x}}) = \frac{1}{|\mathcal{Q}_m|} \sum_{q \in \mathcal{Q}_m} \operatorname{mean} \left|\log(1 + B_q S_q(\boldsymbol{x})) - \log(1 + B_q S_q(\widehat{\boldsymbol{x}}))\right|. \tag{22}$$

The STFT metric uses six FFT sizes (128, 256, 512, 1024, 2048, and 4096), Hann windows, and 75% overlap (Parker et al., 2026). The mel metric uses 64 bands at FFT sizes $1024, 2048, 4096$, with a $64/F_q$ projection scaling for F_q frequency bins. The shared implementation uses HTK triangular filters without area normalization, frequency limits from zero to Nyquist, centered reflect padding, periodic Hann windows, and unnormalized FFTs. It is a reproduction of the paper definitions with these defaults fixed, rather than a claim of bitwise equivalence to the original evaluator. Historical spectral scores computed with other definitions are not used to populate these columns.

SI-SDR measures error relative to the least-squares projection of the reconstructed waveform onto its reference. As it is invariant to a global gain error, amplitude analysis additionally measures global

log-RMS error and local log-envelope error under the envelope definition in Appendix B.1. Spectral metrics are computed at natural amplitude without reference-based normalization.

Following Stable Audio 3 (Evans et al., 2026), FAD (Kilgour et al., 2019) is computed as the Fréchet distance between audio embeddings from LAION-CLAP (Wu et al., 2022), whose audio representation was found to provide strong perceptual correspondence. We write rFAD when the compared outputs are reconstructions and FAD when they are generated samples; the estimator and embedding space are otherwise the same. Our implementation uses 512-dimensional embeddings from a fixed fusion-enabled official checkpoint, 48-kHz mono audio, a PCM16 round trip, peak normalization to -1 dBFS, and one full-recording embedding per clip for each specified cohort. The main reconstruction table reports the equal-weight mean of the dataset-level AudioCaps and Song Describer estimates; controlled studies that specify an AudioCaps-only cohort retain that narrower contract. This distributional metric is reported separately from paired waveform and spectral errors.

For metric m, category c, and dataset d, the aggregate is

$$\overline{m}_d = \frac{1}{n_d}\sum_{i=1}^{n_d} m_{di}, \qquad \overline{m}_c = \frac{1}{|\mathcal{D}_c|}\sum_{d\in\mathcal{D}_c} \overline{m}_d, \qquad \overline{m} = \frac{1}{3}\sum_{c} \overline{m}_c. \tag{23}$$

Each category contains two datasets, making the overall statistic an equal-weight mean of six dataset means. Metrics retain separate units and are not combined into a composite score. Every compared model uses the same input cohort; missing outputs and invalid inputs are reported explicitly.

Objective tables report point estimates unless an interval is explicitly shown. Decoder-noise variation and variation across independently trained models are not pooled into those point estimates. Distributional metrics use their own evaluation populations; uncertainty from paired spectral errors is not transferred to rFAD or FAD.

Subjective evaluation. Twelve listeners evaluate a fixed panel of 60 source items: 20 each from speech, music, and sound, comprising 10 items from each of the six reconstruction test sets. The item list is selected before reconstruction without reference to model outputs and is shared by every system. We use a balanced incomplete-block assignment at the source level: each source is assigned to five listeners, and each assigned listener rates the outputs of all ten systems for that source. Thus, every item–system pair receives five scores, while each listener rates 25 sources (250 trials) in two sessions containing 12 and 13 sources.

Only listeners who pass a headphone check proceed to a short practice round and the two scoring sessions. Overall audio quality is rated on a five-point scale from 1 (bad) to 5 (excellent), without access to system identity or model metadata. Trial order is independently randomized within each session, and the same rating question is used throughout.

For each system and item, the five listener scores are averaged to obtain an item-level MOS. Table 1 reports the mean and standard deviation of these 60 item-level MOS values. Because the panel contains equal numbers from the three domains and equal numbers from the two datasets within each domain, this aggregation weights every domain and dataset equally. The shared item panel also makes system contrasts paired at the item level. No automatic MOS predictor or source-consistency rating is included in the reported quality MOS.

C Analysis of the Design Principles

The following arguments explain the design choices in Section 3. They distinguish properties of ideal regression and smooth flows from the behavior of the finite, jointly trained model.

C.1 Conditional ambiguity and generative reconstruction

Fix the encoder and its latent channel, and assume finite second moments. Let X and Z_c denote the random waveform and decoder condition, and write $m(Z_c) = \mathbb{E}[X \mid Z_c]$. For any deterministic decoder d, conditional expectation gives the orthogonal decomposition

$$\mathbb{E}\|X - d(Z_c)\|_2^2 = \mathbb{E}\|X - m(Z_c)\|_2^2 + \mathbb{E}\|m(Z_c) - d(Z_c)\|_2^2. \tag{24}$$

Thus the best squared-error prediction is the conditional mean. If the conditional distribution contains different fine waveform realizations, their mean need not be a typical realization. This argument is specific to squared error; it does not characterize the optimum under spectral or adversarial losses.

Conversely, an exact conditional sample X' drawn independently of X given Z_c satisfies

$$\mathbb{E}\|X - X'\|_2^2 = 2\,\mathbb{E}\|X - m(Z_c)\|_2^2. \tag{25}$$

Representing conditional variability therefore does not by itself lower paired distortion or recover source information absent from the latent representation. The learned representation must carry the source-specific information used during conditional synthesis. These identities establish no general quality ordering between decoder families.

