LATENT SPECTRAL EQUIVARIANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Scalable audio generation systems typically cascade a pre-trained variational autoencoder (VAE) for audio compression and a latent diffusion model for distribution modeling. However, existing VAEs prioritize reconstruction fidelity without explicitly considering how readily the resulting latent distribution can be modeled by diffusion, potentially limiting generation quality. Recent work highlights the importance of aligning latent representations with the structure of the underlying data for diffusion modeling. Building on this direction, we introduce Latent Spectral Equivariance (LATENTEQ), the first framework that systematically introduce equivariance into audio VAE optimization by pairing latent transformations with corresponding spectrogram transformations. Specifically, temporal scale equivariance applies matched low-pass filtering along time, aligning slowly varying latent components with slowly varying acoustic structure. Spectral granularity equivariance pairs latent channel prefixes with frequency-smoothed spectrograms, organizing channels to progressively capture spectral envelopes and fine harmonic detail. We instantiate LATENTEQ on the VAEs of two well-known audio generation systems, Stable Audio Open and MMAudio, through a fine-tuning process without changing their architecture or introducing additional objectives. Comprehensive results demonstrate improved generation quality across waveform- and spectrogram-based VAEs and both text-to-audio and video-to-audio tasks, highlighting LATENTEQ as an effective principle to learn structured audio latents.

1 INTRODUCTION

Latent diffusion models (LDMs) have become the prevailing framework for audio synthesis, achieving strong results in text-to-audio (T2A) (Liu et al., 2023; 2024a; Huang et al., 2023b; Majumder et al., 2024) and video-to-audio (V2A) (Cheng et al., 2025a; Dai et al., 2026) generation. These models are trained in two stages. A variational autoencoder (VAE) (Kingma & Welling, 2013) is first trained to compress waveforms or spectrograms into a low-frame-rate latent sequence, and a diffusion model, typically a diffusion transformer (DiT) (Peebles & Xie, 2023), then learns the distribution of these latents with the VAE frozen, so that generation quality depends on both stages. Recent work focuses on the diffusion model, improving its conditioning (Cheng et al., 2025a; Tian et al., 2025), training objective (Hung et al., 2024; Li et al., 2026), architecture (Huang et al., 2023a; Hai et al., 2024), and sampling (Liu et al., 2024b; 2025b), or extending it to new target domains (Jiang et al., 2025; Jung et al., 2025). The VAE, in contrast, is rarely tailored to the audio diffusion model.

Early systems such as AudioLDM (Liu et al., 2023; 2024a) built their VAEs on mel spectrograms, following image LDMs (Rombach et al., 2022) in compressing them with 2D convolutions and recovering the waveform with a separate vocoder. Neural audio codecs (Défossez et al., 2022; Kumar et al., 2023), developed in parallel, compress the waveform directly into discrete tokens with a 1D convolutional encoder and decoder, enabling language-model-based audio generation (Kreuk et al., 2022; Copet et al., 2023). Stable Audio (Evans et al., 2024a;b; 2025) then combined this 1D convolutional design with a continuous VAE that compresses the waveform into a low-frame-rate latent sequence and decodes it back directly, which removes the vocoder and the information loss of spectrogram inversion. Despite these architectural changes, continuous audio VAEs are still trained for reconstruction alone, with spectral, adversarial, and KL terms, none of which accounts for how easily the diffusion model can learn the resulting latent. Since the VAE fixes the distribution that the diffusion model must learn, this limits generation quality. Accounting for the diffusion model when training the VAE is therefore a natural way to improve it.

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.

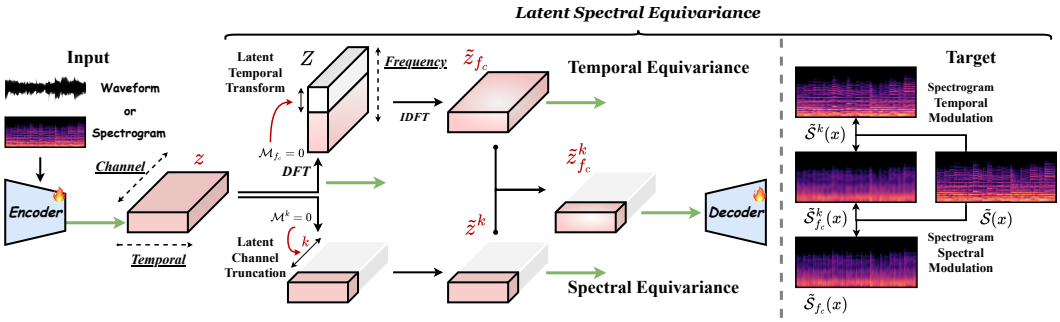

Figure 1: Overview of LATENTEQ, where temporal scale equivariance and spectral granularity equivariance pair coarsenings of the latent along its temporal and channel axes with matched coarsenings of the target spectrogram.

The structure of the latent space is a central design consideration in latent diffusion: the VAE determines not only what information is preserved, but also how that information is organized for the diffusion model to learn. Recent work increasingly emphasizes preserving the structure of the underlying signal in latent representations (Liu et al., 2025c; Lai et al., 2026; Ning et al., 2026). For audio, this calls for latents that explicitly reflect the temporal and spectral organization of acoustic signals. Existing approaches regularize audio latents through pretrained encoder features (Parker et al., 2026; Shutkin et al., 2026; MiniMax, 2026), temporal-spectrum or channel masking (Wang & Huang, 2026; Cheng et al., 2025b), and prior geometry (Liu et al., 2026), yet do not explicitly enforce a correspondence between latent transformations and audio acoustic transformations. Equivariance provides a direct mechanism for establishing this correspondence: by pairing operations in latent space with acoustic transformations of the audio signal, it can give the latent representation an explicit acoustic structure, facilitating diffusion modeling.

In this paper, we design Latent Spectral Equivariance (LATENTEQ), which to our knowledge is the first framework to introduce equivariance into the training of audio VAEs. Audio has long been analyzed along two axes, ranging over time from long-term dynamics to short transients such as onsets (Ding et al., 2017) and over frequency from the smooth envelope that characterizes timbre to fine harmonic detail (Serra & Smith, 1990; Chi et al., 2005). LATENTEQ therefore imposes equivariance along both axes by pairing each coarsening of the latent with a matched coarsening of the target spectrogram. Temporal Scale Equivariance (TSE) exploits the shared time axis of the latent and the target spectrogram, pairing low-pass filtering of both at the same cutoff frequency. This encourages slowly varying latent components to encode slowly varying acoustic structure. Since the latent has no frequency axis, Spectral Granularity Equivariance (SGE) instead pairs a prefix of its channels with matched smoothing of the spectrogram along frequency, so that the leading channels capture the spectral envelope and later ones progressively refine it. LATENTEQ requires no architectural change and no additional loss term, since fine-tuning a pretrained VAE suffices to impose both constraints jointly. Figure 1 summarizes the framework.

We instantiate LATENTEQ on the waveform and spectrogram VAEs of Stable Audio Open (Evans et al., 2025) and MMAudio (Cheng et al., 2025a), respectively. We evaluate the VAE on audio reconstruction and the DiT on both T2A and V2A generation tasks. In both systems, imposing equivariance on the VAE alone improves the generation quality of the DiT trained on its latents and also raises reconstruction fidelity. An analysis of the fine-tuned latents confirms that coarsening them along either axis yields a correspondingly coarsened sound, a property that the pretrained VAEs lack. We make the following contributions:

- We introduce equivariance into the training of audio VAEs for the first time, which characterizes the structure of audio across temporal scales and spectral granularities, gives latent operations an explicit acoustic meaning, and encourages the representation to follow these acoustic scales.
- We propose LATENTEQ, which consists of temporal scale equivariance and spectral granularity equivariance, coarsening the latent along the temporal and channel dimensions respectively and matching each with a corresponding smoothing of the target spectrogram.
- For both waveform- and spectrogram-based VAEs, we show that fine-tuning them with LATENTEQ improves both generation and reconstruction.

2 PRELIMINARIES

Audio Variational Autoencoders. We consider waveform and spectrogram VAEs that compress the input only along time and provide the latents on which diffusion models are trained. Given an audio input x, the encoder defines an approximate posterior $q_\phi(z \mid x)$ with prior $p(z) = \mathcal{N}(0, I)$, and we write $\mathcal{E}(x)$ for a sample from it. The input x has one of two forms:

$$x \in \begin{cases} \mathbb{R}^{C \times L}, & \text{for a waveform,} \\ \mathbb{R}^{B \times N}, & \text{for a spectrogram,} \end{cases} \tag{1}$$

where C and L are the numbers of audio channels and waveform samples, and B and N those of frequency bands and time frames, respectively. In both cases, the encoder produces a temporal latent sequence $z = \mathcal{E}(x) \in \mathbb{R}^{F \times d}$ with F frames and d channels, and the decoder maps it back to the form of x as $\hat{x} = \mathcal{D}(z)$. The latent frames lie on a uniform temporal grid over the same D-second duration as x, but the latent has no explicit frequency axis. Spectral content is instead represented across the d channels. We write $\mathcal{S}(x) \in \mathbb{R}^{B \times N}$ for the spectrogram on which reconstruction is measured. For a waveform VAE, $\mathcal{S}$ computes STFT magnitude spectrograms at several resolutions, each with its own B and N. For a spectrogram VAE, $\mathcal{S}$ is the identity on its log-mel input, and a separate vocoder converts $\hat{x}$ into a waveform. The VAE is trained with the following objective:

$$\mathcal{L}_{\mathrm{VAE}}(x) = \mathcal{L}_{\mathrm{rec}}(x, \hat{x}) + \lambda_{\mathrm{adv}} \mathcal{L}_{\mathrm{adv}}(x, \hat{x}) + \lambda_{\mathrm{KL}} \mathcal{L}_{\mathrm{KL}}(x). \tag{2}$$

The reconstruction term $\mathcal{L}_{\mathrm{rec}}$ compares $\mathcal{S}(\hat{x})$ with $\mathcal{S}(x)$. For a waveform VAE, it is a multi-resolution STFT loss on the magnitudes and their logarithms, and for a spectrogram VAE, it acts on $\hat{x}$ and x directly. The adversarial term $\mathcal{L}_{\mathrm{adv}}$ is induced by a discriminator that distinguishes the reconstruction $\hat{x}$ from the real input x. The term $\mathcal{L}_{\mathrm{KL}}(x)$ is the KL divergence between the approximate posterior and the prior. The coefficients λ_{adv} and λ_{KL} weight the corresponding terms.

Equivariance Regularization. Equivariance requires a mapping to commute with paired transformations of its input and output. For the VAE decoder, let T_z and T_x be paired latent and signal transformations sharing a composition law, so that equivariance reads as follows:

$$\mathcal{D}(T_z(z)) = T_x(\mathcal{D}(z)). \tag{3}$$

Group-equivariant networks enforce such relations through their architecture for transformations such as rotations and translations (Cohen & Welling, 2016). The formulation also accommodates noninvertible transformations such as coarsening (Skorokhodov et al., 2025) through semigroup actions (Worrall & Welling, 2019). Equivariance regularization instead encourages this relation through training, requiring latent operations to produce prescribed changes in the reconstructed signal. When reconstruction is accurate and T_x does not substantially amplify reconstruction errors, $T_x(x)$ provides a surrogate target for $T_x(\mathcal{D}(z))$. This motivates reconstructing the transformed signal from the transformed latent as implicit equivariance regularization (Kouzelis et al., 2025).

