Improving the Diffusability of Autoencoders

Ivan Skorokhodov[1] **Sharath Girish**[1] **Benran Hu**[1 2] **Willi Menapace**[1] **Yanyu Li**[1] **Rameen Abdal**[1]
Sergey Tulyakov[1] **Aliaksandr Siarohin**[1]

Abstract

Latent diffusion models have emerged as the leading approach for generating high-quality images and videos, utilizing compressed latent representations to reduce the computational burden of the diffusion process. While recent advancements have primarily focused on scaling diffusion backbones and improving autoencoder reconstruction quality, the interaction between these components has received comparatively less attention. In this work, we perform a spectral analysis of modern autoencoders and identify inordinate high-frequency components in their latent spaces, which are especially pronounced in the autoencoders with a large bottleneck channel size. We hypothesize that this high-frequency component interferes with the coarse-to-fine nature of the diffusion synthesis process and hinders the generation quality. To mitigate the issue, we propose *scale equivariance*: a simple regularization strategy that aligns latent and RGB spaces across frequencies by enforcing scale equivariance in the decoder. It requires minimal code changes and only up to 20K autoencoder fine-tuning steps, yet significantly improves generation quality, reducing FID by 19% for image generation on ImageNet-1K 256^2 and FVD by at least 44% for video generation on Kinetics-700 17×256^2. The source code is available at https://github.com/snap-research/diffusability.

1. Introduction

In recent years, diffusion models (DMs) have emerged as the dominant generative modeling paradigm in computer vision. However, the high dimensionality of visual data poses a significant challenge, making the direct application
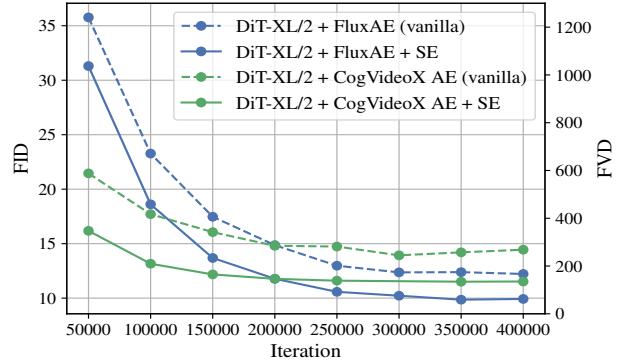

Figure 1. Convergence speed of DiT-XL/2 on top of vanilla FluxAE vs FluxAE fine-tuned for 10K steps with scale equivariance (SE) regularization on ImageNet-1K-256^2; and on top of CogVideoX-AE vs CogVideoX-AE with SE on Kinetics-700-17×256^2. Our regularization improves the performance of image and video LDMs by refinng the frequency profile of their autoencoders' latent spaces.

of diffusion models impractical. Latent diffusion models (LDMs) (Vahdat et al., 2021; Rombach et al., 2022) have become the main approach in mitigating this issue, demonstrating remarkable success in generating high-resolution images (Black Forest Labs, 2023; Betker et al., 2023; Esser et al., 2024) and videos (Brooks et al., 2024; Yang et al., 2024; Kong et al., 2024). A typical LDM consists of two main components: an autoencoder and a diffusion backbone. Most recent breakthroughs have been driven by scaling up diffusion backbones (Peebles & Xie, 2022), while autoencoders (AEs) have received comparatively less attention.

Recently, the research community has begun focusing more on improving the autoencoders, recognizing their crucial impact on overall performance, but most effort has been concentrated on enhancing reconstruction quality (Black Forest Labs, 2023; Hong et al., 2022; Agarwal et al., 2025; HaCohen et al., 2024; Chen et al., 2025) and achieving higher compression ratios (Agarwal et al., 2025; HaCohen et al., 2024; Chen et al., 2025) to accelerate the diffusion process. However, we argue that a critical yet under-explored aspect, which we refer to as *diffusability*[1], also plays a key role in

---

[1]*Diffusability* describes how easily a distribution can be modeled by diffusion: high *diffusability* indicates that the distribution is easy to fit, while low diffusability implies a higher difficulty.

determining the utility of autoencoders. Indeed, all three factors—reconstruction quality, compression efficiency, and diffusability—are essential for the practical effectiveness of LDMs. Specifically, inaccurate reconstruction sets an upper bound on generation fidelity, low compression efficiency leads to slow and costly generation, and poor diffusability necessitates the use of heavier, more expensive, and sophisticated diffusion backbones, further limiting LDM quality.

Diffusion models possess a unique property of being coarse-to-fine in nature (Dieleman, 2024; Ning et al., 2024): in the denoising process, they synthesize low-frequency signal components first and add high-frequency ones on top of them later. It is a beneficial trait since it allows to defer error accumulation to higher frequency parts of the spectrum, which aligns well with how humans perceive quality: we are sensitive to the image structure and composition, but oblivious of its fine-grained textural details. However, when applying the diffusion process in the latent spaces of pre-trained autoencoders, the correspondence between latent low-frequency components and their RGB counterparts may be lost, hindering the spectral autoregression property.

In this work, we identify a correlation between the spectral properties of the latent space and its diffusability. We analyze the spectral characteristics of latent representations across several widely used image and video autoencoders. Our investigation reveals a prominent high-frequency component in these latent spaces, deviating significantly from the spectral distribution of RGB signals. This component becomes even more pronounced as the channel size increases, which the recent autoencoders use to improve reconstruction. We hypothesize that the flat spectral distribution induced by the strong high-frequency component harms the spectral autoregression property. Moreover, we demonstrate that these high-frequency components substantially influence the final RGB result, and their inaccurate modeling can introduce noticeable visual artifacts. Finally, we show that standard KL regularization is insufficient to address spectrum defects and, in some cases, may even amplify the issue.

To mitigate these spurious high-frequency components in latent representations, we propose a simple and effective regularization strategy. Our approach involves aligning the latent space and the RGB space at different frequencies. This is achieved by enforcing scale equivariance in the decoder — ensuring that downsampled latents correspond to downsampled RGB representations. Our method requires minimal modifications and few additional autoencoder fine-tuning steps, yet significantly enhances diffusability across various architectures, ultimately improving the quality of generated samples. We validate our approach on both image and video autoencoders, including FluxAE (Black Forest Labs, 2023), CosmosTokenizer (Agarwal et al., 2025), CogVideoX-AE (Hong et al., 2022), and LTX-AE (HaCo-hen et al., 2024), consistently demonstrating improved LDM performance on ImageNet-1K (Deng et al., 2009) 256^2, reducing FID by 19% for DiT-XL, and Kinetics-700 (Carreira et al., 2019) 17×256^2, reducing FVD by at least 44%.

2. Related Work

Diffusion models. Diffusion models (Vahdat et al., 2021; Rombach et al., 2022; Song et al., 2021; Ho et al., 2020; Nichol & Dhariwal, 2021; Karras et al., 2022) have emerged as the dominant framework for generative modeling, surpassing traditional approaches like GANs (Goodfellow et al., 2014; Karras et al., 2019) and VAEs (Kingma & Welling, 2013). Diffusion models express generation as a denoising process producing the generated content by progressively denoising an initial noise sample. Owing to their efficiency and scalability, foundational generative models (Saharia et al., 2022; Ho et al., 2022; Yang et al., 2024; Podell et al., 2023; Blattmann et al., 2023; Polyak et al., 2024) have made significant strides in synthesizing visually stunning and semantically aligned images and videos.