With $\boldsymbol{s} = \boldsymbol{c}$ in Equation 1, the MSE-only optimum at any sampled $t < 1$ is $f^* = \mathbb{E}[\boldsymbol{c} \mid \boldsymbol{c}_t, t, \boldsymbol{z}_c]$. Since $\boldsymbol{c} - \boldsymbol{c}_t = (1-t)(\boldsymbol{c} - \boldsymbol{\epsilon})$,

$$\frac{f^*(\boldsymbol{c}_t, t, \boldsymbol{z}_c) - \boldsymbol{c}_t}{1-t} = \mathbb{E}[\boldsymbol{c} - \boldsymbol{\epsilon} \mid \boldsymbol{c}_t, t, \boldsymbol{z}_c]. \tag{26}$$

This is the marginal velocity conditioned on the latent representation, by the flow-matching construction (Lipman et al., 2023). Under the usual flow regularity assumptions, ideal integration transports the Gaussian source along the conditional carrier marginals. It is not direct mean prediction from $\boldsymbol{z}_c$ alone. Positive time-only weighting leaves this unrestricted pointwise optimum unchanged, although it affects finite-model fitting. In particular, the implemented clean-target loss obeys $\|f_\theta - \boldsymbol{c}\|_2^2 = (1-t)^2\|\boldsymbol{v}_\theta - (\boldsymbol{c} - \boldsymbol{\epsilon})\|_2^2$. These identities characterize the ideal carrier-MSE problem, not the complete practical training objective. Numerical integration and endpoint clipping introduce further differences from the ideal construction. Moreover, deterministic amplitude prediction does not model the full joint uncertainty in carrier and envelope given the latent representation.

C.2 Amplitude dependence and reconstruction coordinates

Let $e = |\mathcal{W}|^{-1}\sum_{i\in\mathcal{W}} x_i^2 > 0$ be signal energy in a local window, with unit-variance independent corruption noise. The ratio of signal energy to expected noise energy is

$$\mathrm{SNR}_t(\boldsymbol{x}) = \frac{t^2 e}{(1-t)^2}, \qquad \log \mathrm{SNR}_t(g\boldsymbol{x}) - \log \mathrm{SNR}_t(\boldsymbol{x}) = 2\log g, \quad 0 < t < 1, \quad g > 0. \tag{27}$$

For the envelope in Equation 10, set the numerical floor to zero and assume positive window energies. Linearity of the energy operator and preservation of constants by smoothing and interpolation yield

$$\boldsymbol{a}(g\boldsymbol{x}) = g\boldsymbol{a}(\boldsymbol{x}), \qquad \boldsymbol{c}(g\boldsymbol{x}) = \boldsymbol{c}(\boldsymbol{x}). \tag{28}$$

The implemented positive floor makes this relation approximate when window energy is large relative to the floor squared, for both gains. Time-varying gains need not commute with energy pooling and smoothing. The construction therefore controls scale dependence without promising exact local unit energy or invariance to arbitrary amplitude modulation.

To separate carrier and envelope errors, let $\delta\boldsymbol{c} = \widehat{\boldsymbol{c}} - \boldsymbol{c}$ and $\delta\boldsymbol{a} = \widehat{\boldsymbol{a}} - \boldsymbol{a}$. Then

$$\widehat{\boldsymbol{x}} - \boldsymbol{x} = \boldsymbol{a} \odot \delta\boldsymbol{c} + \boldsymbol{c} \odot \delta\boldsymbol{a} + \delta\boldsymbol{a} \odot \delta\boldsymbol{c}. \tag{29}$$

If $\delta\boldsymbol{a} = 0$, carrier MSE is waveform squared error weighted by a_i^{-2}. When amplitude is predicted, the other two terms also contribute. The decomposition distinguishes carrier and amplitude errors in the synthesized waveform; controlling corruption scale alone does not guarantee accurate amplitude recovery.

C.3 High-noise prediction and temporal conditioning

For a fixed encoder and latent channel, the clean-carrier MSE optimum is the conditional mean predictor:

$$f^*(\boldsymbol{c}_t, t, \boldsymbol{z}_c) = \mathbb{E}[\boldsymbol{c} \mid \boldsymbol{c}_t, t, \boldsymbol{z}_c], \qquad f^*(\boldsymbol{\epsilon}, 0, \boldsymbol{z}_c) = \mathbb{E}[\boldsymbol{c} \mid \boldsymbol{z}_c]. \tag{30}$$

At the ideal noise endpoint, $\boldsymbol{c}_0 = \boldsymbol{\epsilon}$ is independent of $(\boldsymbol{c}, \boldsymbol{z}_c)$, and therefore

$$\mathbb{E}[\boldsymbol{c} \mid \boldsymbol{\epsilon}, \boldsymbol{z}_c] = \mathbb{E}[\boldsymbol{c} \mid \boldsymbol{z}_c], \qquad I(X; \boldsymbol{\epsilon} \mid \boldsymbol{z}_c) = 0. \tag{31}$$

The noisy input supplies no source information at this endpoint. Its source-dependent conditional mean must be obtained from the latent representation. This does not require the latent representation to determine every acoustic detail, and the conditional mean need not be a typical realization. The argument motivates direct access to time-indexed conditions; it does not prescribe a unique conditioning architecture. The implemented schedule clips its endpoints, as specified in Appendix B.3.