Discrete Fourier and Cosine Transforms. The discrete Fourier and cosine transforms (DFT and DCT) represent a real-valued sequence $u \in \mathbb{R}^J$ as coefficients over a family of oscillations, and their inverses (IDFT and IDCT) synthesize it from these coefficients. With the complex sinusoids $w_j[n] = e^{\mathrm{i}2\pi jn/J}$ and the orthonormal DCT-II basis $\{c_j\}_{j=0}^{J-1}$, the two pairs are defined as follows:

$$U[j] = u \cdot w_j, \quad u = \frac{1}{J} \sum_{j=0}^{J-1} U[j]\, w_j, \qquad V[j] = u \cdot c_j, \quad u = \sum_{j=0}^{J-1} V[j]\, c_j, \tag{4}$$

where $\cdot$ denotes the complex dot product, which conjugates its second argument. For real-valued u, only the first $\lfloor J/2 \rfloor + 1$ DFT coefficients are independent, and in both transforms the jth component oscillates faster as j grows, so lower-index coefficients describe slower variation. Zeroing the coefficients above a chosen index and inverting the transform therefore coarsens the sequence.

3 METHOD

3.1 ANALYSIS OF A PRETRAINED AUDIO VAE

We examine whether the latent of a pretrained audio VAE reflects the structure of audio across temporal scales and spectral granularities. Taking the pretrained waveform VAE of Stable Audio Open (SAO) (Evans et al., 2025) as an example, we apply temporal filtering and channel truncation to the latent and compare the decoded spectrograms with matched coarsened targets, testing whether these operations produce corresponding acoustic changes.

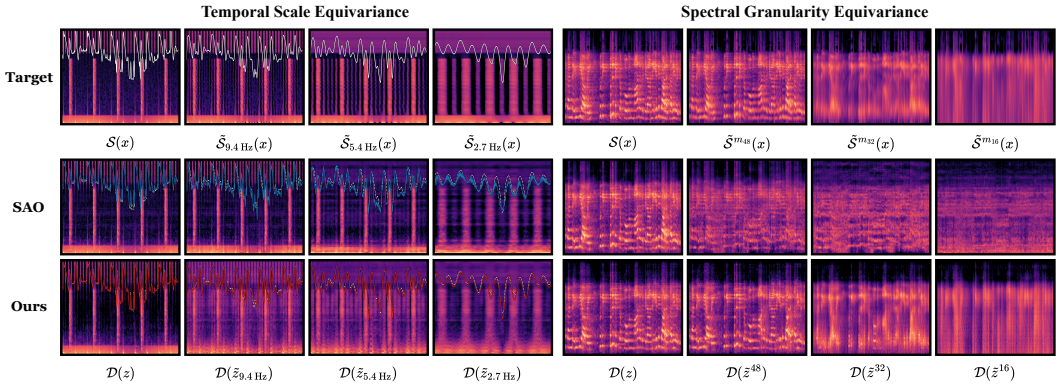

Figure 2: Qualitative comparison of the pretrained SAO VAE and its LATENTEQ counterpart under temporal low-pass filtering (left) and prefix-channel truncation (right), where the top row shows the original and matched coarsened spectrogram targets and the lower rows show the decoded outputs. The curves overlaid in the left panel trace the mean log-magnitude per frame.

Temporal Scale Inconsistency. The latent and the spectrogram span the same time interval, so their variation can be compared across temporal scales. Ideally, slowly varying latent components encode slowly varying acoustic structure. The desired correspondence follows the equivariance principle in Eq. 3, applied to $\mathcal{S} \circ \mathcal{D}$ with low-pass filters matched in modulation frequency as T_z and T_x. To test this relation, we low-pass filter the latent at progressively lower cutoffs, decode each result, and compare the resulting spectrogram with the target filtered at the same cutoff, as shown in Figure 2. As the cutoff decreases, the target loses its rapid variation and keeps only the slow contour, whereas the decoded spectrogram retains rapid fluctuations and deviates from the slow structure of the target. Low-pass filtering the pretrained latent therefore does not low-pass filter the sound.

Spectral Granularity Inconsistency. A spectrum can be smoothed from fine peaks and valleys to a coarse envelope, but the latent has no explicit frequency axis. We therefore consider an ordering in which nested channel prefixes (Rippel et al., 2014) preserve the spectral envelope and recover finer detail as they grow. The desired correspondence follows the equivariance principle in Eq. 3, applied to $\mathcal{S} \circ \mathcal{D}$ with channel truncation as T_z and spectral smoothing at matched granularity as T_x. To test this relation, we keep the first k channels, zero the rest, and compare the decoded spectrogram with the target smoothed to the matched granularity, as shown in Figure 2. As k decreases, the reconstruction becomes diffuse and noise-like without preserving the target envelope, indicating that the channel order does not reflect the intended spectral hierarchy. Nor does the training objective prescribe such an arrangement, since jointly permuting the encoder outputs and decoder inputs along the channel axis leaves Eq. 2 unchanged under an isotropic Gaussian prior (Rolinek et al., 2019).

Our Motivation. These observations motivate learning an explicit correspondence between latent coarsening and spectrogram smoothing. Because low-pass filtering the waveform changes acoustic bandwidth rather than temporal modulation, both coarsenings are defined on the spectrogram. The desired property is equivariance under coarsening, where coarsening the latent coarsens the sound and adding components adds detail. Imposing it as a regularization biases the latent toward transforming as the sound does, which is easier for the downstream diffusion model to learn.

3.2 PROBLEM FORMULATION

We fine-tune $\mathcal{E}$ and $\mathcal{D}$ so that the latent transforms as the sound does under coarsening along time and across channels, while preserving reconstruction from the full latent. The desired latent spectral equivariance applies Eq. 3 to $\mathcal{S} \circ \mathcal{D}$ and reads as follows:

$$\mathcal{S}(\mathcal{D}(T_z(z))) = T_x(\mathcal{S}(\mathcal{D}(z)))\,. \tag{5}$$

Here, T_z performs temporal filtering or channel truncation, and T_x applies the matched coarsening in the spectrogram domain. During training, we replace $T_x(\mathcal{S}(\mathcal{D}(z)))$ with the fixed surrogate $\tilde{\mathcal{S}}(x) = T_x(\mathcal{S}(x))$ and write $\tilde{z} = T_z(z)$, so that fine-tuning targets the following relation:

$$\mathcal{S}(\mathcal{D}(\tilde{z})) \approx \tilde{\mathcal{S}}(x). \tag{6}$$

We include the identity case to retain the original reconstruction task. Both latent transformations preserve the dimensions of z, so the same decoder processes full and coarsened latents alike.

3.3 TEMPORAL SCALE EQUIVARIANCE

Latent Temporal Transform. The latent $z \in \mathbb{R}^{F\times d}$ has F frames over the D-second clip, so its frame rate is F/D and its time axis is linearly aligned with that of the audio. A temporal scale in the latent can therefore be measured as a modulation frequency in Hz, with coarse scales corresponding to slow variation along the latent time axis. Applying the discrete Fourier transform (DFT) to each channel along the time axis gives the following representation:

$$Z = \mathrm{DFT}(z) \in \mathbb{C}^{K\times d}, \qquad K = \lfloor F/2 \rfloor + 1. \tag{7}$$

This representation expresses every channel as a sum of K sinusoids with linearly spaced modulation frequencies. The jth sinusoid oscillates j times over the clip and thus has modulation frequency j/D Hz, so the first sinusoid is constant and the highest one does not exceed half the frame rate of the latent. Given a cutoff f_c in this range, we define the coarsened latent using a mask $M_{f_c} \in \{0,1\}^K$ that is shared across channels and keeps only the sinusoids whose modulation frequency is at most f_c:

$$\tilde{z}_{f_c} = \mathrm{IDFT}\big(Z \odot M_{f_c}\big). \tag{8}$$

The result has F frames that carry the modulations of z up to f_c and none above it. The transform leaves the shape and the frame rate of the latent unchanged, so the same decoder applies without modification and $\mathcal{D}(\tilde{z}_{f_c})$ is still a full-length reconstruction of the same clip.

Spectrogram Temporal Modulation. At each STFT resolution, $\mathcal{S}(x) \in \mathbb{R}^{B\times N}$ has B frequency bands and N frames. The DFT of each band along time yields $K' = \lfloor N/2 \rfloor + 1$ sinusoids at the modulation frequencies j/D shared with the latent. The mask $M'_{f_c} \in \{0,1\}^{K'}$ retains the sinusoids with modulation frequency at most f_c, the same cutoff as M_{f_c}, and is shared across bands. Masking the DFT of every band at every resolution gives the following coarsened target:

$$\tilde{\mathcal{S}}_{f_c}(x) = \mathrm{IDFT}\big(\mathrm{DFT}(\mathcal{S}(x)) \odot M'_{f_c}\big). \tag{9}$$

The target $\tilde{\mathcal{S}}_{f_c}(x)$ keeps the modulations up to f_c at every resolution.

Cutoff Sampling. During training, f_c is drawn from a set of cutoffs spanning the modulation range of z and applied to both z and $\mathcal{S}(x)$, so that the latent learns to represent temporal variation from smooth global dynamics to sharp local transients. Meanwhile, we retain the unfiltered case to preserve full-bandwidth reconstruction.

3.4 SPECTRAL GRANULARITY EQUIVARIANCE

Latent Channel Truncation. The d channels of each latent frame jointly encode the spectral content of its segment, and the channel axis carries no intrinsic order. We therefore require the decoder to reconstruct from the first k channels alone, so that front channels capture coarse spectral structure and later channels add detail. Given $k \le d$, we define the truncated latent using a channel mask $M^k \in \{0,1\}^d$ that is shared across frames and keeps only the first k channels:

$$\tilde{z}^k = z \odot M^k. \tag{10}$$

The result has all F frames with the first k channels of z and nothing from the remaining channels. The truncation leaves the shape of the latent unchanged, so the same decoder applies without modification and $\mathcal{D}(\tilde{z}^k)$ is still a full-length reconstruction of the same clip.

Spectrogram Spectral Modulation. At each resolution, every frame of $\mathcal{S}(x)$ has B bands, and the DCT of its log-magnitude along frequency yields B cepstral coefficients, with the leading coefficients describing the broad spectral envelope and the later ones the fine spectral structure. The spectral granularity paired with k is set by the number of retained coefficients m_k, which increases monotonically with k. The mask $M'^k \in \{0,1\}^B$ retains the first m_k coefficients and zeroes the rest, and is shared across frames. Masking the DCT of every frame at every resolution and exponentiating back gives the following coarsened target:

$$\tilde{\mathcal{S}}^k(x) = \exp\Big(\mathrm{IDCT}\big(\mathrm{DCT}(\log \mathcal{S}(x)) \odot M'^k\big)\Big). \tag{11}$$

This is cepstral smoothing, so $\tilde{\mathcal{S}}^k(x)$ is the spectral envelope of $\mathcal{S}(x)$ at granularity m_k. For a spectrogram VAE, the input is already a log-mel spectrogram, so the logarithm and the exponential are omitted.

Pairing Schedule. During training, k is drawn from a set of channel counts spanning the width of z, each paired with a coefficient count m_k that grows geometrically with k, and the pair is applied to z and $\mathcal{S}(x)$, so that the latent learns to represent spectral structure from the broad envelope to fine harmonic detail. Meanwhile, we retain the untruncated case to preserve full-capacity reconstruction.

3.5 Overall LatentEQ Training

Implicit Regularization. We impose Eq. 6 implicitly by applying the transforms inside the reconstruction and adversarial losses of Eq. 2. Since f_c and k are sampled independently and each may take the identity setting, the transforms of Eqs. 8 to 11 give a training example one of four forms,

$$(\tilde{z}, \tilde{\mathcal{S}}(x)) = \begin{cases} \big(z, \mathcal{S}(x)\big) & \text{full reconstruction,} \\ \big(\tilde{z}_{f_c}, \tilde{\mathcal{S}}_{f_c}(x)\big) & \text{TSE only,} \\ \big(\tilde{z}^{k}, \tilde{\mathcal{S}}^{k}(x)\big) & \text{SGE only,} \\ \big(\tilde{z}^{k}_{f_c}, \tilde{\mathcal{S}}^{k}_{f_c}(x)\big) & \text{TSE and SGE,} \end{cases}$$

where the last form applies both transforms, first truncating channels and then low-passing along time. With $\hat{x} = \mathcal{D}(\tilde{z})$, we minimize Eq. 2, where every spectrogram of the real input that enters a loss term, $\mathcal{S}(x)$, is replaced by $\tilde{\mathcal{S}}(x)$, so that the first form recovers the original objective.