Initially applied to low-resolution visual content in the pixel space (Vahdat et al., 2021; Ho et al., 2020; Nichol & Dhariwal, 2021; Karras et al., 2022), they have soon been extended to higher resolutions. In Latent Diffusion Models (LDMs) (Vahdat et al., 2021; Rombach et al., 2022) high-resolution visual content is modeled in the compact latent space produced by a variational autoencoder (VAE) (Kingma & Welling, 2013) within a two-stage framework. Latent Flow Models (LFMs) (Dao et al., 2023; Liu et al., 2024), follow the same approach but leverage Rectified Flows (RFs) to enable faster and more stable sampling.

Recent work attributes the success of diffusion models to a form of implicit spectral autoregression (Rissanen et al., 2023; Ning et al., 2024) implied by the progressive removal of noise during sampling, resulting in the generation of visual content in a coarse-to-fine manner. Such result holds in the pixel-space of natural images, based on its pattern of decreasing spectral power (Ruderman, 1997). We show that popular autoencoders (Black Forest Labs, 2023; Agarwal et al., 2025; HaCohen et al., 2024; Yang et al., 2024) have a less pronounced pattern of decreasing spectral power, inhibiting implicit spectral autoregressive generation. Building on this observation, this work proposes a regularization scheme that re-establishes this property, consistently showing improved LDM performance and avoiding the need for explicit coarse-to-fine generation.

Image and video autoencoders. Due to the success of LDMs, a lot of effort has been devoted to the development of better AEs. Image LDMs (Rombach et al., 2022) and early video diffusion models (Blattmann et al., 2023; Guo et al., 2023) employ a spatial AE with a compression ra-

tio of $1\times8\times8$. The rapid advancement of video diffusion models poses the demand for 3D AEs that jointly compress spatial and temporal dimensions to further improve efficiency (Hong et al., 2022; Zhou et al., 2024; Kong et al., 2024). Among them, Open-Sora (Zheng et al., 2024) inherits the $1\times8\times8$ spatial AE and trains a decoupled $4\times$ temporal AE on top of its latent space, while others tend to build a hierarchical spatio-temporal AE with 3D causal convolutions (Xing et al., 2024; Wu et al., 2024; Chen et al., 2024a; Zhao et al., 2024; Sadat et al., 2024; Hansen-Estruch et al., 2025). To accelerate the model and improve the reconstruction, Open-Sora-Plan (Lin et al., 2024) and CosmosTokenizer (Agarwal et al., 2025) propose to employ the wavelet transform of the input. Another popular trend is to further increase the compression ratio to reduce the number of tokens in the latent space (Xie et al., 2024; Tian et al., 2024; HaCohen et al., 2024), thus enabling a more efficient denoising process. In addition to the continuous AEs explored in this work, multiple discrete AEs (Wang et al., 2024; Tang et al., 2024; Agarwal et al., 2025) are proposed to aid autoregressive tasks. Esteves et al. (2024) leverages a wavelet transform to produce latents corresponding to different frequency components.

AEs for compression. Many works train neural-based AEs for image compression (Ballé et al., 2016; 2018; Minnen et al., 2018; Cheng et al., 2020), typically with 16×16 downsampled latents which are discrete and entropy constrained. Video compression AEs involve autoregressive AEs (Li et al., 2021; 2023; Sheng et al., 2022) with explicit framewise formulations that utilize motion vectors or implicit modeling (Mentzer et al., 2022). These approaches target high-quality reconstruction with low bitrates and adopt complex designs for learnable entropy models. They typically employ a larger number of latent bottleneck channels ($96-192$), which is not generally suited for the generation task, thus we do not consider them in this work.

Concurrent works. Independently from us, AF-LDM (Zhou et al., 2025) enforces shift equivariance in both the autoencoder and LDM, and EQ-VAE (Kouzelis et al., 2025) proposes scale/shift equivariance regularization for autoencoders, but with a different motivation of improving the models' equivariance to spatial transformations.

3. Improving Diffusability

We begin this section by discussing the spectral decomposition of 2D signals and providing some background on discrete cosine transform in Section 3.1. In Section 3.2, we analyze the spectral properties of latent spaces across different autoencoders and compare them to those of the RGB space. Our main insight is that the frequency profile of the latent space includes large-magnitude high-frequency components. We also show that as the channel size increases, the
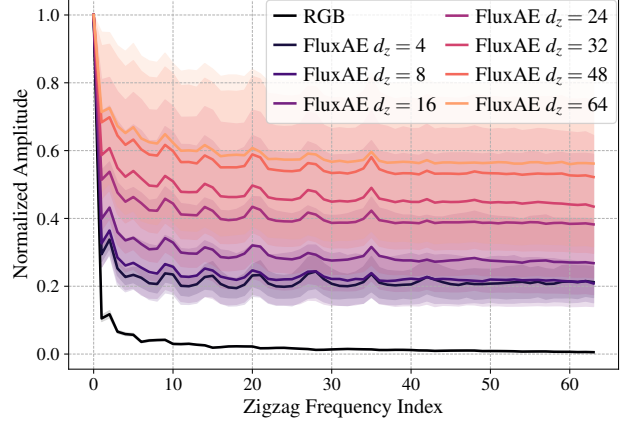

Figure 2. Latent frequency profiles of FluxAE autoencoders of varying bottleneck sizes, and also RGB (of the same 32^2 spatial dimension). One can notice two things: 1) the latent space of an autoencoder exhibits a different power profile from RGB; and 2) high frequency amplitudes increase with the latent channel size.

high-frequency components become more pronounced. Additionally, we demonstrate that the widely adopted KL regularization only increases the strength of these components. Finally, Section 3.3 presents a straightforward method to improve the diffusability of a latent space of an autoencoder by enhancing its spectral properties.

3.1. Background: Blockwise 2D DCT

The discrete cosine transform (DCT) (Ahmed et al., 2006) over a 2D signal is a transformation converting the signal's representation between the spatial and frequency domains. DCT, in particular, represents the original input signal as coefficients for a set of horizontal and vertical cosine basis oscillating with different frequencies. More formally, given a 2D signal block $\mathbf{A} \in \mathbb{R}^{B \times B}$ whose values P_{xy} denote the pixel intensity at position (x, y), the two-dimensional type-II DCT yields a frequency-domain block $\mathbf{D} \in \mathbb{R}^{B \times B}$ where D_{uv} captures the coefficient for the corresponding horizontal and vertical cosine bases:

$$D_{uv} = \alpha(u)\alpha(v) \sum_{x=0}^{B-1} \sum_{y=0}^{B-1} P_{xy} f(x, u) f(y, v),$$

$$\text{where} \quad \alpha(u) = \begin{cases} \sqrt{1/B}, & u = 0, \\ \sqrt{2/B}, & u \neq 0, \end{cases}$$

$$f(x, u) = \cos\left(\frac{(2x+1)u\pi}{2B}\right).$$

In practice, we split the input 2D signal into non-overlapping blocks of size $B \times B$ and treat each channel independently.