Content-based attention and temporal indexing play different roles. Consider a cross-attention readout $\mathcal{A}(Q, K, V) = \mathrm{softmax}(QK^\top/\sqrt{d})V$, where softmax is row-wise. For any permutation matrix Π acting on the context sequence,

$$\mathcal{A}(Q, \Pi K, \Pi V) = \mathrm{softmax}(QK^\top \Pi^\top/\sqrt{d})\Pi V = \mathcal{A}(Q, K, V). \tag{32}$$

When keys and values contain no identifying time coordinates, their ordering is consequently unavailable to this readout. Positional features or aligned paths can supply the missing association. This identity characterizes unindexed content access, not an inability of transformers in general to represent temporal alignment. In FLORE-DiT, ordered synthesis queries and stage-wise resampling provide explicit time-indexed paths; global latent cross-attention supplies complementary content access. Its attention weights are not constrained to a temporal diagonal, and temporal alignment does not denote a semantic embedding alignment loss.

Information and computational access. For a fixed trained model, the synthesis expansion $\mathcal{S}_\theta(Z_c)$ is deterministic. Hence $X \to Z_c \to \mathcal{S}_\theta(Z_c)$ is a Markov chain, and the data-processing inequality gives

$$I\big(X; \mathcal{S}_\theta(Z_c)\big) \le I(X; Z_c). \tag{33}$$

The data-processing inequality is valid in the extended real sense and does not itself require finite mutual information. A finite upper bound follows under additional regularity, for example finite encoder second moment together with a conditional Gaussian covariance uniformly bounded away from zero and integrable log determinant. Reading both the expanded features and the original latent representation also adds no information relative to the original latent representation. The motivation for ordered expansion and repeated conditioning is therefore computational access: different positions and stages can use different functions of the same information. This does not imply that any particular set of conditioning paths is necessary or sufficient for high reconstruction quality.

C.4 Waveform fidelity and local time displacement

Consider a continuous carrier c with finite energy and a square-integrable weak derivative, and let $\mathcal{T}_\delta c(\tau) = c(\tau - \delta)$. With Fourier convention $C(\omega) = \int_{\mathbb{R}} c(\tau) e^{-i\omega\tau}\,\mathrm{d}\tau$, Parseval's identity gives

$$\begin{aligned} \|\mathcal{T}_\delta c - c\|_2^2 &= \frac{1}{2\pi}\int_{\mathbb{R}} 4\sin^2(\omega\delta/2)|C(\omega)|^2\,\mathrm{d}\omega, \\ &= \delta^2\|c'\|_2^2 + o(\delta^2). \end{aligned} \tag{34}$$

The limit follows by dominated convergence from $4\sin^2(\omega\delta/2)/\delta^2 \le \omega^2$ and the finite derivative energy. The argument applies on the full line; a corresponding periodic form avoids boundary terms. For sampled audio it describes a fixed continuous interpolation, rather than an extra transformation in the training pipeline.

At leading order, local time displacement is weighted by derivative energy, so rapidly varying waveforms can incur substantial paired error under small shifts. This sensitivity motivates explicit local waveform processing together with temporal conditioning. It is not a claim that patchification necessarily discards high frequencies: a sufficiently wide, full-rank patch projection can preserve its input samples. The architectural question concerns how a finite model organizes local interactions and wider context. Neither this expansion nor the conditioning identities prove a quality ordering between U-Nets, isotropic DiTs, and FLORE-DiT; Appendix D.4 specifies the matched controls for these design choices.

C.5 Instantaneous and finite-interval motion

Fix the latent condition and consider an interval away from the singular endpoint $t = 1$. Suppose the field $\boldsymbol{v}$ is twice continuously differentiable with bounded derivatives in a neighborhood of its

trajectory. For $h > 0$, the exact interval-average velocity along $\dot{\boldsymbol{y}} = \boldsymbol{v}(\boldsymbol{y}, t)$ is

$$\overline{\boldsymbol{v}}_{t,h} = \frac{1}{h}\int_t^{t+h} \boldsymbol{v}(\boldsymbol{y}(\tau), \tau, \boldsymbol{z}_c)\,\mathrm{d}\tau, \qquad \boldsymbol{y}(t+h) = \boldsymbol{y}(t) + h\overline{\boldsymbol{v}}_{t,h}. \tag{35}$$

Taylor expansion along this trajectory gives

$$\begin{aligned} \boldsymbol{y}(t+h) &= \boldsymbol{y}(t) + h\boldsymbol{v} + \frac{h^2}{2}\big(\partial_t \boldsymbol{v} + J_{\boldsymbol{y}}\boldsymbol{v}\,\boldsymbol{v}\big) + O(h^3), \\ \overline{\boldsymbol{v}}_{t,h} &= \boldsymbol{v} + \frac{h}{2}\big(\partial_t \boldsymbol{v} + J_{\boldsymbol{y}}\boldsymbol{v}\,\boldsymbol{v}\big) + O(h^2), \end{aligned} \tag{36}$$

where the field and its derivatives on the right are evaluated at $(\boldsymbol{y}(t), t)$, and $J_{\boldsymbol{y}}\boldsymbol{v}$ is the state Jacobian. The leading correction depends on motion along the trajectory, even if individual training interpolation paths are straight. This expansion motivates predicting interval-average motion instead of using one instantaneous velocity for a long update. It does not assert that the learned map has a particular numerical convergence order.

Holding the terminal time r fixed and differentiating the interval average along a trajectory yields

$$(\partial_t + \boldsymbol{v}\cdot\nabla_{\boldsymbol{y}})u = \frac{u - \boldsymbol{v}}{r - t}, \qquad \boldsymbol{v} = u - (r-t)(\partial_t + \boldsymbol{v}\cdot\nabla_{\boldsymbol{y}})u. \tag{37}$$

This is the noise-to-data form of the MeanFlow identity (Geng et al., 2025a;b). Equation 18 uses a learned instantaneous field for the state direction and stops gradients through the derivative. Auxiliary spectral terms and adaptive weighting influence the trained solution; the identity alone is not a guarantee of exact conditional sampling or improved reconstruction.