Algorithm 1 LatentEQ Training

Require: Pretrained $\mathcal{E}$, $\mathcal{D}$; spectrogram operator $\mathcal{S}$; cutoff set $\mathcal{F}$; channel set $\mathcal{K}$; each set contains the identity setting

for each training step **do**
  Sample a clip x and encode $z \leftarrow \mathcal{E}(x)$
  Sample $f_c \in \mathcal{F}$ and $k \in \mathcal{K}$
  $\tilde{z} \leftarrow z,\ \tilde{\mathcal{S}}(x) \leftarrow \mathcal{S}(x)$
  if k is not the identity setting **then**
    Truncate $\tilde{z}$ by Eq. 10, $\tilde{\mathcal{S}}(x)$ by Eq. 11
  end if
  if f_c is not the identity setting **then**
    Low-pass $\tilde{z}$ by Eq. 8, $\tilde{\mathcal{S}}(x)$ by Eq. 9
  end if
  $\hat{x} \leftarrow \mathcal{D}(\tilde{z})$
  Update $\mathcal{E}$ and $\mathcal{D}$ by Eq. 2
end for
return $\mathcal{E}$, $\mathcal{D}$

Training Procedure. Algorithm 1 summarizes the procedure. Starting from a pretrained VAE, each step samples f_c and k, coarsens the latent and the target accordingly, and updates the encoder and decoder with the resulting objective. Each example is decoded once, no loss term is added, and the architecture is unchanged. After fine-tuning, the VAE is frozen and the diffusion model is trained on its unmodified latents, with no transform applied during diffusion training or sampling.

4 Experiments

4.1 Experimental Setup

Datasets. For the waveform VAE, we fine-tune on Freesound (Font et al., 2013), FMA (Defferrard et al., 2016), and FSD50K (Fonseca et al., 2021). We convert the recordings to stereo at 44.1 kHz and remove silent segments. We train the downstream DiT for T2A generation on WavCaps (Mei et al., 2024) and AudioCaps (Kim et al., 2019). For the spectrogram VAE, we fine-tune on AudioSet (Gemmeke et al., 2017) and Freesound. We train the downstream DiT of this branch for VT2A generation on VGGSound (Chen et al., 2020), WavCaps, AudioCaps, and Clotho (Drossos et al., 2020), so that a single model serves both T2A and V2A. We evaluate the reconstruction of both VAEs on the AudioCaps test split and the Song Describer Dataset (Manco et al., 2023). For T2A evaluation, we use AudioCaps and VGGSound-Omni (Dai et al., 2026), whose clips carry caption-level annotations that serve as text prompts. V2A generation is evaluated on the VGGSound test split. Further details of the datasets and their preprocessing are provided in Appendix A.

Evaluation Metrics. For generation, we adopt metrics from AudioLDM (Liu et al., 2023), Stable Audio Open (Evans et al., 2025), and MMAudio (Cheng et al., 2025a). Fréchet distances are computed on OpenL3 (Cramer et al., 2019), VGGish (Hershey et al., 2017), and PANNs (Kong et al., 2020) embeddings and denoted $\mathrm{FD_O}$, $\mathrm{FD_V}$, and $\mathrm{FD_P}$, where $\mathrm{FD_V}$ is the FAD of Kilgour et al. (2018). The PANNs classifier additionally yields the Inception score and the KL divergence ($\mathrm{IS_P}$ and $\mathrm{KL_P}$), and PaSST (Koutini et al., 2021) gives a second estimate of the latter ($\mathrm{KL_S}$). Text-audio and video-audio alignment are measured as cosine similarities in the embedding spaces of CLAP (Wu et al., 2023) and ImageBind (Girdhar et al., 2023) (IB). For reconstruction, we follow SAO and report the STFT and mel-spectrogram distances (STFT-D and MEL-D) and SI-SDR (Le Roux et al., 2019), together with $\mathrm{FD_V}$. Details of all metrics are provided in Appendix B.

Table 1: Text-to-Audio generation quality on AudioCaps and VGGSound-Omni. Our entries keep the diffusion architecture of the system above them and replace only its VAE.

Method	Duration	AudioCaps							VGGSound-Omni						
		$FD_O\downarrow$	$FD_V\downarrow$	$FD_P\downarrow$	$KL_S\downarrow$	$KL_P\downarrow$	$IS_P\uparrow$	CLAP↑	$FD_O\downarrow$	$FD_V\downarrow$	$FD_P\downarrow$	$KL_S\downarrow$	$KL_P\downarrow$	$IS_P\uparrow$	CLAP↑
AudioGen	10 sec	189.15	1.88	17.92	1.42	1.54	9.33	0.458	217.30	2.37	13.91	1.97	2.10	11.00	0.287
MAGNET	10 sec	193.63	3.02	34.49	1.97	1.94	6.78	0.343	214.20	2.85	22.52	1.98	2.03	8.61	0.304
Spectrogram Latent Diffusion															
AudioLDM	10 sec	148.95	4.87	35.88	2.27	2.10	6.60	0.238	213.18	3.55	24.89	2.47	2.32	7.02	0.389
Tango	10 sec	125.31	4.00	29.75	1.61	1.61	5.08	0.330	200.00	3.71	21.36	1.88	1.96	6.61	0.395
AudioLDM 2	10 sec	107.85	3.66	35.62	1.75	1.56	7.40	0.410	198.31	2.78	21.83	1.84	1.90	11.37	0.295
MMAudio	8 sec	64.33	4.43	17.42	1.28	1.40	11.85	0.444	129.09	1.56	8.98	1.96	2.03	14.62	0.377
Ours	8 sec	62.71	3.01	14.23	1.19	1.30	12.35	0.458	92.65	1.48	8.45	1.87	1.89	13.93	0.391
Waveform Latent Diffusion															
EzAudio	10 sec	116.11	4.65	27.13	1.43	1.49	8.02	0.459	218.05	4.63	20.85	1.62	1.74	9.23	0.301
TangoFlux	30 sec	75.66	2.21	21.66	1.05	1.25	11.92	0.544	151.06	2.09	16.54	1.56	1.74	10.47	0.346
AudioX	10 sec	61.08	2.31	19.45	1.43	1.58	10.38	0.476	106.44	1.41	9.16	1.60	1.64	15.35	0.378
Stable Audio Open	47 sec	99.70	3.47	27.87	2.20	2.16	9.92	0.303	117.73	2.71	18.21	2.20	2.15	13.15	0.363
Ours	30 sec	60.26	1.99	17.05	1.33	1.35	11.73	0.495	94.73	1.62	17.56	1.53	1.60	12.93	0.374

Implementation Details. We instantiate LATENTEQ by fine-tuning the pretrained waveform and spectrogram VAEs of Stable Audio Open (Evans et al., 2025) and MMAudio (Cheng et al., 2025a), respectively, both operating at 44.1 kHz. The former encodes stereo audio into a 64-channel latent at 21.5 Hz, and the latter maps a 128-band mel spectrogram to 40 channels at 43.07 Hz. We fine-tune the two VAEs for 500k steps on 1.5-second and 8-second clips, respectively. At each step, one of the four training forms is drawn with equal probability, and the cutoff f_c and the channel count k follow fixed schedules. On each latent space, we train the diffusion model of the corresponding system from scratch with its original architecture, namely the T2A DiT of Stable Audio Open for 1M steps and the VT2A one of MMAudio for 300k. They generate 30-second and 8-second audio, respectively. As a controlled reference, we fine-tune each VAE under the same configuration without TSE and SGE and train an identical diffusion model on its latents. All models are trained on 8 NVIDIA A100 GPUs, and further details are provided in Appendix C.

4.2 MAIN RESULTS

Text-to-Audio Generation. Table 1 compares T2A generation on AudioCaps and VGGSound-Omni across three representation families. The discrete-token baselines are AudioGen (Kreuk et al., 2022) and MAGNET (Ziv et al., 2024). The spectrogram-latent baselines are AudioLDM, Tango (Ghosal et al., 2023), AudioLDM 2 (Liu et al., 2024a), and MMAudio. The waveform-latent baselines are EzAudio (Hai et al., 2024), AudioX (Tian et al., 2025), TangoFlux (Hung et al., 2024), and Stable Audio Open. Our two entries keep the diffusion architecture of MMAudio and Stable Audio Open and replace the VAE of each with its LATENTEQ counterpart. All baselines use their official checkpoints and default inference settings, and our entries follow those of the system they replace. Both entries improve nearly every metric over the systems they build on. Against systems that instead target the training objective or the architecture of the diffusion model, LATENTEQ reaches competitive quality by changing the VAE alone.

Table 2: V2A on the VGGSound test set (top) and VT2A on VGGSound-Omni (bottom).

Method	$FD_V\downarrow$	$KL_P\downarrow$	$IS_P\uparrow$	$FD_P\downarrow$	IB↑
V2A					
V-AURA	2.88	2.42	10.08	14.80	0.28
ReWaS	1.79	2.87	8.51	17.54	0.15
FRIEREN	1.34	2.73	12.25	11.45	0.23
FoleyCrafter	2.51	2.30	15.68	16.24	0.26
AudioX	1.55	3.03	14.26	14.98	0.25
MMAudio	0.85	2.09	14.19	6.56	0.29
Ours	0.75	2.00	11.44	5.68	0.30
VT2A					
ThinkSound(w/o. CoT)	1.10	1.60	11.73	7.41	0.26
HunyuanVideo-Foley	2.36	1.74	11.58	10.02	0.32
AudioX	1.18	1.59	15.08	8.30	0.26
MMAudio	0.81	1.63	13.16	5.82	0.30
Ours	0.73	1.60	13.95	5.89	0.31

Video-to-Audio Generation. Table 2 compares V2A generation on the VGGSound test split against V-AURA (Viertola et al., 2025), ReWaS (Jeong et al., 2025), FRIEREN (Wang et al., 2024), FoleyCrafter (Zhang et al., 2026), AudioX, and MMAudio, with the first four quoted from MMAudio and the remaining two evaluated by us. Since the spectrogram branch is trained on mixed audio-visual and audio-text data, it also accepts a video and a caption jointly, and the bottom of the table evaluates this VT2A setting on VGGSound-Omni against ThinkSound (Liu et al., 2025a) without chain-of-thought, HunyuanVideo-Foley (Shan et al., 2025), AudioX, and MMAudio, with the first two quoted from Omni2Sound. In both settings, LATENTEQ improves over MMAudio on most metrics, including video-audio alignment, so its gains extend to conditioning modalities beyond text.

Table 3: Reconstruction fidelity on AudioCaps (Sound) and the Song Describer Dataset (Music). Baselines use the official model weights, every entry is evaluated under the same protocol at 44.1 kHz, and † marks publicly reported results. SR denotes the sample rate.