By analyzing RGB images and latents in the DCT frequency domain, we produce a *frequency profile* that relates to the energy of the signal at every frequency. A zigzag frequency index is used to map each DCT block $\mathbf{D} \in \mathbb{R}^{B \times B}$ into a
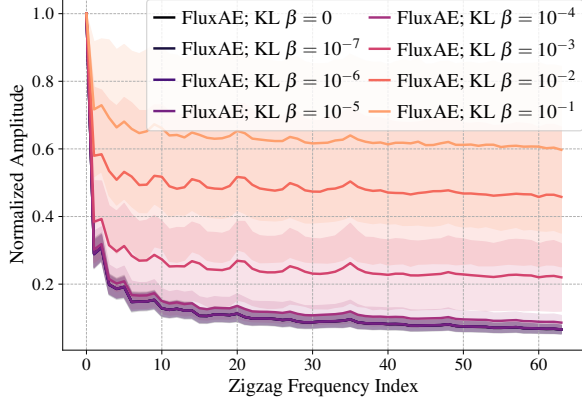
Figure 3. Spectrums for FluxAE autoencoders trained (from scratch) with different KL regularization strengths. KL regularization is a double-edged sword: it pushes the latents distribution closer to standard Gaussian (the distribution the reverse diffusion process starts with), so that the LDM has less work to do (Vahdat et al., 2021), but it also introduces high-frequency components into the latents due to the random noise addition (see Figure 13), which LDM is forced to model as well. Figure 13 shows the influence of noise addition on the frequency profile.

one-dimensional sequence following the standard *zigzag* ordering as in JPEG (Wallace, 1991), which indexes the DCT coefficients from lowest frequency $D_{0,0}$ to highest one $D_{B-1,B-1}$. Formally, let $\mathrm{zigzag}(u,v) \in \{0, \ldots, B^2 - 1\}$ denote the ranks of the coefficients D_{uv} in the ascending frequency order. Given a block, we compute its DCT and produce the normalized amplitudes for each frequency component (u, v) as:

$$A_{uv} = \left| \frac{D_{uv}}{D_{0,0}} \right|. \tag{1}$$

We define the *frequency profile* as the sequence of normalized amplitudes in the standard *zigzag* order.

When analyzing the frequency profiles of videos (or latent codes with an additional time dimension), we still rely on per-frame 2D DCT since the temporal and spatial domains possess different spectral properties.

3.2. Spectral Analysis of the Latent Space

We begin our analysis by observing the frequency profile of the latent space in the Flux (Black Forest Labs, 2023) family of autoencoders to establish a relationship with *diffusability*. For the purpose of this study, we train a family of FluxAE models with various channel sizes for 100k steps (where performance saturates in this setting) and, for each of them, compute the averaged frequency profile over 256 samples, all channels, and all DCT blocks. Figure 2 presents the frequency profiles of both Flux autoencoders and RGB space, from which we observe: (i) The Flux profile exhibits significantly larger high-frequency components compared
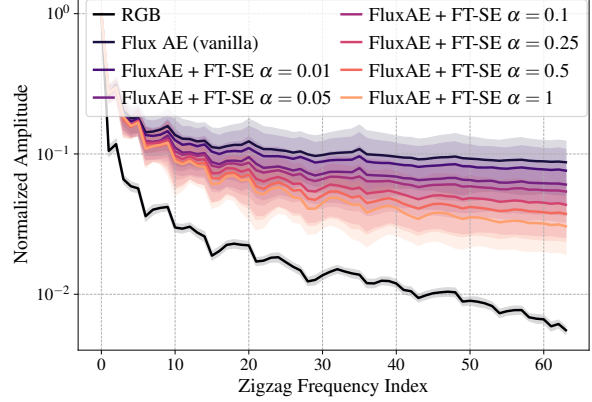

Figure 4. DCT Spectrum of the FluxAE latents with and without scale equivariance (SE) regularization. Fine-tuning AEs with SE brings the spectrum closer to the RGB domain, the higher the regularization strength.

to the RGB profile. (ii) As the number of channels in the autoencoder's bottleneck increases, high-frequency components become more pronounced. We hypothesize that a larger bottleneck allows the autoencoder to capture finer, high-frequency details. Initially, limited capacity prioritizes smoother, low-frequency information. But as the capacity expands, the model encodes additional high-frequency content, distributing it unevenly and in the unstructured manner across channels. This finding is of particular interest as the number of channels is positively correlated with autoencoder's reconstruction quality.

A common way to regularize the latent space for latent diffusion models (LDMs) is to employ a variational autoencoder (VAE) (Kingma & Welling, 2013) framework with a KL divergence term, encouraging the latent distribution to align with the standard Gaussian prior. Since the reverse diffusion process also starts with the same standard normal distribution, such KL regularization is argued to simplify the job for the diffusion model (Vahdat et al., 2021), since it now has less "work" to do. However, as we show in Figure 3 which compares FluxAE with varying levels of KL regularization, higher KL regularization introduces more high frequencies due to the underlying noise addition process, creating a harmful side-effect, reducing diffusability.

Previous work (Ning et al., 2024; Dieleman, 2024; Rissanen et al., 2023) interprets diffusion models as autoregressive ones, but in the spectral domain: when noise level is high, low frequencies are generated, then, as the level of noise lowers during sampling, progressively higher frequencies are generated. This is an attractive property since it allows the model to leverage the cleaner lower frequencies as a conditioning signal for the current prediction. However, the strength of this autoregressive pattern is directly related to the shape of the frequency profile for the signal to generate. Since the white noise that is applied as part of the diffusion process has a flat frequency profile, it follows that the flatter
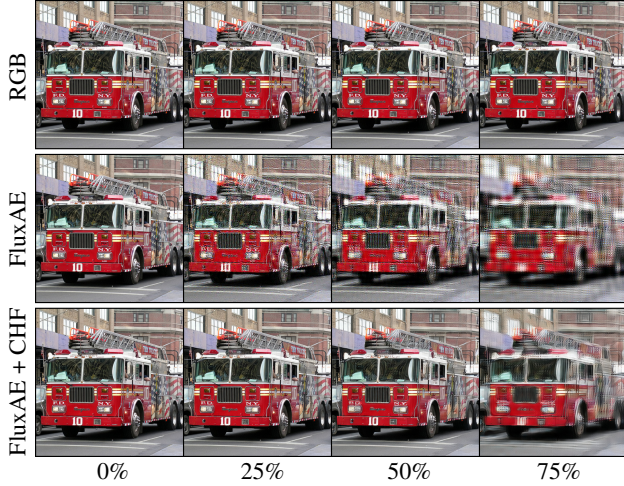
Figure 5. RGB and autoencoder reconstructions with progressively erased DCT high-frequency components. RGB faces minimal degradation (top), as a higher percentage of the latent DCT spectrum is removed, but the Flux AE reconstructions (middle) quickly degrade when the high-frequency components from the latents are being removed. A high-frequency cutoff regularization forces the autoencoder to rely more on the low frequency region of the latents and leads to better compression and resilience to high-frequency error accumulation in diffusion models.

the frequency profile of the signal, the lower the cleanliness of low frequencies that can act as conditioning for the model. For a flat frequency profile, no autoregressive generation is possible as all frequencies would be erased at the same speed by white noise. We also hypothesize that higher frequencies components are harder to model than lower frequency components for the following reasons and thus should be avoided: (i) they have higher dimensionality[2]; (ii) they are generated only in the final steps of sampling, thus must emerge more rapidly; (iii) they are more susceptible to error accumulation over time (Li & van der Schaar, 2024).