The interval-state update and the auxiliary carrier estimate have different endpoints. Define $\widehat{\boldsymbol{c}}_r = \boldsymbol{c}_t + (r-t)u_\vartheta(\boldsymbol{c}_t, t, r, \boldsymbol{z}_c)$. Equation 7 then gives

$$\widehat{\boldsymbol{c}}^{\,\mathrm{aux}}_{t,r} - \widehat{\boldsymbol{c}}_r = (1-r)u_\vartheta(\boldsymbol{c}_t, t, r, \boldsymbol{z}_c). \tag{38}$$

For $r = 1$, the auxiliary estimate equals the finite-interval endpoint prediction. For $t < r < 1$, it extrapolates the estimated average motion beyond the interval. Restricting r to a near-clean range limits the extrapolation duration, but does not establish an error bound or make this estimate an exact sample from the learned flow. For $r = t$, the same conversion recovers the instantaneous clean prediction used by the pretrained parameterization. These relations justify distinguishing the two auxiliary prediction regimes; their spectral supervision is an additional reconstruction regularizer, not part of the exact MeanFlow identity.

D Additional Experimental Evidence

D.1 Generation and Editing Evaluation Protocol

The main-text comparison evaluates sound generation on AudioCaps and music generation on Song Describer. Each row uses the designated text-conditioned generator for the autoencoder's frozen latent interface. Within each dataset, the prompt population, waveform export, sampling settings, FAD implementation, and CLAP implementation are fixed. Generator architecture is not treated as a controlled component, so Table 2 supports system-level comparisons rather than a causal claim about the autoencoder alone.

Latents are standardized with fixed training-set per-channel statistics and inverted once before decoding. Main text-to-audio generation uses the selected two-evaluation FLORE-AE and FLORE-AE-C decoders from Table 1; external systems retain their native decoders. AudioCaps editing keeps the same frozen latent interfaces but uses four decoder evaluations for both FLORE variants under the masked inference protocol defined in the main text.

Generation and editing use the CLAP-based FAD defined in Appendix B.6.4; scores from other embedding spaces are not pooled into this comparison. CLAP measures caption–audio similarity. The reported results use fixed dataset-specific test populations and one selected generator training run per interface, so they establish system comparisons but do not quantify generator-training variation.

Table 8: **Complete single-region inpainting and continuation comparison on AudioCaps.** Each autoencoder uses its designated AudioCaps generator. FAD uses CLAP embeddings; Ctx. denotes the unedited context; Cons. is CLAP similarity between generated and original edited regions.

Task	Autoencoder	$\text{FAD}_{\text{full}}\downarrow$	$\text{FAD}_{\text{edit}}\downarrow$	Ctx. SI-SDR↑	Ctx. STFT↓	$\text{CLAP}_{\text{text}}\uparrow$	$\text{CLAP}_{\text{cons.}}\uparrow$
Single	ϵar-VAE	0.115	0.130	7.07	0.086	0.145	0.552
	ϵar-VAE2	0.058	0.100	-1.00	0.112	**0.167**	**0.555**
	Stable Audio Open	0.181	0.141	0.20	0.093	0.140	0.545
	SAME-L	0.150	0.154	7.18	0.093	0.125	0.541
	Moûsai	0.286	0.230	-30.81	0.141	0.097	0.546
	CoDiCodec	0.084	0.189	-6.33	0.100	0.062	0.531
	FLORE-AE	**0.046**	**0.084**	**9.36**	**0.081**	0.145	**0.555**
	FLORE-AE-C	0.072	0.109	7.45	0.089	0.135	0.549
Continuation	ϵar-VAE	0.126	0.149	6.85	0.098	**0.282**	0.591
	ϵar-VAE2	0.087	0.107	-1.16	0.122	0.267	0.589
	Stable Audio Open	0.194	0.195	-0.13	0.103	0.269	0.563
	SAME-L	0.184	0.232	7.04	0.110	0.238	0.534
	Moûsai	0.264	0.250	-27.44	0.142	0.245	0.546
	CoDiCodec	0.156	0.209	-6.69	0.105	0.227	0.543
	FLORE-AE	**0.075**	**0.099**	**9.46**	**0.096**	0.278	**0.595**
	FLORE-AE-C	0.109	0.135	7.43	0.102	0.268	0.581

D.2 Complete audio editing comparison

The complete metrics qualify the aggregate result in Table 3. FLORE-AE gives the lowest full-output and edited-region FAD and the strongest context fidelity in both tasks. External baselines retain the highest text-alignment point estimate for single-region inpainting, while FLORE-AE ties the best generated–original consistency estimate there. The result therefore supports output fidelity and context preservation rather than uniform dominance on every semantic metric.

D.3 Decoder adaptation on fixed generated latents

Table 4(c) uses the frozen AudioCaps latent generator and test population associated with the FLORE-AE row in Table 2. Identical cached latent predictions serve every decoder row. Per-item initial noise, inverse standardization, amplitude recovery, and waveform export are held fixed. Conditional Flow Matching uses the selected four-evaluation base decoder; both MeanFlow objectives use two evaluations and a common 60k adaptation budget with the same frozen encoder. The EA-MF row uses the selected FLORE-AE decoder and reconstruction population reported in Table 1; the CFM and iMF rows are matched adaptations from the same base checkpoint.

Table 9: **Paired reconstruction under matched decoder adaptations.** Fidelity is the six-test-set macro average. The encoder is fixed for every row; iMF denotes Improved MeanFlow.