Method	SR (kHz)	Latent Rate (Hz)	Latent Channels	AudioCaps (Sound)				Song Describer Dataset (Music)			
				STFT-D↓	MEL-D↓	SI-SDR↑	FD_V↓	STFT-D↓	MEL-D↓	SI-SDR↑	FD_V↓
DAC	44.1	86	discrete	1.10	0.65	5.67	1.37	1.31	0.96	10.86	0.48
AGC-Codec†	48	100	discrete	2.18	1.41	-1.25	2.36	2.62	1.50	5.55	1.16
Encodec	32	50	discrete	1.69	1.28	3.14	11.58	1.56	1.01	5.42	0.76
Spectrogram VAE											
AudioLDM	16	25	8	2.48	1.81	-35.83	1.25	3.31	2.34	-45.18	0.96
Make-An-Audio 2	16	31.25	20	2.43	1.66	-34.18	1.39	3.04	1.87	-40.25	0.71
AudioLDM 2	16	25	8	2.10	1.31	-35.02	1.25	2.63	1.45	-40.89	0.92
MMAudio	44.1	43.07	40	1.25	0.87	-33.88	0.97	1.12	0.74	-39.55	0.57
Ours	44.1	43.07	40	1.12	0.65	-34.04	0.76	1.01	0.68	-35.72	0.42
Waveform VAE											
HunyuanVideo Foley	48	50	128	0.99	0.36	11.91	0.19	0.92	0.30	16.96	0.33
ϵar-VAE	44.1	43.07	64	1.06	0.68	6.98	4.40	0.92	0.56	11.58	0.28
STAR-VAE†	44.1	21.5	64	1.17	0.75	-0.03	2.31	1.32	0.80	6.40	0.25
Stable Audio Open	44.1	21.5	64	1.23	0.85	0.08	3.60	1.32	0.79	5.82	0.48
Ours	44.1	21.5	64	1.01	0.73	0.94	2.42	0.94	0.72	6.80	0.31

Audio Reconstruction. Table 3 compares reconstruction fidelity across three families of audio autoencoders. We compare against the discrete neural codecs DAC (Kumar et al., 2023), Encodec (Défossez et al., 2022), and AGC-Codec (Audiogen AI, 2024), which use discrete bottlenecks and operate at higher latent rates. The spectrogram VAE baselines are AudioLDM, Make-An-Audio 2 (Huang et al., 2023a), AudioLDM 2, and MMAudio. The waveform VAE baselines are HunyuanVideo Foley (Shan et al., 2025), ϵar-VAE (Wang et al., 2025), STAR-VAE (Liu et al., 2026), and Stable Audio Open. Under its equivariance constraints, LATENTEQ improves downstream generation while also raising the reconstruction fidelity of both pretrained VAEs.

4.3 Ablation Studies and Analysis

Method	Reconstruction				Generation				
	S-D↓	M-D↓	S-S↑	FD_V↓	FD_V↓	KL_P↓	IS_P↑	FD_P↓	CLAP↑
MMAudio	1.25	0.87	-33.88	0.97	4.43	1.40	11.85	17.42	0.444
FT VAE	1.16	0.82	-34.18	0.95	4.04	1.46	11.75	15.17	0.437
w/ TSE	1.14	0.77	-33.76	0.96	3.82	1.27	11.65	15.61	0.440
w/ SGE	1.10	0.76	-34.00	0.85	3.19	1.30	12.22	14.22	0.451
Ours	1.12	0.65	-34.04	0.76	3.01	1.30	12.35	14.23	0.458
SAO	1.23	0.85	0.08	3.60	3.47	2.16	9.92	27.87	0.303
FT VAE	1.22	0.87	0.12	3.48	3.01	1.85	9.96	22.17	0.401
w/ TSE	1.14	0.78	0.80	3.29	2.43	1.44	10.65	18.76	0.434
w/ SGE	1.22	0.76	0.74	3.21	2.35	1.37	10.27	19.88	0.473
Ours	1.01	0.73	0.94	2.42	1.99	1.35	11.73	17.05	0.495

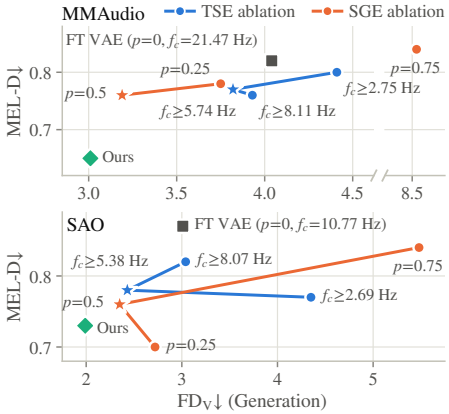

Figure 3: (Left) Ablation of TSE and SGE on reconstruction fidelity and downstream generation quality. (Right) Coarsening schedules on MMAudio (top) and SAO (bottom).

Temporal and Spectral Equivariance Ablation. Figure 3 (left) compares the official systems with four variants sharing our training configuration, using neither constraint, TSE alone, SGE alone, or both. All variants fine-tune the pretrained VAE and train the diffusion model from scratch, and each active constraint is applied independently with probability 0.5. Reconstruction and generation are evaluated on the AudioCaps test split. Fine-tuning alone already lowers generation FD_V relative to the official systems, each constraint lowers it further, their combination is lowest, and LatentEQ improves every reported generation and reconstruction metric over the unconstrained controls.

Coarsening Schedule Ablation. For variants using TSE or SGE alone, Figure 3 (right) varies the minimum cutoff f_c or the truncation probability p, respectively, with settings detailed in Appendix C. Our results show that the defaults minimize generation FD_V on both VAEs, that weaker constraints reduce or eliminate the gains, and that stronger ones fall below the unconstrained controls, while reconstruction MEL-D can favor other settings. We attribute the TSE trend to the envelope modulation rates of natural sounds, which peak near 5 Hz for speech and 2 Hz for music (Ding et al., 2017) and remain within every target under the default minima. Minima near 2.7 Hz instead remove

modulations that carry the timing of acoustic events, as the loss of intelligibility under envelope smearing shows (Drullman et al., 1994), so the coarsened targets no longer preserve the temporal structure that the latent should encode.

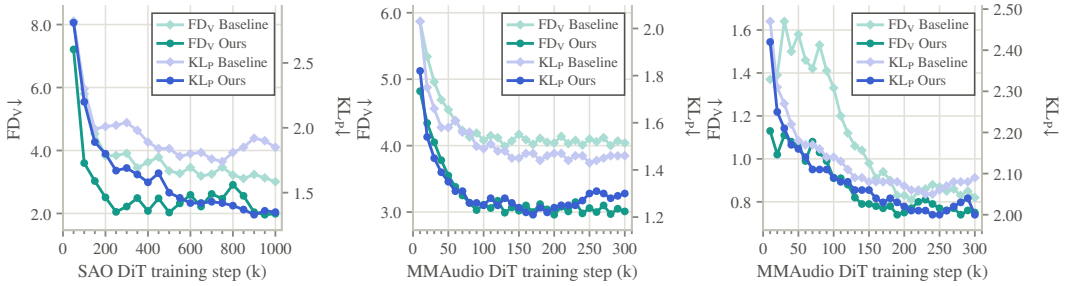

Figure 4: FD$_V$ and KL$_P$ over DiT training for the SAO DiT on AudioCaps (left), the MMAudio DiT on AudioCaps (middle), and the MMAudio DiT on VGGSound for V2A (right), comparing the latents of the VAE fine-tuned without TSE and SGE (Baseline) with those of LatentEQ (Ours).

Training Dynamics. Figure 4 tracks FD$_V$ and KL$_P$ on the test splits over DiT training, comparing the latents of VAEs fine-tuned without TSE and SGE with those of LatentEQ. Across the three settings, the LatentEQ latents give lower FD$_V$ and KL$_P$ throughout training and at the final checkpoint, and the SAO DiT also converges markedly faster on them.

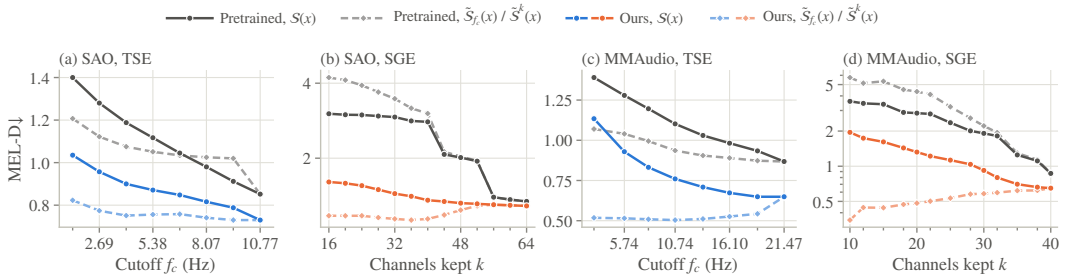

Figure 5: Equivariance of the pretrained and LatentEQ VAEs under temporal low-pass filtering at cutoff f_c (TSE) and truncation to the first k channels (SGE), measured by the MEL-D between the decoded coarsened latent and either $\mathcal{S}(x)$ or the matched target.

Equivariance Analysis. Figure 5 tests the intended equivariance on the AudioCaps test split by low-passing the latent at a cutoff f_c or truncating it to its first k channels, decoding the result, and computing MEL-D against both $\mathcal{S}(x)$ and the matched target. For the pretrained VAEs, the deviation from the matched target grows with the strength of coarsening, and channel truncation drives the output far from both references, which quantifies the two inconsistencies of Section 3.1. With LatentEQ, this deviation stays close to or below the full-capacity reconstruction error at every cutoff and truncation depth, with only a mild increase below the training range, while the deviation from $\mathcal{S}(x)$ grows smoothly with coarsening. The latent trained with TSE and SGE therefore transforms as the sound does along time and across channels.

5 Conclusion

We introduced LatentEQ, an equivariance regularization framework under which audio VAE latents transform as the signal does across temporal scales and spectral granularities. It pairs temporal low-pass filtering and prefix-channel truncation of the latent with matched coarsenings of the target spectrogram, and it is imposed by fine-tuning a pretrained VAE under its existing objective without architectural changes or additional loss terms. On the waveform VAE of Stable Audio Open and the spectrogram VAE of MMAudio, the diffusion model trained on LatentEQ latents improves text-to-audio and video-to-audio generation and converges faster, and the VAE itself reconstructs more faithfully. Ablations attribute the gains to both constraints, and the fine-tuned latents indeed transform as the signal does under coarsening, whereas the pretrained latents do not. Latents that transform as the signal does are thus easier for diffusion models to learn, and equivariance regularization offers a simple way to obtain them from a pretrained VAE.

AI USE STATEMENT

We used generative AI tools to draft and edit parts of the text for readability and to check the manuscript for internal consistency. All AI-assisted text was reviewed and revised by the authors, who take full responsibility for the content of the paper. Generative AI tools were not used to generate data, to develop the method, to design or run the experiments, or to analyze the results.

REFERENCES

Nasir Ahmed, T. Raj Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Trans. Computers*, 23(1):90–93, 1974.

Audiogen AI. AGC: Audiogen codec. GitHub repository, 2024. URL https://github.com/AudiogenAI/agc.

Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. In *Forty-second International Conference on Machine Learning*, 2025a.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 721–725. IEEE, 2020.

Junyu Chen, Dongyun Zou, Wenkun He, Junsong Chen, Enze Xie, Song Han, and Han Cai. Dc-ae 1.5: Accelerating diffusion model convergence with structured latent space. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025b.

Ke Chen, Yusong Wu, Haohe Liu, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies. In *ICASSP 2024-2024 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 1206–1210. IEEE, 2024.

Ho Kei Cheng, Masato Ishii, Akio Hayakawa, Takashi Shibuya, Alexander Schwing, and Yuki Mitsufuji. Mmaudio: Taming multimodal joint training for high-quality video-to-audio synthesis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28901–28911. IEEE, 2025a.

Yuqing Cheng, Xingyu Ma, Jiahe Lei, Guochen Yu, Andong Li, Naihan Li, Xiaotao Gu, Jie Tang, Xiaobing Li, Feng Yu, et al. Listens like Mel: Boosting latent audio diffusion with channel locality. https://openreview.net/forum?id=5PBKxl7o49, 2025b. Submitted to the International Conference on Learning Representations (ICLR) 2026.