Motivated by this analysis, we propose scale equivariance regularization for the autoencoder's latent space.

3.3. Scale Equivariance Regularization

Effective regularization should achieve two key objectives: (i) to suppress high-frequency components in the latent space and (ii) to prevent the decoder from amplifying these components, as their impact on the final result is what ultimately matters. This can be accomplished by aligning the spectral properties of the latent and RGB spaces at different frequencies. A way to achieve this consists in explicitly chopping off a portion of the high frequencies in both spaces and training the decoder to reconstruct the truncated RGB signal from the truncated latent representation. Our preliminary experiments demonstrated that an autoencoder can

---

[2]E.g., going from 256^2 to 512^2 resolution adds little structural content, but increases the dimensionality 3 times.

easily learn to alter its latent frequency profile to encode the inputs in the low-frequency region of the spectrum without sacrificing the reconstruction quality much (see Figure 5). While this regularization, which we name Chopping High Frequencies (CHF), improves the spectrum, we develop a much simpler procedure to achieve the same effect without the need to perform the error-prone DCT transform (the details of CHF are described in Appendix C). The simplest way to achieve high-frequency truncation is through direct downsampling, which we discuss next.

Downsampling involves resizing both the input x and the latent representation z by a fixed scale, yielding $\tilde{x}$ and $\tilde{z}$, respectively. This process effectively removes a portion of the high-frequency components from both the RGB and latent signals. In practice, we use $\times 2 - 4$ bilinear downsampling for all the experiments. Regularization is then enforced by ensuring that $\tilde{x}$ and the decoder's reconstruction of the downsampled latent $\text{Dec}(\tilde{z})$ remain consistent through an additional reconstruction loss. The autoencoder is trained using the following objective:

$$\mathcal{L}(x) = d(x, \text{Dec}(z)) + \alpha d(\tilde{x}, \text{Dec}(\tilde{z})) + \beta \mathcal{L}_{\text{KL}}. \quad (2)$$

Here, $d(\cdot, \cdot)$ represents a distance measure for reconstruction which we instantiate as mean squared error loss and perceptual losses (Zhang et al., 2018) following prior work (Rombach et al., 2022), α is the regularization strength (we use $\alpha = 0.25$ for the main experiments). The term $\mathcal{L}_{\text{KL}}$ is VAE's (Kingma & Welling, 2013) KL regularization, if applicable (we do not use it when we train with our regularization). This regularization effectively enforces scale equivariance in the decoder, which is the basis for its name.

In Figure 4, we illustrate the effect of scale equivariance on the spectrum of FluxAE. Our proposed regularization effectively reduces the high-frequency components of the signal, bringing it closer to the spectral characteristics of the RGB space. This successfully achieves objective (i) for effective regularization. Meanwhile, Figure 8 demonstrates that scale equivariance preserves more content compared to the baseline, as more and more high-frequency components are suppressed, thereby fulfilling objective (ii). Finally, Figure 6 visualizes intermediate steps in the diffusion trajectory. The regularized model exhibits a noticeably smoother and more structured progression, following a healthier coarse-to-fine generation process.

4. Experiments

Data. We trained all the autoencoders on in-the-wild data which do not overlap with ImageNet-1K (Deng et al., 2009) or Kinetics-700 (Carreira et al., 2019) to make sure that there is no data leak in the autoencoders, and that they remain general-purpose. For this, we used our internal image and video datasets of the 256^2 resolution, which
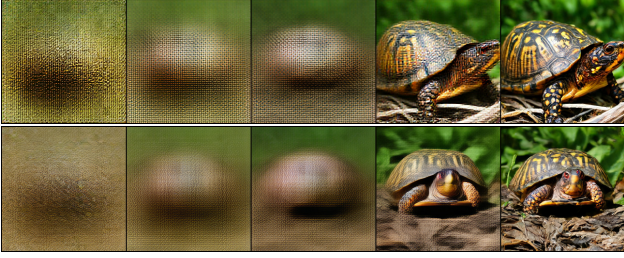
Figure 6. Denoising trajectories (steps 1, 16, 32, 128 and 256 out of 256) for DiT-XL trained with FluxAE (top) and FluxAE+SE. DiT-XL with vanilla FluxAE exhibits prominent high-frequency artifacts early on in the trajectory.

Table 1. Quantitative performance on ImageNet (Deng et al., 2009) without guidance. The original DiT scores are provided for reference from Table 4 of (Peebles & Xie, 2022).

Stage II	Stage I	FID ↓	FDD ↓
DiT-B/2	FluxAE (vanilla)	25.41	536.2
	FluxAE + FT	30.51	575.4
	FluxAE + FT-SE (ours)	18.06	450.6
DiT-L/2	FluxAE (vanilla)	12.42	306.73
	FluxAE + FT	14.48	333.54
	FluxAE + FT-SE (ours)	9.61	236.43
DiT-XL/2	FluxAE (vanilla)	12.21	282.8
	FluxAE + FT	10.62	262.2
	FluxAE + FT-SE (ours)	9.85	235.8
	+1M steps	3.27	85.86
DiT-B/1	$CMS\text{-}AE_I$ (vanilla)	11.69	360.83
	$CMS\text{-}AE_I$ + FT	13.59	375.19
	$CMS\text{-}AE_I$ + FT-SE (ours)	11.85	354.22
DiT-B/2 (orig)	SD-VAE-ft-MSE	43.47	—
DiT-L/2 (orig)		23.33	—
DiT-XL/2 (orig)		19.47	—
+ 7M steps (orig)		12.03	—

are similar in distribution of concepts and aesthetics to the publicly available in-the-wild datasets like COYO (Byeon et al., 2022) and Panda-70M (Chen et al., 2024b). To control for the impact of the data (and also the training recipe), we trained a separate autoencoder baseline for each setup without using our proposed regularization. In several cases, just fine-tuning on such in-the-wild data already yields better diffusion performance (e.g., see DiT-XL results in Table 1).

Evaluation. We evaluate image DiT models via FID and FDD (Frechet Distance computed on top of DINOv2 (Oquab et al., 2023) features), where the latter was shown to be a more reliable metric (Stein et al., 2024; Karras et al., 2024). We evaluate video DiT models with FVD_{10K} (Unterthiner et al., 2018), FID, and FDD, except for ablations where we rely on 5,000 samples. For image models, we use 50,000 samples without any optimization for class balancing. To evaluate autoencoders, we used PSNR, SSIM, LPIPS and FID metrics computed on 512 samples from ImageNet and Kinetics-700 for image and video autoencoders, respectively.