Decoder	NFE	SI-SDR↑	STFT↓	MEL↓
Conditional Flow Matching	4	14.07	0.051	0.034
iMF + terminal reconstruction	2	**14.43**	**0.050**	**0.032**
EA-MF	2	14.11	0.052	0.034

The two controls expose a deployment-relevant trade-off. Terminal reconstruction gives the strongest paired reconstruction among the adapted decoders, but EA-MF gives the best FAD on the fixed generated latents. Paired reconstruction alone therefore does not determine behavior at the generated-latent interface. This conclusion is limited to the matched generator, cached population, and adaptation setting reported here. The generated-latent substitution values are reported in Table 4(c), while Table 9 provides the complementary paired-reconstruction metrics.

D.4 Controlled Evidence for FLORE-DiT

D.4.1 Controlled Comparison Protocol

The architecture study asks two separate questions motivated in Section 3.3: whether explicit source-time access helps a compressed condition guide dense waveform synthesis, and whether local and multiresolution computation preserve both paired fidelity and distributional quality. All reported variants are complete autoencoders trained jointly from scratch. They share the encoder architecture, latent interface, carrier and amplitude parameterization, initialization rule, data order, objective, and inference budget. Encoder weights are not frozen and may co-adapt with the decoder.

All values in this appendix use the common 200k-update EMA checkpoint, clean latent conditions, and fixed four-evaluation decoding. SI-SDR, STFT, and mel are macro-averaged over the same six reconstruction test sets; rFAD uses the fixed 2,000-item AudioCaps reconstruction population. Matching steps and audio exposure does not match decoder parameter count, total training FLOPs, or inference cost. The results therefore support comparisons among the measured systems, not a universal or equal-compute ordering of backbone families. In particular, removing the hierarchy moves transformer processing to the highest-resolution waveform grid; it is not a module removal that preserves the resolution schedule or computational allocation.

The waveform U-Net retains its native convolutional hierarchy and multiscale conditioning. Its transfer variant adds FLORE's learned ordered expansion and direct source injection. The single-resolution DiT retains ordered latent cross-attention and temporal position information; the aligned DiT adds the explicit paths in Equation 5. The hierarchical DiTs add downsampling, upsampling, and skip connections while retaining the same latent-access configuration.

D.4.2 Evidence for Time-Aligned Latent Access

Table 10: **Conditioning–hierarchy factorial control.** All cells retain ordered latent cross-attention. A adds learned expansion and stage-wise aligned injection; H adds hierarchical waveform computation and changes the waveform-grid schedule. Values use the common 200k EMA, clean-condition, four-evaluation protocol. Inner-DiT TFLOPs are analytical forward-pass costs for one 3-s decoder call under the common counting rule defined below.

DiT variant	A	H	Waveform grids	Inner-DiT TFLOPs	SI-SDR ↑	STFT ↓	MEL ↓	rFAD ↓
Single resolution	×	×	Highest only	0.309	10.07	0.057	0.043	**0.108**
Single resolution, aligned	✓	×	Highest only	0.309	**13.33**	**0.053**	**0.037**	0.116
Hierarchical	×	✓	Lower hierarchy	0.102	12.34	0.054	0.038	0.128
Hierarchical, aligned	✓	✓	Lower hierarchy	0.102	13.23	0.053	0.037	0.130

For the analytical count, a 3-s input is padded to $M = 1{,}040$ projected waveform tokens and has $N = 65$ latent tokens. The inner blocks use width 768; global attention uses the full query sequence and sliding attention uses up to 129 keys per query. We count linear projections, attention matrix products, latent cross-attention, and SwiGLU layers with two FLOPs per multiply–add. The highest-resolution global and sliding stacks require 0.309 and 0.274 TFLOPs per call, respectively; their lower-resolution counterparts require 0.102 and 0.098 TFLOPs. These figures intentionally exclude the conditioning resampler, stage transitions, input/output projections, and outer fine-local path, so they compare the resolution-dependent inner DiT rather than claiming complete decoder FLOPs.

Table 10 isolates the aligned interface from the resolution hierarchy. Alignment improves SI-SDR by 3.27 dB in the single-resolution DiT and by 0.90 dB in the hierarchical DiT, while reducing both spectral errors in both settings. The main-table transfer control shows the same paired-fidelity direction in a different backbone: adding the aligned interface to the waveform U-Net raises SI-SDR from 6.67 to 7.65 dB and lowers STFT/mel from 0.122/0.148 to 0.110/0.133. In all three contrasts, rFAD is unchanged or worse. The evidence therefore supports aligned access as a source-fidelity mechanism, rather than as a generic improvement in distributional similarity.

The cross-resolution contrasts require a different interpretation. The single-resolution systems place their transformer blocks on the highest grid, whereas the hierarchical systems place them on the lower-resolution downsampling–upsampling path. Their differences therefore combine resolution, context span, and compute allocation. The factorial study cleanly tests alignment within each schedule; it does not estimate the marginal value of adding lower-resolution context while holding high-resolution computation fixed.

D.4.3 Evidence for Individual Paths and Local Modeling

Table 11: **Direct FLORE-DiT component controls.** Each row is a jointly trained end-to-end system evaluated at the common 200k EMA checkpoint with clean conditions and four decoder evaluations. Fixed expansion also changes parameter count. Removing the hierarchy reallocates the inner blocks to the highest-resolution grid. Inner-DiT TFLOPs follow the same 3-s analytical count used above.