Taishih Chi, Powen Ru, and Shihab A Shamma. Multiresolution spectrotemporal analysis of complex sounds. *The Journal of the Acoustical Society of America*, 118(2):887–906, 2005.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36:47704–47720, 2023.

Aurora Linh Cramer, Ho-Hsiang Wu, Justin Salamon, and Juan Pablo Bello. Look, listen, and learn more: Design choices for deep audio embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3852–3856. IEEE, 2019.

Yusheng Dai, Zehua Chen, Yuxuan Jiang, Baolong Gao, Qiuhong Ke, Jun Zhu, and Jianfei Cai. Omni2sound: Towards unified video-text-to-audio generation. *arXiv preprint arXiv:2601.02731*, 2026.

Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*, 2016.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Nai Ding, Aniruddh D Patel, Lin Chen, Henry Butler, Cheng Luo, and David Poeppel. Temporal modulations in speech and music. *Neuroscience & Biobehavioral Reviews*, 81:181–187, 2017.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 736–740. IEEE, 2020.

Rob Drullman, Joost M Festen, and Reinier Plomp. Effect of temporal envelope smearing on speech reception. *The Journal of the Acoustical Society of America*, 95(2):1053–1064, 1994.

Zach Evans, CJ Carr, Josiah Taylor, Scott H Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion. In *Forty-first International Conference on Machine Learning*, 2024a.

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Long-form music generation with latent diffusion. *arXiv preprint arXiv:2404.10301*, 2024b.

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. Fsd50k: an open dataset of human-labeled sound events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:829–852, 2021.

Frederic Font, Gerard Roma, and Xavier Serra. Freesound technical demo. In *Proceedings of the 21st ACM international conference on Multimedia*, pp. 411–412, 2013.

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria. Text-to-audio generation using instruction guided latent diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3590–3598, 2023.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind one embedding space to bind them all. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15180–15190. IEEE, 2023.

Jiarui Hai, Yong Xu, Hao Zhang, Chenxing Li, Helin Wang, Mounya Elhilali, and Dong Yu. Ezaudio: Enhancing text-to-audio generation with efficient diffusion transformer. *arXiv preprint arXiv:2409.10819*, 2024.

Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pp. 131–135. IEEE, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jiawei Huang, Yi Ren, Rongjie Huang, Dongchao Yang, Zhenhui Ye, Chen Zhang, Jinglin Liu, Xiang Yin, Zejun Ma, and Zhou Zhao. Make-an-audio 2: Temporal-enhanced text-to-audio generation. *arXiv preprint arXiv:2305.18474*, 2023a.

Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *International Conference on Machine Learning*, pp. 13916–13932. PMLR, 2023b.

Chia-Yu Hung, Navonil Majumder, Zhifeng Kong, Ambuj Mehrish, Amir Ali Bagherzadeh, Chuan Li, Rafael Valle, Bryan Catanzaro, and Soujanya Poria. Tangoflux: Super fast and faithful text to audio generation with flow matching and clap-ranked preference optimization. *arXiv preprint arXiv:2412.21037*, 2024.

Yujin Jeong, Yunji Kim, Sanghyuk Chun, and Jiyoung Lee. Read, watch and scream! sound generation from text and video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17590–17598, 2025.

Yuxuan Jiang, Zehua Chen, Zeqian Ju, Yusheng Dai, Weibei Dou, and Jun Zhu. Controlaudio: Tackling text-guided, timing-indicated and intelligible audio generation via progressive diffusion modeling. *arXiv preprint arXiv:2510.08878*, 2025.

Jaemin Jung, Junseok Ahn, Chaeyoung Jung, Tan Dat Nguyen, Youngjoon Jang, and Joon Son Chung. Voicedit: Dual-condition diffusion transformer for environment-aware speech synthesis. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fr\'echet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 119–132, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.

Khaled Koutini, Jan Schlüter, Hamid Eghbal-Zadeh, and Gerhard Widmer. Efficient training of audio transformers with patchout. *arXiv preprint arXiv:2110.05069*, 2021.

Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Eq-vae: Equivariance regularized latent space for improved generative image modeling. *arXiv preprint arXiv:2502.09509*, 2025.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*, 2022.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in neural information processing systems*, 36:27980–27993, 2023.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249, 2022.

Bolin Lai, Xudong Wang, Saketh Rambhatla, James M Rehg, Zsolt Kira, Rohit Girdhar, and Ishan Misra. Toward diffusible high-dimensional latent spaces: A frequency perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43450–43460, 2026.

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. Sdr–half-baked or well done? In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 626–630. IEEE, 2019.

Sang-gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. Bigvgan: A universal neural vocoder with large-scale training. *arXiv preprint arXiv:2206.04658*, 2022.

Sang-gil Lee, Zhifeng Kong, Arushi Goel, Sungwon Kim, Rafael Valle, and Bryan Catanzaro. Etta: Elucidating the design space of text-to-audio models. *arXiv preprint arXiv:2412.19351*, 2024.

Xiquan Li, Junxi Liu, Yuzhe Liang, Zhikang Niu, Wenxi Chen, and Xie Chen. Meanaudio: Fast and faithful text-to-audio generation with mean flows. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14378–14393, 2026.

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. *arXiv preprint arXiv:2301.12503*, 2023.

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

Huadai Liu, Rongjie Huang, Yang Liu, Hengyuan Cao, Jialei Wang, Xize Cheng, Siqi Zheng, and Zhou Zhao. Audiolcm: Text-to-audio generation with latent consistency models. *arXiv preprint arXiv:2406.00356*, 2024b.

Huadai Liu, Kaicheng Luo, Jialei Wang, Wen Wang, Qian Chen, Zhou Zhao, and Wei Xue. Thinksound: Chain-of-thought reasoning in multimodal large language models for audio generation and editing. *arXiv preprint arXiv:2506.21448*, 2025a.

Huadai Liu, Jialei Wang, Rongjie Huang, Yang Liu, Heng Lu, Zhou Zhao, and Wei Xue. Flashaudio: Rectified flow for fast and high-fidelity text-to-audio generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13694–13710, 2025b.

Huadai Liu, Wen Wang, Kaicheng Luo, Qian Chen, Xiangang Li, and Wei Xue. Star-vae: Structured topology-aware regularization for audio reconstruction and generation. *arXiv preprint arXiv:2606.23064*, 2026.

Shizhan Liu, Xinran Deng, Zhuoyi Yang, Jiayan Teng, Xiaotao Gu, and Jie Tang. Delving into latent spectral biasing of video vaes for superior diffusability. *arXiv preprint arXiv:2512.05394*, 2025c.

Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. *Advances in Neural Information Processing Systems*, 36: 48855–48876, 2023.

Navonil Majumder, Chia-Yu Hung, Deepanway Ghosal, Wei-Ning Hsu, Rada Mihalcea, and Soujanya Poria. Tango 2: Aligning diffusion-based text-to-audio generations through direct preference optimization. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 564–572, 2024.

Ilaria Manco, Benno Weck, Seungheon Doh, Minz Won, Yixiao Zhang, Dmitry Bogdanov, Yusong Wu, Ke Chen, Philip Tovstogan, Emmanouil Benetos, et al. The song describer dataset: a corpus of audio captions for music-and-language evaluation. *arXiv preprint arXiv:2311.10057*, 2023.

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

MiniMax. MiniMax H3: An open model breaking the boundaries between tasks and modalities. `https://www.minimax.io/blog/minimax-h3`, 2026.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PmLR, 2021.

Mang Ning, Mingxiao Li, Le Zhang, Lanmiao Liu, Matthew B Blaschko, Albert Ali Salah, and Itir Onal Ertugrul. Spectrum matching: a unified perspective for superior diffusability in latent diffusion. *arXiv preprint arXiv:2603.14645*, 2026.

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.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Oren Rippel, Michael Gelbart, and Ryan Adams. Learning ordered representations with nested dropout. In *International Conference on Machine Learning*, pp. 1746–1754. PMLR, 2014.

Michal Rolinek, Dominik Zietlow, and Georg Martius. Variational autoencoders pursue pca directions (by accident). In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12398–12407. IEEE, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Flavio Schneider, Ojasv Kamal, Zhijing Jin, and Bernhard Schölkopf. Moûsai: Efficient text-to-music diffusion models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8050–8068, 2024.

Prem Seetharaman and Rithesh Kumar. Taming audio vaes via target-kl regularization. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 15487–15491. IEEE, 2026.

Xavier Serra and Julius Smith. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24, 1990.

Sizhe Shan, Qiulin Li, Yutao Cui, Miles Yang, Yuehai Wang, Qun Yang, Jin Zhou, and Zhao Zhong. Hunyuanvideo-foley: Multimodal diffusion with representation alignment for high-fidelity foley audio generation. *arXiv preprint arXiv:2508.16930*, 2025.

Andrey Shutkin, Denis Parkhomenko, Ivan Kirillov, Kirill Chernyshev, Kirill Malakhov, Ilia Vasiliev, Ilia Trushkin, Valeriya Kobenko, David Chikovani, Alexander Ivanov, et al. Kvae: Family of tokenizers for multimodal generative models. *arXiv preprint arXiv:2608.05798*, 2026.

Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. *arXiv preprint arXiv:2502.14831*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Christian J Steinmetz and Joshua D Reiss. auraloss: Audio focused loss functions in pytorch. In *Digital music research network one-day workshop (DMRN+ 15)*, volume 2, 2020.

Zeyue Tian, Yizhu Jin, Zhaoyang Liu, Ruibin Yuan, Xu Tan, Qifeng Chen, Wei Xue, and Yike Guo. Audiox: Diffusion transformer for anything-to-audio generation. *arXiv e-prints*, pp. arXiv–2503, 2025.

Bernardo Torres, Manuel Moussallam, and Gabriel Meseguer-Brocal. Learning linearity in audio consistency autoencoders via implicit regularization. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 15927–15931. IEEE, 2026.

Ilpo Viertola, Vladimir Iashin, and Esa Rahtu. Temporally aligned audio for video with autoregression. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

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.

Mason Wang and Anna Huang. Latent fourier transform. In *International Conference on Learning Representations*, volume 2026, pp. 34791–34822, 2026.

Yongqi Wang, Wenxiang Guo, Rongjie Huang, Jiawei Huang, Zehan Wang, Fuming You, Ruiqi Li, and Zhou Zhao. Frieren: Efficient video-to-audio generation network with rectified flow matching. *Advances in neural information processing systems*, 37:128118–128138, 2024.

Daniel Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. *Advances in Neural Information Processing Systems*, 32, 2019.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Jinlong Xue, Yayue Deng, Yingming Gao, and Ya Li. Auffusion: Leveraging the power of diffusion and large language models for text-to-audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:4700–4712, 2024.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15703–15712. IEEE, 2025.

Yiming Zhang, Yicheng Gu, Yanhong Zeng, Zhening Xing, Yuancheng Wang, Zhizheng Wu, Bin Liu, and Kai Chen. Foleycrafter: Bring silent videos to life with lifelike and synchronized sounds. *International Journal of Computer Vision*, 134(1):46, 2026.

Yifan Zhou, Zeqi Xiao, Shuai Yang, and Xingang Pan. Alias-free latent diffusion models: Improving fractional shift equivariance of diffusion latent space. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 34–44. IEEE, 2025.

Alon Ziv, Itai Gat, Gael Le Lan, Tal Remez, Felix Kreuk, Jade Copet, Alexandre Défossez, Gabriel Synnaeve, and Yossi Adi. Masked audio generation using a single non-autoregressive transformer. In *International Conference on Learning Representations*, volume 2024, pp. 43249–43268, 2024.