Training details. All the LDM models are trained for 400k steps with 10k warmup steps following the rectified flow diffusion parametrization (Albergo & Vanden-Eijnden, 2022; Dao et al., 2023; Esser et al., 2024). Following Esser et al. (2024), we use a logit-normal training noise distribution. We use either 2×2 or 1×1 patchification in DiT (Peebles & Xie, 2022) to match the compute between DiTs trained on top autoencoders with different compression ratios. Our video DiT is a direct adaption of the image one where we additionally unroll the temporal axis, following the recent works on video diffusion models (Yang et al., 2024). We do not use patchification for the temporal axis in video DiTs. In contrast to prior work (e.g., Rombach et al. (2022)), we average the KL loss across the latent channels and resolutions: this has no theoretical impact, but allows to compare autoencoders with different bottleneck sizes.

Inference details. We run DiT inference with 256 steps without classifier-free guidance (Ho & Salimans, 2022) for quantitative evaluations since different models are too sensitive to it and should be tuned separately (Karras et al., 2024; Kynkäänniemi et al., 2024).

4.1. Improving Existing Autoencoders

We apply our training pipeline on top of 3 different autoencoders. For each autoencoder, we trained it while freezing the last output layers to avoid breaking their adversarial fine-tuning, which should have no impact on the latent space (Chen et al., 2025). We emphasize that *none* of the explored modern autoencoders publicly released their training pipelines. For the pretrained snapshots of autoencoders, we used the original snapshots available in the `diffusers` library (von Platen et al., 2022).

Improving image autoencoders. For the image autoencoder, we used FluxAE (Black Forest Labs, 2023) with 8×8 compression ratio and 16 latent channels (since it is the most popular modern autoencoder in the community) and $CMS\text{-}AE_I$ (Agarwal et al., 2025) with 16×16 compression ratio and 16 channels as a high-compression autoencoder. For all the experiments (unless stated otherwise), we fine-tuned it for just 10,000 training steps with a batch size of 32 (320,000 total seen images) using $2\times$ and $4\times$ downsampling ratios, chosen randomly during a forward pass of the regularization loss. DiT training on top of the unchanged Flux Autoencoder is labeled as "vanilla". Autoencoders fine-tuned for 10,000 steps with our proposed SE regularization is denoted via the "+ FT-SE" suffix. To control for the fine-tuning data and training pipeline, we fine-tuned each autoencoder without adding our regularization as an additional loss (denoted via "+ FT").

For the LDM benchmark, we utilized ImageNet (Deng et al., 2009) at 256×256 resolution. We used the DiT (Peebles & Xie, 2022) model as the backbone since it is the most

Figure 7. Uncurated samples on ImageNet 256×256 from DiT-XL trained on top of FluxAE (top) vs DiT-XL with "FluxAE + FT-SE" (bottom). 256 steps with the guidance scale of 3.0. More visualizations are in Appendix D.

Table 2. Results on Kinetics-700 (Carreira et al., 2019) for DiT trained on top of various autoencoders. See Section 4.1 for details.

Stage II	Stage I	FVD$_{10K}$ ↓	FDD ↓	FID$_{10K}$ ↓
	CV-AE (vanilla)	650.40	650.97	28.85
DiT-B/2	CV-AE + FT	447.26	593.02	19.45
	CV-AE + FT-SE (ours)	252.26	466.15	12.19
	CV-AE (vanilla)	268.26	407.23	12.02
DiT-XL/2	CV-AE + FT	270.66	402.91	12.78
	CV-AE + FT-SE (ours)	135.15	245.27	8.59
	LTX-AE (vanilla)	854.47	814.49	50.99
DiT-B/1	LTX-AE + FT	876.61	823.71	50.18
	LTX-AE + FT-SE (ours)	389.56	642.80	22.88

Table 3. Ablating KL regularization weight β for FluxAE fine-tuning in terms of the reconstruction quality and downstream DiT-S/2 and DiT-L/2 (Peebles & Xie, 2022) training. Increasing the KL in general improves the LDM's performance for smaller models, but at the expense of worsened AE reconstruction (and reduced training stability), which can limit scalability (also observed by (Esser et al., 2024)) and results in worse performance of DiT-L/2. Our SE regularization leads to improved LDM performance without hurting the reconstruction and scales well to larger models.

Method	DiT-S/2 FDD$_{5K}$	DiT-L/2 FDD$_{5K}$	AE PSNR$_{512}$
FluxAE (vanilla)	992.05	415.87	30.20
+ KL $\beta = 0$	968.26	472.08	29.97
+ KL $\beta = 10^{-7}$	1018.6	425.35	30.29
+ KL $\beta = 10^{-6}$	1095.2	612.12	19.66
+ KL $\beta = 10^{-5}$	940.13	403.99	29.21
+ KL $\beta = 10^{-4}$	974.67	404.61	30.22
+ KL $\beta = 10^{-3}$	982.91	425.24	29.51
+ KL $\beta = 10^{-2}$	1946.5	1737.47	10.82
+ KL $\beta = 10^{-1}$	929.58	472.74	23.72
+ FT-SE (ours)	924.28	369.15	30.37

popular modern latent diffusion backbone. Compared to the original paper, we incorporated several recent advancements into the DiT architecture to improve the baseline performance, as described in Appendix B. Qualitative samples from DiT-XL/2 are provided in Figure 7. The results are shown in Table 1. One can observe that our proposed regularization greatly improves the *diffusability* of the downstream LDM model, allowing to achieve 19% lower FID compared to the vanilla Flux AE and 8% lower FID compared to the Flux AE, fine-tuned in our training pipeline without the SE regularization.

The improvement for CMS-AE$_I$ is reduced with the main reason being that our training pipeline hurts its performance (we explored over 10 different hyperparameters setups to tune the vanilla model): after fine-tuning it for 10,000 steps with only reconstruction losses (it does not use KL regularization by default), the downstream FID performance increases by 14% from 11.69 to 13.59.

Improving video autoencoders. For video autoencoders, we used CogVideoX-AE (Hong et al., 2022) (CV-AE) with $4 \times 8 \times 8$ compression and 16 latent channels and LTX-AE (HaCohen et al., 2024) with $8 \times 32 \times 32$ compression and 32 latent channels. The latter serves as a strong high-compression autoencoder baseline. All the video AEs are fine-tuned for 20,000 training steps on the joint image and video dataset with the batch size of 32. Image batches are treated as single-frame videos which is possible due to the causal structure of the video autoencoders (Yu et al., 2023).