Decoder	Waveform grids	Inner-DiT TFLOPs	SI-SDR↑	STFT↓	MEL↓	rFAD↓
FLORE-DiT	Highest + lower	0.098	13.45	0.052	0.036	0.110
Without learned expansion	Highest + lower	0.098	12.99	0.053	0.037	**0.107**
Without stage-wise injection	Highest + lower	0.098	**13.49**	**0.052**	0.037	0.118
Without fine-resolution temporal mixing	Highest + lower	0.098	13.44	0.052	0.037	0.113
Without resolution hierarchy	Highest only	0.274	13.42	0.053	**0.036**	0.112

The direct controls qualify the aggregate alignment result. Replacing the learned ordered expansion reduces SI-SDR by 0.46 dB and worsens both spectral errors, although it slightly improves rFAD. Removing stage-wise injection leaves SI-SDR and STFT essentially unchanged while worsening mel and rFAD. Thus the experiment supports the learned fine-grid expansion most clearly; it does not establish that every aligned path is individually necessary.

The strongest local-computation evidence is the main-table comparison between the aligned flat DiT and its sliding-attention variant: SI-SDR improves from 13.33 to 13.46 dB, mel from 0.037 to 0.035, and rFAD from 0.116 to 0.109. Removing only the outer fine-resolution temporal-mixing block produces much smaller changes, so the data do not justify assigning the full local benefit to that block. The no-hierarchy row is an all-highest-resolution alternative, not FLORE-DiT minus a cost-free context module. Its small, mixed metric differences show that the combined high–low design preserves the quality of the all-high-resolution control while reducing inner-DiT cost from 0.274 to 0.098 TFLOPs per call; they do not show that the lower-resolution path is harmful. Together, these controls support the paper's narrow architectural conclusion: explicit source-time access provides the large paired-fidelity gain, and FLORE-DiT retains high-resolution local synthesis while adding lower-resolution context. They do not establish a compute-matched advantage.

D.4.4 Capacity Sensitivity

Table 12 reports a capacity study for the conditional-flow decoder. All three systems use the same $R = 2048$, $D = 192$ latent interface, data mixture, objective, optimizer and scheduler, random seed, global batch of 64, common 10,670-item evaluation cohort, and four-evaluation decoding. We report one selected EMA artifact per scale with complete six-set metrics. Base is the 768-channel, 8-block, 12-head denoiser used by FLORE-AE. Medium jointly scales the encoder, training system, synthesis conditioning, and denoiser; its denoiser has width 1024, 12 blocks, and 16 heads. Large retains the complete Medium encoder, latent interface, non-decoder training components, synthesis conditioning, and 12-block schedule, changing only denoiser width from 1024 to 1280 and heads from 16 to 20 so that the 64-dimensional head width is unchanged. Decoder parameter counts are the non-EMA parameters of the loaded decoder module; they exclude the encoder, training-only auxiliary modules, and EMA copies.

Table 12: **Capacity sensitivity of the waveform diffusion decoder.** Each scale uses one selected EMA artifact on the same 10,670-item, six-set reconstruction cohort with four decoder evaluations. The scale column gives denoiser width/depth and decoder parameters. Macro rows weight the six dataset means equally. Base-to-Medium is whole-system scaling; Medium-to-Large is decoder-only widening. Shading marks the Base configuration used by FLORE-AE; bold marks the best result.

Scale	Test set	SI-SDR ↑	STFT ↓	MEL ↓
Base (used) 768/8; 194.7M	VCTK	18.01	0.0152	0.0118
	EXPRESSO	15.47	0.0250	0.0185
	Song Describer	16.15	0.0662	0.0411
	MusicCaps	13.83	0.0775	0.0510
	AudioCaps	9.36	0.0641	0.0420
	VGGSound	11.80	0.0548	0.0373
	Macro	14.10	0.0505	0.0336
Medium 1024/12; 468.6M	VCTK	17.98	0.0152	0.0119
	EXPRESSO	15.39	0.0249	0.0187
	Song Describer	16.12	0.0655	0.0410
	MusicCaps	13.73	0.0773	0.0514
	AudioCaps	9.21	0.0643	0.0430
	VGGSound	11.67	0.0549	0.0380
	Macro	14.02	0.0503	0.0340
Large 1280/12; 707.2M	VCTK	**18.32**	**0.0147**	**0.0114**
	EXPRESSO	**15.72**	**0.0243**	**0.0179**
	Song Describer	**16.42**	**0.0647**	**0.0399**
	MusicCaps	**14.05**	**0.0762**	**0.0500**
	AudioCaps	**9.54**	**0.0637**	**0.0414**
	VGGSound	**12.00**	**0.0542**	**0.0366**
	Macro	**14.34**	**0.0496**	**0.0329**

The selected Large artifact is best on all three metrics for every test set. Relative to Base, it raises macro SI-SDR from 14.10 to 14.34 dB and reduces STFT/mel from 0.0505/0.0336 to 0.0496/0.0329. Decoder-only widening from Medium to Large therefore confirms that additional capacity can improve quality within the shared encoder contract. The gains over Base nevertheless remain modest relative to the 3.6-fold decoder-size increase, motivating the highlighted Base configuration as the quality–efficiency operating point used in FLORE-AE. Because these are single-seed, scale-specific selected artifacts with different training compute, the result is a capacity sensitivity analysis rather than a statistical scaling law or an equal-compute comparison.

D.5 Controlled Evidence for Signal Factorization

The signal-parameterization study asks whether the reconstruction gain comes from the local carrier–envelope coordinate or can be reproduced by simpler scale handling. Natural-waveform prediction fixes the envelope to one and removes its predictor. Global normalization uses one clip-level RMS target and a latent-conditioned gain predictor. Local factorization uses the envelope in Appendix B.1. The inverse-energy control retains the fixed-scale target and instead reweights its loss toward low-energy regions.