A DATASETS

A.1 TRAINING DATASETS

The data are organized by stage, since each VAE is fine-tuned and its diffusion model is then re-trained from scratch. The paragraphs below describe its datasets, preprocessing, and sampling in turn. Table 4 lists the duration of each dataset as used in this work.

Waveform VAE Fine-Tuning. We adopt the training data setup of Stable Audio Open (Evans et al., 2025) and STAR-VAE (Liu et al., 2026) and fine-tune the waveform VAE on Freesound (Font et al., 2013), FMA (Defferrard et al., 2016), and FSD50K (Fonseca et al., 2021). Freesound is a collaborative repository of user-uploaded audio, ranging from field recordings and sound effects to instrument samples. FMA is a collection of Creative Commons music across many genres, and its large subset (FMA-large) provides a 30-second clip of every track. FSD50K is a dataset of Freesound clips with human-verified labels of everyday sound events. Our preprocessing follows that of Stable Audio Open. Each recording is divided into 5-second chunks, and those whose peak level falls below -60 dBFS are discarded as silent. At loading time, each chunk is resampled to 44.1 kHz, converted to stereo, and randomly cropped or zero-padded to 65,536 samples, about 1.49 seconds. The polarity of each chunk is inverted with probability 0.5. Each chunk of FSD50K is drawn three times as often as the others.

Spectrogram VAE Fine-Tuning. We adopt the training data setup of MMAudio (Cheng et al., 2025a) and fine-tune the spectrogram VAE on AudioSet (Gemmeke et al., 2017) and Freesound. AudioSet is a large collection of YouTube clips annotated with a hierarchical ontology of sound events. MMAudio trains its 44.1 kHz VAE on these two datasets because reconstruction is harder at a higher sample rate, and this model is our starting point. Our preprocessing follows that of MMAudio and Make-An-Audio 2 (Huang et al., 2023a). Each recording is downmixed to mono by averaging its channels, peak-normalized to 0.95, and resampled to 44.1 kHz. We then compute a natural-log mel spectrogram over the full recording from a 2048-point STFT with a hop size of 512 samples, using 128 bands that span 0 to 22.05 kHz. From AudioSet, we remove the recordings that overlap with the test splits of VGGSound (Chen et al., 2020) and AudioCaps (Kim et al., 2019). During training, we crop 690 mel frames, about 8 seconds, at a random position, and reject audio that is shorter than this length or silent instead of padding it.

Waveform Latent Diffusion Training. We adopt the training data setup of STAR-VAE and TangoFlux (Hung et al., 2024) and train the waveform latent diffusion model on WavCaps (Mei et al., 2024) and the training split of AudioCaps. WavCaps gathers audio from several online sources and pairs each clip with a caption generated by a large language model from its raw description. AudioCaps is a subset of AudioSet with human-written captions. The model is trained for T2A generation, with the captions of each dataset serving as text prompts. Since the public release of WavCaps is downsampled to 32 kHz, we follow MMAudio and retrieve the original recordings wherever we can locate them, which avoids artifacts when training at 44.1 kHz. Clips that overlap with the test splits of AudioCaps and VGGSound are removed from the training data. All clips are resampled to 44.1 kHz, and mono audio is duplicated across two channels to match the stereo input of the VAE. Each clip is fixed to 30 seconds by padding shorter ones with silence and randomly cropping longer ones.

Spectrogram Latent Diffusion Training. We adopt the training data setup of MMAudio and train the spectrogram latent diffusion model on VGGSound, WavCaps, AudioCaps, and Clotho (Drossos et al., 2020). As in MMAudio, training mixes audio-visual and audio-text data, so that a single model handles VT2A, V2A, and T2A generation. VGGSound is a collection of YouTube videos labeled with sound classes, whose names serve as text prompts, whereas the other three datasets contain captioned audio. Clotho pairs Freesound clips with human-written captions. For captioned audio without video, the visual and synchronization features are replaced with learnable empty tokens. WavCaps is used with its original high-sample-rate recordings as in the waveform branch, and its SoundBible subset is excluded. Each training sample is an 8-second audio crop. Recordings shorter than 16 seconds contribute a single crop from their start, whereas longer ones yield up to five without overlap. Clips that overlap with the test splits of VGGSound, AudioCaps, and Clotho are removed from the training data. Each VGGSound crop is drawn three times as often as the others.

Table 4: Durations of the training datasets in hours.

Dataset	Duration (h)
Freesound	5,771.7
FMA	877.5
FSD50K	76.7
AudioSet	5,305.8
WavCaps	7,494.5
AudioCaps	135.4
VGGSound	472.9
Clotho	30.5

A.2 Evaluation Datasets

We evaluate T2A and V2A generation, and reconstruction on sound and music. The test clips of AudioCaps and VGGSound are excluded from all training data.

Text-to-Audio Generation. The test split of AudioCaps contains 975 clips, each with five human-written captions. Following Stable Audio Open, we generate one sample per caption. VGGSound-Omni, introduced by Omni2Sound (Dai et al., 2026), builds on the test split of VGGSound and provides captions for over 14K clips, which serve as text prompts. Its captions are generated automatically, audited by a language model, and corrected by human annotators where flagged. On both benchmarks, the diffusion models are conditioned on the captions alone, and the videos of VGGSound-Omni are not used.

Video-to-Audio Generation. Following MMAudio, we evaluate V2A generation on the VGGSound test split, which contains about 15K videos. Our model and MMAudio generate 8 seconds of audio, and the longer outputs of the other baselines are truncated to this length. Each test video is 10 seconds long. For VT2A generation, the captions of VGGSound-Omni serve as the text condition alongside the videos.

Audio Reconstruction. Following Stable Audio Open, we evaluate reconstruction on the test split of AudioCaps and on the no-singing subset of the Song Describer Dataset (Manco et al., 2023). The Song Describer Dataset pairs Creative Commons music recordings with human-written captions, and the no-singing subset defined by Stable Audio Open keeps the instrumental tracks.

B Evaluation Metrics

Text-to-Audio Generation. Following AudioLDM (Liu et al., 2023), we compute $\mathrm{FD_P}$, $\mathrm{KL_P}$, $\mathrm{IS_P}$, and $\mathrm{FD_V}$ with its `audioldm_eval` toolkit. $\mathrm{FD_P}$, $\mathrm{KL_P}$, and $\mathrm{IS_P}$ are built on the CNN14 audio classifier of PANNs (Kong et al., 2020). $\mathrm{FD_P}$ compares the embedding distributions of the generated and reference sets, $\mathrm{KL_P}$ averages the divergence between the class probabilities of each generated sample and its paired reference, and $\mathrm{IS_P}$ reflects both the quality and the diversity of the outputs. $\mathrm{FD_V}$ is the Fréchet audio distance of Kilgour et al. (2018), computed on the embeddings of VGGish (Hershey et al., 2017). Following Stable Audio Open, we further compute $\mathrm{FD_O}$, $\mathrm{KL_S}$, and CLAP with its `stable-audio-metrics` toolkit. $\mathrm{FD_O}$ is the Fréchet distance on OpenL3 (Cramer et al., 2019) embeddings, which are computed at 44.1 kHz and concatenated across the left and right channels. $\mathrm{KL_S}$ uses PaSST (Koutini et al., 2021), an audio tagger trained on AudioSet that operates at 32 kHz, and averages its logits over overlapping windows of long inputs before the softmax. CLAP is the cosine similarity between the embeddings of each prompt and its generated sample under the `630k-audioset-fusion-best` checkpoint of LAION-CLAP (Wu et al., 2023).

Video-to-Audio Generation. Following MMAudio, we compute $\mathrm{FD_P}$, $\mathrm{FD_V}$, $\mathrm{KL_P}$, and $\mathrm{IS_P}$ together with IB using its `av-benchmark` toolkit. IB measures audio-visual alignment by the cosine similarity between the ImageBind (Girdhar et al., 2023) embeddings of the input video and the generated audio.

Audio Reconstruction. Following Stable Audio Open, we compute STFT-D, MEL-D, and SI-SDR (Le Roux et al., 2019) with the auraloss library (Steinmetz & Reiss, 2020) using its default parameters. STFT-D and MEL-D are distances between the magnitude spectrograms of the reconstruction and the reference on linear and mel frequency scales, SI-SDR is their scale-invariant signal-to-distortion ratio, and $\mathrm{FD_V}$ is computed between the reconstructed and reference sets. SI-SDR is computed on the waveform, so the values of the spectrogram VAEs in Table 3 mainly reflect that their vocoder does not reproduce the phase of the input. All metrics are computed at 44.1 kHz, and for each baseline the input is resampled to its native rate and the output back again. Mono models reconstruct the left and right channels independently, so that every reconstruction is stereo.

C Implementation Details

VAE Fine-Tuning. Both VAEs are initialized from their pretrained weights and fine-tuned with their original losses and discriminators. For the waveform VAE, the objective of Stable Audio Open combines a multi-resolution STFT loss over seven FFT sizes from 2048 down to 32 with adversarial, feature-matching, and KL terms, where the adversarial term uses the multi-scale STFT discriminator of EnCodec over five FFT sizes from 2048 to 128 and the feature-matching weight is 5.0. For the spectrogram VAE, we keep the architecture of MMAudio, which adopts the 1D convolutional design of Make-An-Audio 2 (Huang et al., 2023a) with a temporal downsampling factor of 2, and fine-tune it with the training recipe of Make-An-Audio 2, whose objective combines reconstruction, adversarial, and KL terms and whose adversarial term uses a PatchGAN discriminator with an R_1 penalty and is introduced after 40k steps. At every step, one of the four training forms in Section 3.5 is drawn with equal probability. The controlled reference of the main experiments shares this configuration without TSE and SGE.

Coarsening Schedules. For TSE, the cutoff of a latent that retains its first n Fourier components is $f_c = (n-1)\,\Delta f$, where Δf is their spacing. The waveform latent has 17 such components with $\Delta f = 0.6729$ Hz, and the spectrogram latent has 173 with $\Delta f = 0.1248$ Hz. In the waveform branch, the top r components are removed with r drawn uniformly from $\{1, \ldots, 8\}$, which retains 9 to 16 of them and places f_c in the range 5.38–10.09 Hz. In the spectrogram branch, n is drawn uniformly from $\{47, \ldots, 95\}$, which places f_c in the range 5.74–11.73 Hz. For SGE, Table 5 lists the twelve pairs (k, m_k) used by each branch, where k grows linearly and m_k by a factor of $\sqrt{2}$ per level, so that using all channels recovers the complete spectrum. The counts m_k of the waveform branch refer to the 1025 bands of the largest resolution. Retaining m_k coefficients keeps the spectral structure that repeats no faster than every $44100/(m_k - 1)$ Hz, which is 2.8 kHz at the coarsest level of the waveform branch and 61 Hz at its finest. Because every STFT resolution covers the same frequency range, this scale applies identically to all of them, and the ones with at most m_k bands are left unchanged by the mask. The SGE mask acts on the log-magnitude spectrogram of every resolution, retaining the first m_k DCT coefficients before exponentiating back, which is cepstral smoothing and yields the spectral envelope. For the waveform VAE, we construct separate targets for the magnitude and log-magnitude terms of the multi-resolution STFT (MR-STFT) loss by applying the temporal projection in their respective domains. At each frequency bin, the filtered magnitude trajectory $y[t]$ is shifted by $\max\{0, -\min_t y[t]\}$ to ensure non-negativity. This constant shift changes only the temporal DC component and therefore preserves the cutoff. The logarithmic target is computed independently and need not equal the logarithm of the corrected linear target. When both constraints are active, we apply SGE before TSE to the latent and to each reconstruction target. Temporal filtering and channel truncation commute in the latent space because they act on different axes. The corresponding target projections also commute in the logarithmic reconstruction term and in the spectrogram branch, where both operate linearly on different axes. For the linear reconstruction term of the waveform branch, the target transforms need not commute, and the joint objective uses the specified order.