Similar to the image autoencoder experiments, we train a DiT model on Kinetics-700 (Carreira et al., 2019) on three variants: 1) a "vanilla" autoencoder snapshot; 2) the vanilla autoencoder fine-tuned for 20,000 steps in our training pipeline (denoted as "FT"); and 3) the autoencoder snapshot, fine-tuned with our downsampling regularization (denoted as "FT-SE"). For LTX-AE, we used a reduced patchification resolution of 1×1 to compensate for its extreme compression ratio. The results are presented in Table 2: DiT model on top SE-regularized autoencoders has drastically better performance: 44% and 54% lower FVD$_{10K}$ for CV-AE and LTX-AE, respectively. Our training pipeline allowed to achieve better DiT-B training for CV-AE (650.4 vs 447.3 FVD$_{10K}$), but led to worse scores for LTX-AE (854.4 vs 876.6), which we found less stable to train. Adding our regularization strategy greatly improves the LDM performance in each case. One can also observe that the boost for video autoencoders is larger than in the image domain (at least -44% reduced FVD$_{10K}$ for DiT-B for video generation vs -7% reduced FID for DiT-XL for image generation). We attribute this to two factors. First, improvements in Frechet Distances (Heusel et al., 2017) do not scale linearly (i.e.,
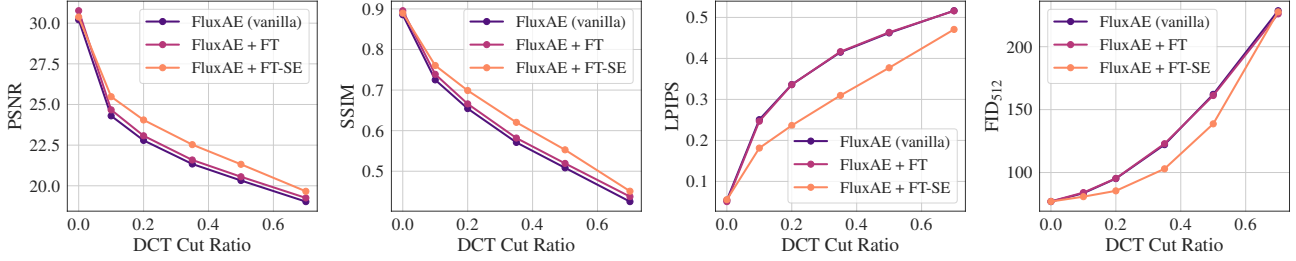
Figure 8. **Effect of DCT spectrum cutting.** We plot reconstruction metrics on ImageNet for the baseline FluxAE and its finetuned version, with and without scale equivariance. As more high-frequency DCT coefficients are removed from the latents, the AE with regularization consistently achieves the best reconstruction quality.
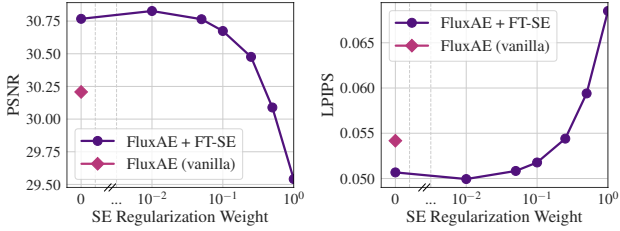

Figure 9. Reconstruction quality with varying SE regularization strength for fine-tuning. We find the value of 0.25 to be optimal in maintaining reconstruction quality compared to the Base while also improving generation quality as shown in Tables 1 and 2.

Table 4. Influence of spectrum regularization on the reconstruction quality for image and video autoencoders. Image autoencoders are evaluated on 50,000 images from ImageNet-1K, while the video ones are on 50,000 videos from Kinetics-700.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	FVD$_{50K}$ ↓
FluxAE	30.243	0.883	0.054	0.183	–
+FT-SE (ours)	30.474	0.888	0.055	0.550	–
CMS-AE$_I$	23.230	0.638	0.181	1.077	–
+FT-SE (ours)	24.558	0.677	0.166	1.570	–
CogVideoX-AE	34.961	0.947	0.073	2.992	4.614
+FT-SE (ours)	35.399	0.948	0.067	2.986	3.328
LTX-AE	30.897	0.886	0.152	5.928	36.783
+FT-SE (ours)	30.386	0.885	0.137	5.303	34.148

their smaller values are progressively harder to improve). Next, causal video autoencoders have less regular latent structure: the first frame in a video is encoded into the same representation size as the subsequent chunks, which leaves more room to enhance the diffusability of the latent space.

We additionally trained a DiT-XL/2 model for the CV-AE family to explore the scalability of our regularization. For this large-scale setup, our SE regularization improved the FVD$_{10K}$ score by almost twice.

4.2. Ablations

Does scale equivariance hurt reconstruction? scale equivariance in VAEs improves downstream generation quality in terms of FID (Tables 1 and 2). We now examine its impact on AE reconstruction quality. Table 4 presents results across

four reconstruction metrics — PSNR, SSIM, LPIPS (Zhang et al., 2018), and FID, on 50,000 samples from ImageNet and Kinetics for image and video autoencoders, respectively. Reconstruction quality remains similar across the models.

Can LDM performance be improved by tweaking the KL weight instead? In Table 3, we show that increasing the KL strength can indeed improve the LDM performance for DiT-S/2, but it inevitably hurts reconstruction, as shown by the PSNRs, which bottlenecks the scalability of larger LDM models. In contrast, our proposed scale equivariance allows to achieve good LDM performance without hurting the reconstruction quality of the autoencoder. For these ablations, we trained the DiT-S/2 variants for 200K training steps and DiT-L/2 variants for 400K steps, and ran inference for 80 steps without classifier-free guidance. One can see that for a small compute budget, increasing the KL strength is beneficial: the best LDM score is obtained with the highest KL $\beta = 0.1$. But it severely affected the reconstruction quality of the autoencoder, which limited its scaling: the corresponding DiT-L/2 LDM variant is ranked among the worst. At the same time, our developed regularization performs well for all DiT variants and does limit scaling.

Can the quality improvement come from implicit time shifting? SE smoothes the latent space, which can result in "implicit time shifting" (Gao et al., 2024): it eliminates high-frequency components from the latents, allowing the model to spend more compute on low-frequency ones, especially at inference time (see B). We trained a family of DiT-B/2 models for various location and scale coefficients for logit-normal noise distribution (Esser et al., 2024). Figure 10 presents the FDD$_{5K}$ results with 128 inference steps (and without CFG) for time shifting coefficients of $[0.1, 0.25, 0.5, 0.75, 1, 2, 3, 4, 5, 6, 8, 10]$. Our regularized FluxAE-FT+SE achieves the best results across *all* the setups, confirming that our improved quality is not due to improved diffusability, rather than implicit time shifting. Note that our original noise schedule was tuned for the vanilla setup, which is why the default location of 0 and scale of 1 perform the best for non-regularized AEs.