All trained variants share the encoder architecture, latent interface, initialization rule, KL bottleneck, training mixture, data order, noise-time sampling, 200k-update EMA budget, and four-evaluation decoder. Every row is trained jointly from scratch, so the measured effect includes representation co-adaptation rather than holding the encoder weights fixed. SI-SDR, STFT, and mel first average within each test set and then average the six test sets equally. rFAD uses the fixed AudioCaps reconstruction cohort.

Table 13: **Matched signal-parameterization controls.** All variants use the same six-test-set reconstruction panel, training exposure, and inference budget.

Variant	SI-SDR ↑	$\text{STFT}_{\log 1p}$ ↓	$\text{MEL}_{\log 1p}$ ↓	rFAD ↓
Fixed-scale waveform	12.32	0.056	0.041	0.344
Global RMS + predicted gain	11.95	0.055	0.044	0.149
Local carrier + predicted envelope	**13.45**	**0.052**	**0.036**	**0.110**
Fixed-scale + unit-mean local loss reweighting	11.96	0.056	0.040	0.345

Local factorization is the only tested intervention that improves all four criteria over fixed-scale waveform prediction. Clip-level RMS normalization substantially improves rFAD and modestly improves STFT, but reduces SI-SDR and worsens mel. Inverse-energy reweighting remains near the fixed-scale control and does not reproduce the gain. The controlled contrast therefore supports local scale factorization as a signal-conditioning intervention; it does not show that generic emphasis on low-energy regions is sufficient or that the amplitude predictor alone causes the complete system-level change.

D.6 Inference Scope and Deployed Model Sizes

Table 14: **Deployed model parameters.** Counts in millions are computed from loaded modules, including frozen parameters. Shared/interface parameters are listed separately; training-only auxiliary modules and discriminators are excluded from the deployed total.

Model	Encoder	Decoder	Shared/interface	Total
Stable Audio Open	78.0	78.1	0.0	156.1
ϵar-VAE	73.8	74.0	0.0	147.8
ϵar-VAE2	35.3	143.0	0.0	178.4
SAME-S	53.7	54.5	0.0	108.2
SAME-L	426.1	426.1	0.0	852.1
Moûsai (v2)	24.4	160.7	0.0	185.1
Music2Latent	16.3	41.9	0.0	58.2
CoDiCodec (continuous)	42.2	106.3	0.0	148.5
FLORE-AE	63.7	196.0	0.0	259.7
FLORE-AE-C	140.5	194.8	0.0	335.3

Table 14 reports deployed parameter counts from the loaded inference modules. The RTF values in Table 1 use 30-second, batch-one FP32 decoding on the same A800-SXM4-80GB GPU. Each evaluated model is measured after ten warm-up calls over thirty CUDA-synchronized invocations, with TF32, autocast, and cuDNN autotuning disabled. Decoding starts from device-resident cached latents and includes condition expansion, native sampling steps, amplitude recovery, output resampling, and cropping, but excludes encoding, weight loading, disk access, latent generation, guidance branches, inverse standardization, and metric extraction.

RTF divides decoder time by the original unpadded duration. It is a hardware-, duration-, precision-, and implementation-specific operating point, not a measure of streaming first-output latency, complete text-to-audio latency, or equal-compute training. Training-only auxiliary modules and discriminators are excluded from deployed parameter totals. The selected two-evaluation, 335.3M-parameter FLORE-AE-C checkpoint is the compact reconstruction model used in Table 1.

D.7 Complementary Single-Dataset and Single-Source Studies

D.7.1 AudioCaps-Only Reconstruction

Table 15 retains an earlier comparison on the same 2,000 AudioCaps inputs. It uses the original Mel-distance and STFT-distance implementations, not the log1p definitions of the main benchmark. FLORE-AE uses its 260k EMA checkpoint and FLORE-AE-C its 300k EMA checkpoint, with clean encoder conditions and four-evaluation Adams–Bashforth decoding. External systems use their native released interfaces under the common mono-input evaluation. This is a comparison of the recorded systems, not a matched-training-budget ablation.

FLORE-AE improves Mel-distance and SI-SDR over Stable Audio Open in this study, while Stable Audio Open has lower STFT-distance. The spectral and waveform rankings therefore do not coincide. These observations motivate complementary fidelity measurements and broader evaluation populations; changes between this table and the main log1p table cannot be interpreted as training gains because the spectral definitions differ.

Table 15: **AudioCaps-only reconstruction under the original metric protocol.** Mel-distance and STFT-distance retain their original definitions and are distinct from the log1p metrics in Table 1. Bold numbers identify the best reported value in each metric.

Autoencoder	Checkpoint	Mel-dist. ↓	STFT-dist. ↓	SI-SDR ↑
Stable Audio Open	Release	0.844	**1.217**	-0.35
SAME-L	Release	0.883	1.497	7.03
FLORE-AE	260k	**0.645**	1.313	**9.14**
FLORE-AE-C	300k	0.754	1.387	6.76

D.7.2 Single-Source Generator-Training Dynamics

This diagnostic uses 45,044 AudioCaps training items and nominal ten-second waveforms. Each autoencoder is frozen, and its latents are modeled by a text-conditioned generator initialized from the same Qwen3-1.7B backbone. Training uses global batch size 64, seed 4396, a learning rate of 10^{-4} with 2,000 warm-up updates, and a 200k-update budget. Native latent dimensions and sequence lengths differ, so equal updates match audio exposure rather than training compute. The term single-source refers to generator training on AudioCaps; the frozen autoencoders are not claimed to have been trained on this source alone.