Coarsened Targets for Adversarial Training The waveform VAE uses the multi-scale STFT discriminator of EnCodec (Défossez et al., 2022), which takes real and imaginary components as input. At each resolution, we construct a complex reference by replacing the magnitude with $\tilde{\mathcal{S}}(x)$ while retaining the original phase. Its real and imaginary components form the discriminator's real input, whose intermediate activations also provide the reference for feature matching. These targets are constructed independently at each STFT resolution, and the reconstruction $\hat{x}$ is evaluated without

Table 5: Channel counts k and the paired numbers of retained DCT coefficients m_k used by SGE.

Level	Waveform VAE		Spectrogram VAE	
	k (of 64)	m_k (of 1025)	k (of 40)	m_k (of 128)
1	16	17	10	3
2	20	24	12	4
3	24	33	15	5
4	28	46	18	7
5	32	65	20	9
6	36	92	22	12
7	40	129	25	17
8	44	182	28	23
9	48	257	30	33
10	52	363	32	46
11	56	513	35	65
12	60	725	38	91

further coarsening. For the spectrogram VAE, the transformed log-mel target $\tilde{\mathcal{S}}(x)$ is used directly as the real input.

Diffusion Model Training. The waveform diffusion model is the DiT of Stable Audio Open, conditioned on T5 text embeddings and timing information. It is trained on 30-second windows of the data described in Appendix A, whereas the official model uses 47-second windows. The spectrogram diffusion model is the flow-matching transformer of MMAudio, conditioned on video, text, and synchronization features. Both are trained from scratch on the latents of the corresponding VAE.

Inference. Following Stable Audio Open, the waveform branch uses DPM-Solver++ (3M SDE) with 100 steps and a classifier-free guidance scale of 7. Following MMAudio, the spectrogram branch uses the Euler solver with 25 steps and a guidance scale of 4.5, and its decoded mel spectrograms are converted to waveforms by the 44.1 kHz BigVGAN-v2 (Lee et al., 2022) vocoder of MMAudio for evaluation. Table 6 summarizes the training and inference settings of the four models, whose optimization settings follow the public configurations of the two systems.

Table 6: Training and inference settings of each model.

	Waveform		Spectrogram	
	VAE	Diffusion	VAE	Diffusion
Initialization	Pretrained	Random	Pretrained	Random
Training steps	500k	1M	500k	300k
Clip length	1.49 s	30 s	8 s	8 s
Batch size	64	64	64	512
Optimizer	AdamW	AdamW	Adam	AdamW
Learning rate	1.5×10^{-4}	5×10^{-5}	2.9×10^{-4}	1×10^{-4}
LR schedule	Inverse sqrt	Inverse sqrt	Constant	Step
Weight decay	10^{-3}	10^{-3}	0	10^{-6}
EMA	Yes	Yes	No	Yes
λ_{KL}	10^{-4}	–	10^{-6}	–
λ_{adv}	0.1	–	0.5	–
Sampler	–	DPM-Solver++	–	Euler
Sampling steps	–	100	–	25
Guidance scale	–	7	–	4.5

Schedule Ablation. The schedule ablation in Figure 3 (right) starts from the variants with TSE alone and SGE alone and changes one setting at a time. For TSE in the waveform branch, the lower end of the cutoff range is moved from its default of 5.38 Hz to 8.07 Hz or 2.69 Hz by drawing r from $\{1, \ldots, 4\}$ or $\{1, \ldots, 12\}$. In the spectrogram branch, it is moved from 5.74 Hz to 8.11 Hz or 2.75

Hz by drawing n from $\{66, \ldots, 95\}$ or $\{23, \ldots, 95\}$. For SGE, channel truncation is applied with probability 0.25 or 0.75 instead of the default 0.5. The variant without TSE and SGE corresponds to a truncation probability of 0 and a cutoff at the highest modulation frequency of the latent.

Equivariance Evaluation. The equivariance analysis in Figure 5 uses the AudioCaps test split and computes MEL-D as for Table 3. For TSE, eight cutoffs are evaluated up to the highest modulation frequency, retaining $n \in \{3, 5, \ldots, 17\}$ components of the waveform latent (1.35–10.77 Hz) and $n \in \{23, 47, 66, 87, 109, 130, 152, 173\}$ of the spectrogram one (2.75–21.47 Hz). The largest n keeps every component, while the other values reach below the training range of both models and, in the spectrogram case, above it. For SGE, k runs over the twelve levels of Table 5 together with the untruncated latent, and the matched target keeps the paired m_k DCT coefficients. At the 1024-point STFT of MEL-D, the waveform target is unchanged for $k \geq 56$, since m_k then reaches the 513 frequency bins of that resolution.

D Related Work

D.1 Audio Autoencoders and Latent Diffusion

Audio LDMs generate in the latent space of a frozen autoencoder, following the two-stage recipe established for images (Rombach et al., 2022). Early systems treat the mel spectrogram as an image, compressing it along both time and frequency with a 2D convolutional VAE and relying on a vocoder to recover the waveform (Liu et al., 2023; Huang et al., 2023b; Ghosal et al., 2023; Luo et al., 2023; Liu et al., 2024a; Majumder et al., 2024; Xue et al., 2024; Chen et al., 2024; Jeong et al., 2025; Zhang et al., 2026). Make-An-Audio 2 (Huang et al., 2023a) instead compresses the mel spectrogram only along time with a 1D convolutional VAE, a choice that later V2A systems adopt (Wang et al., 2024; Cheng et al., 2025a). Stable Audio (Evans et al., 2024a) applies a fully convolutional autoencoder to the raw waveform, so that the latent sequence is decoded directly to audio without a vocoder or spectrogram inversion, and such latents have since become the prevailing choice (Lee et al., 2024; Hung et al., 2024; Tian et al., 2025; Shan et al., 2025). The diffusion model was initially a U-Net, inherited from image LDMs for spectrogram latents and, in Stable Audio, from Moûsai (Schneider et al., 2024). Transformer backbones have since taken over, beginning with the feed-forward denoiser of Make-An-Audio 2 and the multimodal variants of recent V2A systems. Stable Audio 2 (Evans et al., 2024b) and Stable Audio Open (Evans et al., 2025) adopt a DiT, which EzAudio (Hai et al., 2024) and most later waveform-latent systems share. Work on the VAE itself has pursued fidelity, for example through perceptually motivated losses (Wang et al., 2025), while its objective of spectral reconstruction, adversarial, and KL terms has remained the same across all three representations and leaves the organization of the latent unspecified. LatentEQ instead returns to the autoencoding stage and specifies the organization of the latent through the existing reconstruction objective.

D.2 Structured Latent Spaces for Image and Video Generation

In visual generation, the organization of the latent has been shown to affect the diffusion model trained on it, a property termed diffusability (Skorokhodov et al., 2025) that measures how easily the distribution induced by an autoencoder can be learned. Frequency analyses trace much of this difficulty to the high-frequency components of the latent, which a diffusion model fits poorly (Lai et al., 2026). Regularizing the latent toward the features of a pretrained vision encoder, through direct alignment in VA-VAE (Yao et al., 2025) or auxiliary prediction in MAETok (Chen et al., 2025a), relieves the tension between reconstruction and generation that appears as the bottleneck widens. Such alignment constrains what the latent encodes rather than how its axes relate to the scales of the signal. Other methods shape the latent through its spectrum or its geometry instead. EQ-VAE (Kouzelis et al., 2025) requires the decoder to commute with scaling and rotation of the latent, SE-VAE (Skorokhodov et al., 2025) with downsampling, which suppresses spurious high-frequency content, and AF-LDM (Zhou et al., 2025) with fractional shifts by making the autoencoder alias-free. SS-VAE (Liu et al., 2025c) biases video latents toward low spatiotemporal frequencies and a few dominant channel modes through local correlation and masked reconstruction. Spectrum Matching (Ning et al., 2026) aligns the power spectra of image and latent and applies one spectral mask to both when the reconstruction loss is computed, so that their frequency bands correspond. Beyond

these, DC-AE 1.5 (Chen et al., 2025b) orders the channels so that a prefix of them already yields a coarse reconstruction, following nested dropout and Matryoshka representation learning (Rippel et al., 2014; Kusupati et al., 2022). Apart from channel ordering, these constraints are defined on the two spatial axes of a 2D latent, both of which the image shares, whereas an audio latent shares only its time axis with the signal and has no frequency dimension. LATENTEQ therefore pairs each coarsening of the latent with a matched coarsening of the target spectrogram, using modulation-frequency cutoffs along time and granularity levels across channels, so that the target supplies the frequency structure that the representation lacks.

D.3 Structured Latent Spaces for Audio Generation

In audio generation, the organization of the latent has only recently received attention. KVAE (Shutkin et al., 2026) and MiniMax H3 (MiniMax, 2026) follow VA-VAE in pulling the latent toward the representation of a frozen audio foundation model, whereas SAME (Parker et al., 2026) imposes semantic regularization through auxiliary tasks and replaces the VAE bottleneck with a lightly constrained one. Such regularization improves generation but, as in vision, does not relate the organization of the latent to acoustic scale. Other work constrains the structure of the latent directly. LatentFT (Wang & Huang, 2026) applies a DFT along the time axis of a diffusion autoencoder's latent and randomly masks frequency components during training, so that timescales become controllable, and shows that post-hoc filtering of pretrained representations fails to synthesize coherent audio. LAFA (Cheng et al., 2025b) masks spans of channels of a waveform VAE to induce channel locality, which steepens the latent spectrum toward a power law and speeds up DiT convergence. STAR-VAE (Liu et al., 2026) weights the KL term according to a structured topology, so that the latent space accommodates the hierarchy of audio instead of the flat geometry imposed by an isotropic Gaussian prior. Target-KL regularization (Seetharaman & Kumar, 2026) fixes the KL term to a target bitrate and uses the compression rate to trade reconstruction fidelity against the predictability of the latent. CAE (Torres et al., 2026) induces linearity in a consistency autoencoder (Pasini et al., 2024) through data augmentation alone, so that the decoder satisfies additivity and homogeneity, and mixing or scaling latents corresponds to the same operations on audio. These constraints either act on the latent alone or rest on diffusion and consistency decoders, and none pairs a transformation of the latent with a matched transformation of the acoustic target across temporal and spectral scales. LATENTEQ imposes exactly this relation on both axes, and does so by fine-tuning the standard VAE of an audio LDM under its original objective, without a new architecture or an additional loss term.

E Extended Background

E.1 Variational Autoencoders

We further elaborate on the variational autoencoder (VAE) (Kingma & Welling, 2013) formulation introduced in the preliminaries. Given an input x, a VAE models its latent representation as a random variable rather than a deterministic code. The generative model assumes a latent prior $p(z) = \mathcal{N}(0, I)$ and a conditional likelihood $p_\theta(x \mid z)$, which together define the following joint distribution:

$$p_\theta(x, z) = p(z)p_\theta(x \mid z). \tag{12}$$

Since the true posterior $p_\theta(z \mid x)$ is generally intractable, the encoder parameterizes the following tractable approximation:

$$q_\phi(z \mid x) = \mathcal{N}\big(\mu_\phi(x), \mathrm{diag}(\sigma^2_\phi(x))\big) \approx p_\theta(z \mid x). \tag{13}$$

A latent sample $z = \mathcal{E}(x) \in \mathbb{R}^{F\times d}$ is obtained with the reparameterization trick, which enables end-to-end optimization by backpropagation:

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{14}$$

The VAE maximizes the evidence lower bound (ELBO), which balances reconstruction fidelity and latent regularization:

$$\mathcal{L}_{\mathrm{ELBO}}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right] - D_{\mathrm{KL}}(q_\phi(z \mid x) \,\|\, p(z))\,. \tag{15}$$

The first term encourages the sampled latent z to retain sufficient information for the decoder to recover the input, while the KL term $\mathcal{L}_{\mathrm{KL}}(x) = D_{\mathrm{KL}}(q_\phi(z \mid x) \,\|\, p(z))$ regularizes the approximate posterior toward the shared Gaussian prior. This discourages different inputs from being mapped to arbitrary and disconnected regions and yields a continuous latent space. The latent therefore preserves enough information to reconstruct x while remaining constrained enough to stay well behaved. In practice, audio VAEs replace the likelihood term with the spectral reconstruction and adversarial losses of Eq. 2 and weight the KL term by a small coefficient λ_{KL}.