Effect of SE regularization strength. In Figure 9, we
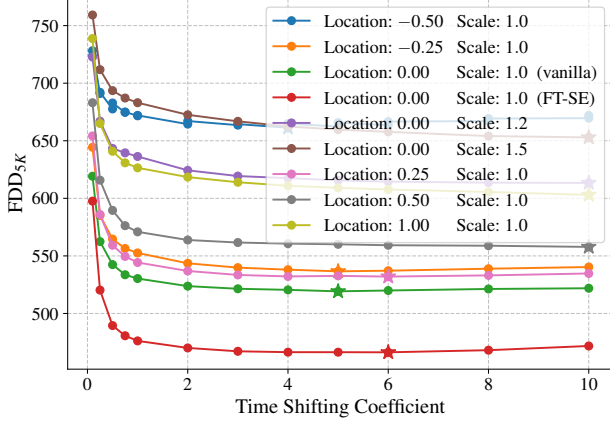
Figure 10. Inference-time time shifting and train-time noise schedule ablation for DiT-B/2 models trained on top of vanilla FluxAE, and also on top of our regularized autoencoder. Our regularized FluxAE-FT+SE achieves the best results across *all* the setups, confirming that our improved quality is not due to improved diffusability, rather than implicit time shifting. The optimal time shifting coefficient is marked with a star.

show the effect of varying the loss weight on the FluxAE performance on ImageNet. Increasing the SE regularization strength naturally worsens the total reconstruction quality since the decoder is trained to generalize its performance across both low frequency and high frequency latents (via downsampling) while having the same capacity. We choose the value of 0.25 to maintain reconstruction performance compared to the base AE model while improving the generation quality as shown in Tables 1 and 2.

Effect of DCT spectrum cutting. To examine the impact of downsampling regularization, we evaluate reconstruction quality by progressively removing high-frequency DCT components from the latents. Figure 8 presents reconstruction metrics on 512 samples from the ImageNet validation set for the baseline FluxAE and fine-tuning, with and without regularization. As the cut ratio increases on the x-axis, indicating the removal of more high-frequency components, the AE with regularized latents consistently achieves the best reconstruction quality across all metrics. This underscores the regularization role in aligning the spectral properties of the latent and RGB spaces.

5. Conclusion

We have shown that modern Latent Diffusion Models (LDMs) rely just as critically on their autoencoders as on the more frequently investigated diffusion architectures. While prior work has largely focused on improving reconstruction quality and compression rates for the autoencoders, our study focuses on *diffusability*, revealing how latent spectra with excessive high-frequency components can hamper the downstream diffusion process. Through a systematic analysis of several autoencoders, we uncovered stark dis-

crepancies between latent and RGB spectral properties and demonstrated that they lead to worse LDM synthesis quality. Building on this insight, we developed a regularization strategy that aligns the latent and RGB spaces across different frequencies. Our approach maintains reconstruction fidelity and improves diffusion training by suppressing spurious high-frequency details in the latent code. Potential future directions include exploring more advanced frequency-based regularizations, adaptive compression methods and scale equivariance regularization in the temporal axis for video autoencoders to further optimize the trade-off between reconstruction quality, compression rate, and diffusability.

Impact Statement

This work focuses on improving the representations in autoencoders that serve as a backbone for latent diffusion training, ultimately enhancing generative performance. Our improvements can facilitate beneficial applications such as boosting creativity, supporting educational content creation, and reducing computational overhead in generative workflows. Beyond these considerations, we do not identify additional ethical or societal implications beyond those already known to accompany large-scale generative modeling.

References

Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Ahmed, N., Natarajan, T., and Rao, K. R. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 2006.

Albergo, M. S. and Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Ballé, J., Laparra, V., and Simoncelli, E. P. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

Betker, J., Goh, G., Jing, L., Brooks†, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo†, Y., Manassra, W., Dhariwal, P., Chu, C., Jiao†, Y., and Ramesh, A. Improving image generation with better captions. https://cdn.openai.com/papers/dall-e-3.pdf, 2023. Accessed: 2023-11-14.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2023.

Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., and Ramesh, A. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Byeon, M., Park, B., Kim, H., Lee, S., Baek, W., and Kim, S. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022.

Carreira, J., Noland, E., Hillier, C., and Zisserman, A. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

Chen, J., Cai, H., Chen, J., Xie, E., Yang, S., Tang, H., Li, M., Lu, Y., and Han, S. Deep compression autoencoder for efficient high-resolution diffusion models. *arXiv preprint arXiv:2410.10733*, 2024a.

Chen, J., Cai, H., Chen, J., Xie, E., Yang, S., Tang, H., Li, M., Lu, Y., and Han, S. Deep compression autoencoder for efficient high-resolution diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=wH8XXUOUZU.

Chen, T. and Li, L. Fit: Far-reaching interleaved transformers. *arXiv preprint arXiv:2305.12689*, 2023.

Chen, T.-S., Siarohin, A., Menapace, W., Deyneka, E., Chao, H.-w., Jeon, B. E., Fang, Y., Lee, H.-Y., Ren, J., Yang, M.-H., and Tulyakov, S. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Cheng, Z., Sun, H., Takeuchi, M., and Katto, J. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7939–7948, 2020.

Dao, Q., Phung, H., Nguyen, B., and Tran, A. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 2009.

Dieleman, S. Diffusion is spectral autoregression, 2024. URL https://sander.ai/2024/09/02/spectral-autoregression.html.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Esteves, C., Suhail, M., and Makadia, A. Spectral image tokenizer. *arXiv preprint arXiv:2412.09607*, 2024.

Gao, P., Zhuo, L., Liu, D., Du, R., Luo, X., Qiu, L., Zhang, Y., Lin, C., Huang, R., Geng, S., et al. Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers. *arXiv preprint arXiv:2405.05945*, 2024.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Guo, Y., Yang, C., Rao, A., Wang, Y., Qiao, Y., Lin, D., and Dai, B. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Li, F.-F., Essa, I., Jiang, L., and Lezama, J. Photorealistic video generation with diffusion models. In *European Conference on Computer Vision*, pp. 393–411. Springer, 2024.

HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., Richardson, E., Levin, E., Shiran, G., Zabari, N., Gordon, O., et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Hansen-Estruch, P., Yan, D., Chung, C.-Y., Zohar, O., Wang, J., Hou, T., Xu, T., Vishwanath, S., Vajda, P., and Chen, X. Learnings from scaling visual tokenizers for reconstruction and generation. *arXiv preprint arXiv:2501.09755*, 2025.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Hong, W., Ding, M., Zheng, W., Liu, X., and Tang, J. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

Jabri, A., Fleet, D. J., and Chen, T. Scalable adaptive computation for iterative generation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577, 2022.

Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024.

Kingma, D. P. and Gao, R. Understanding the diffusion objective as a weighted integral of elbos. *arXiv preprint arXiv:2303.00848*, 2023.

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

Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Kouzelis, T., Ioannis, K., Spyros, G., and Nikos, K. Eqvae: Equivariance regularized latent space for improved generative image modeling. In *arxiv*, 2025.

Kynkäänniemi, T., Aittala, M., Karras, T., Laine, S., Aila, T., and Lehtinen, J. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724*, 2024.

Li, J., Li, B., and Lu, Y. Deep contextual video compression. *Advances in Neural Information Processing Systems*, 34: 18114–18125, 2021.

Li, J., Li, B., and Lu, Y. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22616–22626, 2023.

Li, Y. and van der Schaar, M. On error propagation of diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RtAct1E2zS.

Lin, B., Ge, Y., Cheng, X., Li, Z., Zhu, B., Wang, S., He, X., Ye, Y., Yuan, S., Chen, L., et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Liu, X., Zhang, X., Ma, J., Peng, J., and Liu, Q. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation, 2024. URL https://arxiv.org/abs/2309.06380.