Table 16: **Single-source text-to-audio study.** Generators are trained only on AudioCaps. Columns compare common generator checkpoints; bold numbers identify the best value within each checkpoint and metric.

Frozen autoencoder	100k generator updates			200k generator updates		
	FD-PaSST ↓	KL-PaSST ↓	CLAP ↑	FD-PaSST ↓	KL-PaSST ↓	CLAP ↑
Stable Audio Open	**173.46**	**1.115**	**0.373**	231.30	**1.112**	0.341
SAME-L	212.89	1.283	0.346	201.22	1.224	0.360
FLORE-AE	197.64	1.319	0.369	**182.17**	1.203	**0.386**
FLORE-AE-C	200.63	1.274	0.369	284.18	1.257	0.329

The evaluations use generator EMA weights, 24 latent-field evaluations, classifier-free guidance of 5 with empty-caption conditioning, and the fixed guidance-norm rescaling. Each model reverses its own training-set per-channel standardization once. FLORE-AE and FLORE-AE-C use their 300k EMA autoencoders, one analytic latent-endpoint perturbation, and the four-evaluation Adams–Bashforth decoder. Stable Audio Open and SAME-L retain their native released decoders. The outputs in this study use peak-normalized PCM16 export. FD-PaSST and KL-PaSST use the same 4,395 reference-audio IDs for every model; LAION-CLAP similarity uses all 4,875 caption IDs. This generation population is separate from the 2,000-ID reconstruction FD population.

Figure 4 shows that model ranking depends on training stage. Stable Audio Open has the lowest FD-PaSST at 100k, whereas FLORE-AE has the lowest FD-PaSST and highest CLAP similarity at 200k; Stable Audio Open retains the lowest KL-PaSST at that checkpoint. FLORE-AE-C reaches a lower FD-PaSST earlier in training and deteriorates at later checkpoints under this single-source interface. Thus neither reconstruction ranking nor a single terminal generator checkpoint suffices to characterize downstream utility.

The table reports common 100k and 200k checkpoints, and the figure retains the full common curve. No row combines different checkpoints' best metric values. This diagnostic uses one generator family, one training source, and one training seed. It does not establish mixed-data generation, cross-generator transfer, or an intrinsic limitation of a latent representation. Subsequent normalization, guidance, and endpoint variants belong to separate interface comparisons and are not mixed into these curves.

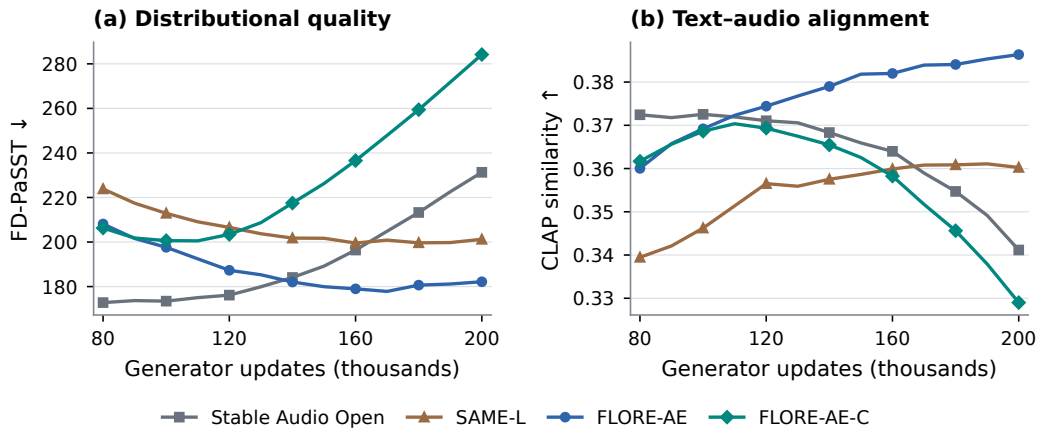


Figure 4: **Single-source generation curves.** All common evaluation checkpoints from 80k to 200k generator updates are shown. Autoencoders and inference settings are fixed within each curve. Rankings vary across training: FLORE-AE improves through 200k updates, whereas FLORE-AE-C peaks earlier under this single-source, single-seed contract.

E Limitations and Future Work

Although FLORE-AE is motivated by universal audio reconstruction and generation, the present study does not yet realize or validate that goal at full scale. Training uses a fixed mixture of three audio sources, and downstream generation is evaluated under dataset-specific, fixed-duration protocols. The evidence therefore spans speech, music, and sound, but should not be interpreted as establishing universality across arbitrary data scales, domains, or durations. Moreover, this work focuses on establishing waveform diffusion decoding as a new autoencoding route; it does not introduce corresponding advances to the encoder architecture or the standard KL bottleneck.

Future work will scale training and validation to substantially larger and broader audio collections, extend downstream generators to variable-length and long-form generation, and jointly develop improved encoder architectures and bottleneck objectives with the diffusion decoder. These extensions target the breadth required by the universal-audio goal while building on the controlled reconstruction, alignment, compute-allocation, and fixed-latent evidence established here.

F Ethics Statement

The only human-participant component is the MOS listening study described in Appendix B.6.4. Participation was voluntary and limited to perceptual quality ratings of a fixed, balanced panel of 60 source items. Each source was assigned to five of the 12 listeners, and every assigned listener rated all ten system outputs for that source in two sessions. Listeners passed a headphone check, system identities were hidden, and presentation order was randomized. We collected only five-point quality scores: no names, contact details, demographic attributes, listener audio, or other personally identifying information were collected. Ratings are reported only as aggregate statistics, and no participant-level results are reported or released. The study therefore makes no claims about listener subgroups or individual speakers.