E.2 Latent Diffusion Models

Latent diffusion models (LDMs) perform generative modeling in the compressed representation of a pretrained autoencoder rather than directly in the high-dimensional input space (Rombach et al., 2022). Given an audio input x, the VAE encoder produces the latent sequence $z = \mathcal{E}(x) \in \mathbb{R}^{F \times d}$, which the decoder maps back to a waveform or spectrogram as $\hat{x} = \mathcal{D}(z)$. Once trained, the VAE is frozen, and the downstream diffusion model learns the distribution of the latents produced by $\mathcal{E}$. Operating on this lower-dimensional representation substantially reduces the computational cost of generative modeling while retaining the perceptually relevant information needed for reconstruction. This paradigm has been widely adopted for audio generation, using latents produced by either spectrogram-domain or waveform-domain autoencoders (Liu et al., 2023; 2024a; Evans et al., 2024a; 2025; Cheng et al., 2025a).

Latent Diffusion Process. We first describe the standard discrete-time diffusion formulation. We denote the clean VAE latent by $z_0 = z$. Diffusion defines a forward process that progressively corrupts z_0 with Gaussian noise over T noise levels (Ho et al., 2020; Nichol & Dhariwal, 2021). Let β_t denote the noise variance at step t, and define the following cumulative quantities:

$$\alpha_t = 1 - \beta_t, \qquad \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s. \tag{16}$$

The noisy latent at an arbitrary timestep can then be sampled directly as follows:

$$z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{17}$$

For small t, z_t remains close to the encoded audio latent, while increasing t progressively destroys its structure until the distribution approaches Gaussian noise. Continuous-time formulations describe the same process with stochastic differential equations (Song et al., 2020), and noise schedules and parameterizations vary across works (Nichol & Dhariwal, 2021; Karras et al., 2022).

The diffusion model learns to reverse this corruption process. Under a common noise-prediction parameterization, a denoising network ϵ_θ receives the noisy latent z_t and its noise level t and predicts the noise used to construct it. The denoising network is trained by minimizing the following objective:

$$\mathcal{L}_{\mathrm{diff}} = \mathbb{E}_{x,t,\epsilon}\left[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2\right], \qquad z_0 = \mathcal{E}(x). \tag{18}$$

Training samples t and ϵ directly, so the complete forward trajectory does not need to be simulated. Other diffusion parameterizations instead predict the clean latent, the velocity v, or the score while describing the same underlying denoising process (Song et al., 2020; Karras et al., 2022; Salimans & Ho, 2022). Stable Audio Open, used in our T2A experiments, adopts a v-parameterized diffusion objective (Evans et al., 2025). MMAudio, used in our VT2A experiments, instead adopts conditional flow matching, which learns a time-dependent velocity field that transports noise toward the latent data distribution (Cheng et al., 2025a). In our experiments, we retain the original training objective of each system.

At inference time, diffusion models begin from Gaussian noise and follow the learned reverse dynamics to obtain a clean latent. Flow-matching models similarly begin from Gaussian noise and integrate the learned velocity field to a clean latent. The resulting latent z is finally decoded as $\hat{x} = \mathcal{D}(z)$. Conditional generation additionally provides representations of text, video, or other modalities to the diffusion model, allowing it to produce latents consistent with the conditioning signal (Liu et al., 2023; Evans et al., 2024b; Cheng et al., 2025a). In modern LDMs, the backbone is often a transformer, as in the DiT (Peebles & Xie, 2023). For the audio latents considered here,

the DiT operates on $z_t \in \mathbb{R}^{F\times d}$ and models interactions across the temporal latent sequence while being conditioned on the timestep and, when applicable, external modalities.

Because the downstream diffusion model is trained on $z = \mathcal{E}(x)$ rather than directly on x, the VAE determines not only which information is preserved but also the distribution that the downstream model must learn. This motivates LATENTEQ, which improves the organization of the VAE latent while preserving its dimensionality, so that the diffusion model of each system is trained on the new latents with its original architecture and objective.

E.3 DISCRETE FOURIER TRANSFORM

We further explain the discrete Fourier transform (DFT) for a real-valued sequence $u = (u[0], \dots, u[J-1]) \in \mathbb{R}^J$. As in Eq. 4, the complex Fourier basis is defined as follows:

$$w_j[n] = e^{\mathrm{i}2\pi jn/J}, \qquad j, n \in \{0, \dots, J-1\}. \tag{19}$$

The basis vector w_j completes j oscillations over the J samples. The DFT coefficient associated with this basis vector is the following projection:

$$U[j] = u \cdot w_j = \sum_{n=0}^{J-1} u[n] e^{-\mathrm{i}2\pi jn/J}, \tag{20}$$

where $\cdot$ denotes the complex dot product and therefore conjugates its second argument.

The Fourier basis vectors are mutually orthogonal:

$$w_j \cdot w_\ell = J\,\delta_{j\ell}, \tag{21}$$

where $\delta_{j\ell}$ is the Kronecker delta. Consequently, the transform is invertible, and the original sequence is reconstructed exactly by the following inverse DFT:

$$u[n] = \frac{1}{J}\sum_{j=0}^{J-1} U[j] e^{\mathrm{i}2\pi jn/J} = \frac{1}{J}\sum_{j=0}^{J-1} U[j] w_j[n]. \tag{22}$$

Thus, the DFT expresses u as a superposition of complex sinusoids, with $U[j]$ specifying the magnitude and phase of the component at index j.

The coefficient index j determines the oscillation rate of the corresponding basis function. The component $j = 0$ is constant over the sequence and is commonly called the DC component.

For real-valued u, the DFT satisfies the following conjugate symmetry:

$$U[J-j] = \overline{U[j]}, \qquad j = 1, \dots, J-1. \tag{23}$$

The negative-frequency coefficients therefore contain no additional information. It is thus sufficient to retain $K = \lfloor J/2 \rfloor + 1$ independent coefficients, which form the one-sided representation produced by the real-valued DFT. Within this one-sided representation, the indices $j = 0, \dots, K-1$ are ordered from the slowest variation to the highest representable frequency, which is the Nyquist frequency when J is even. The DFT therefore separates variations in a sequence according to their scale.

When the J samples are uniformly distributed over a duration of D seconds, the index of a DFT coefficient can be expressed in physical frequency. Since the jth basis completes j cycles over the duration D, its frequency is given by the following expression:

$$f_j = \frac{j}{D} \quad \text{Hz}. \tag{24}$$

Adjacent DFT coefficients are therefore separated by $1/D$ Hz. In particular, when the transform is applied along the time axis of a sequence, f_j measures the frequency of variation of that sequence over time. For example, slowly changing temporal structure is represented by low-index coefficients, whereas rapid changes require higher-index coefficients.

A useful consequence of this ordering is that the sequence can be coarsened by removing high-frequency Fourier components. Let $m \le K$ denote the number of retained one-sided coefficients.

Setting the coefficients with index $j \geq m$ to zero and applying the inverse transform produces a low-pass version of the sequence, which corresponds to the cutoff $f_c = (m-1)/D$ Hz in Eq. 8. Equivalently, in the full DFT, the corresponding positive- and negative-frequency components are removed together so that conjugate symmetry is preserved.

Because the Fourier basis is orthogonal, this operation can be viewed as a projection onto the subspace spanned by the retained Fourier components. Retaining only low-index coefficients restricts the reconstructed sequence to slowly varying functions. As more coefficients are retained, increasingly rapid variation can be represented, and retaining all coefficients exactly recovers the original sequence. Fourier truncation therefore provides a natural hierarchy from coarse temporal variation to fine temporal detail.

E.4 Discrete Cosine Transform

The DCT (Ahmed et al., 1974) provides a closely related decomposition using only real-valued cosine functions. We use the orthonormal DCT-II basis $\{c_j\}_{j=0}^{J-1}$, whose elements are defined as follows:

$$c_j[n] = \alpha_j \cos\left(\frac{\pi}{J}\left(n + \frac{1}{2}\right) j\right), \tag{25}$$

where $\alpha_0 = 1/\sqrt{J}$ and $\alpha_j = \sqrt{2/J}$ for $j > 0$. With this normalization, the basis vectors satisfy the following orthonormality relation:

$$c_j \cdot c_\ell = \delta_{j\ell}. \tag{26}$$

The DCT coefficient at index j is therefore the following projection:

$$V[j] = u \cdot c_j = \sum_{n=0}^{J-1} u[n] c_j[n]. \tag{27}$$

The inverse DCT reconstructs the sequence as follows:

$$u[n] = \sum_{j=0}^{J-1} V[j] c_j[n]. \tag{28}$$

As with the DFT, the DCT basis is ordered by oscillation rate. The zeroth basis vector c_0 is constant. Accordingly, $V[0] = \sqrt{J}\,\bar{u}$ is proportional to the sample mean, and its reconstructed component $V[0]c_0$ is the constant mean level of the sequence. Low-index basis vectors vary gradually over n, whereas higher-index ones oscillate increasingly rapidly, so low-order DCT coefficients represent broad and smooth variation and high-order coefficients represent finer local variation.

Unlike the DFT, the DCT contains only real-valued coefficients and does not require a conjugate-symmetric representation. All J coefficients are independent. The DCT can be interpreted as a Fourier representation of an even-symmetric extension of the finite sequence. This differs from the DFT, which treats the sequence as one period of a periodically repeated signal. The even extension avoids directly connecting the last sample to the first and is particularly useful for representing smoothly varying finite profiles.

As with the DFT, truncating the DCT coefficients produces a controlled coarsening of the sequence. Retaining only the first m coefficients gives the following truncated sequence:

$$u^{(m)} = \sum_{j=0}^{m-1} V[j] c_j. \tag{29}$$

Because the DCT basis is orthonormal, $u^{(m)}$ is the orthogonal projection of u onto $\mathrm{span}\{c_0, \ldots, c_{m-1}\}$. For small m, only slowly varying structure can be represented and the result is correspondingly smooth. Increasing m introduces progressively finer variation, while $m = J$ recovers u exactly.

When the DCT is applied to the frequency axis of a log-magnitude spectrogram, its coefficients form the cepstrum, and their interpretation should be distinguished from acoustic frequency itself. Consider one frame of log-magnitudes $s \in \mathbb{R}^B$, whose index already enumerates acoustic frequency

bands. A DCT across these B bands describes how rapidly the spectral representation varies along the frequency-band axis. Low-order DCT coefficients describe slowly varying spectral structure, such as a broad spectral envelope, whereas higher-order coefficients describe increasingly fine variation across neighboring frequency bands.

Thus, retaining low-order DCT coefficients does not correspond to retaining only low acoustic frequencies. Instead, it preserves a coarse description over the entire frequency range. A spectral representation reconstructed from a small number of DCT coefficients can still describe content at both low and high acoustic frequencies, but its variation across frequency is constrained to be smooth. As more DCT coefficients are included, narrower peaks, harmonic structure, and other fine spectral details can be represented. This granularity is what m_k controls in Eq. 11.