Loshchilov, I. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Menapace, W., Siarohin, A., Skorokhodov, I., Deyneka, E., Chen, T.-S., Kag, A., Fang, Y., Stoliar, A., Ricci, E., Ren, J., et al. Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. *arXiv preprint arXiv:2402.14797*, 2024.

Mentzer, F., Toderici, G., Minnen, D., Hwang, S.-J., Caelles, S., Lucic, M., and Agustsson, E. Vct: A video compression transformer. *arXiv preprint arXiv:2206.07307*, 2022.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for gans do actually converge? In *International conference on machine learning*, 2018.

Minnen, D., Ballé, J., and Toderici, G. D. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171, 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

Ning, M., Li, M., Su, J., Jia, H., Liu, L., Beneš, M., Salah, A. A., and Ertugrul, I. O. Dctdiff: Intriguing properties of image generative modeling in the dct space. *arXiv preprint arXiv:2412.15032*, 2024.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.-Y., Xu, H., Sharma, V., Li, S.-W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. Dinov2: Learning robust visual features without supervision, 2023.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Polyak, A., Zohar, A., Brown, A., Tjandra, A., Sinha, A., Lee, A., Vyas, A., Shi, B., Ma, C.-Y., Chuang, C.-Y., et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Rissanen, S., Heinonen, M., and Solin, A. Generative modelling with inverse heat dissipation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4PJUBT9f2Ol.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Ruderman, D. L. Origins of scaling in natural images. *Vision Research*, 37(23):3385–3398, 1997. ISSN 0042-6989. doi: https://doi.org/10.1016/S0042-6989(97)00008-4. URL https://www.sciencedirect.com/science/article/pii/S0042698997000084.

Sadat, S., Buhmann, J., Bradley, D., Hilliges, O., and Weber, R. M. Litevae: Lightweight and efficient variational autoencoders for latent diffusion models. *arXiv preprint arXiv:2405.14477*, 2024.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494, 2022.

Sheng, X., Li, J., Li, B., Li, L., Liu, D., and Lu, Y. Temporal context mining for learned video compression. *IEEE Transactions on Multimedia*, 25:7311–7322, 2022.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

Stein, G., Cresswell, J., Hosseinzadeh, R., Sui, Y., Ross, B., Villecroze, V., Liu, Z., Caterini, A. L., Taylor, E., and Loaiza-Ganem, G. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Tang, A., He, T., Guo, J., Cheng, X., Song, L., and Bian, J. Vidtok: A versatile and open-source video tokenizer. *arXiv preprint arXiv:2412.13061*, 2024.

Tian, R., Dai, Q., Bao, J., Qiu, K., Yang, Y., Luo, C., Wu, Z., and Jiang, Y.-G. Reducio! generating 1024x1024 video within 16 seconds using extremely compressed motion latents. *arXiv preprint arXiv:2411.13552*, 2024.

Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Vahdat, A., Kreis, K., and Kautz, J. Score-based generative modeling in latent space. 2021. *arXiv preprint arXiv:2106.05931*, 2021.

von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Nair, D., Paul, S., Berman, W., Xu, Y., Liu, S., and Wolf, T. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

Wallace, G. K. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.

Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Wang, J., Jiang, Y., Yuan, Z., Peng, B., Wu, Z., and Jiang, Y.-G. Omnitokenizer: A joint image-video tokenizer for visual generation. *arXiv preprint arXiv:2406.09399*, 2024.

Wu, P., Zhu, K., Liu, Y., Zhao, L., Zhai, W., Cao, Y., and Zha, Z.-J. Improved video vae for latent video diffusion model. *arXiv preprint arXiv:2411.06449*, 2024.

Xie, E., Chen, J., Chen, J., Cai, H., Tang, H., Lin, Y., Zhang, Z., Li, M., Zhu, L., Lu, Y., et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

Xing, Y., Fei, Y., He, Y., Chen, J., Xie, J., Chi, X., and Chen, Q. Large motion video autoencoding with cross-modal video vae. *arXiv preprint arXiv:2412.17805*, 2024.

Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

Yu, L., Lezama, J., Gundavarapu, N. B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Gupta, A., Gu, X., Hauptmann, A. G., et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Zhao, S., Zhang, Y., Cun, X., Yang, S., Niu, M., Li, X., Hu, W., and Shan, Y. Cv-vae: A compatible video vae for latent generative video models. *arXiv preprint arXiv:2405.20279*, 2024.

Zhao, Y., Gu, A., Varma, R., Luo, L., Huang, C.-C., Xu, M., Wright, L., Shojanazeri, H., Ott, M., Shleifer, S., et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., and You, Y. Open-sora: Democratizing efficient video production for all, 2024. URL https://github.com/hpcaitech/Open-Sora.

Zhou, Y., Wang, Q., Cai, Y., and Yang, H. Allegro: Open the black box of commercial-level video generation model. *arXiv preprint arXiv:2410.15458*, 2024.

Zhou, Y., Xiao, Z., Yang, S., and Pan, X. Alias-free latent diffusion models: Improving fractional shift equivariance of diffusion latent space. In *CVPR*, 2025.

A. Limitations.

We identify the following limitations of our work and the proposed regularization:

1. While we did our best to verify that our framework works in the most general setup possible, testing 4 different autoencoders across 2 different domains (image and videos), our study would be more complete when verified across other diffusion parametrizations (Karras et al., 2022; Ho et al., 2020; Kingma & Gao, 2023) or architectures (Karras et al., 2024).

2. We observed that our regularization still affects the reconstruction slightly: for example, Table 4 shows that FluxAE FID increased from 0.183 to 0.55 (though for some AEs, like CogVideoX-AE, it improves). We are convinced that this FID increase could be mitigated by training with adversarial losses, which we omitted in this work for simplicity.

3. There is a mild sensitivity to hyperparameters: for example, we found that varying the SHF regularization weight might improve the results (see Table 9), or adding a small KL regularization (which we disabled in the end for our regularization for simplicity).

4. None of the explored autoencoders released their training pipelines, and it is non-trivial to fine-tune them even without any extra regularization. For example, we observed that any fine-tuning of DC-AE (Chen et al., 2025) was leading to divergent reconstructions in our training pipeline (we explored dozens of different hyperparameter setups).

We leave the exploration of these limitations for future work.

B. Implementation Details

DiT model details. To strengthen the baseline DiT performance, we integrated into it the latest advancements from the diffusion model literature. Namely, we used self conditioning (Jabri et al., 2023) and RoPE (Su et al., 2024) positional embeddings. Besides, we switched to the rectified flow diffusion parametrization (Albergo & Vanden-Eijnden, 2022; Liu et al., 2022; Vahdat et al., 2021), which was recently shown to have better scalability with a fewer amount of inference steps (Esser et al., 2024).

Following (Jabri et al., 2023; Chen & Li, 2023; Gupta et al., 2024; Menapace et al., 2024), we employ *self-conditioning* for our DiT training and inference. During training with a 90% probability, we run an auxiliary forward pass with the DiT model, take its activations from the last block (i.e., right before the "unpatchify" projection), project them with a linear layer and add as residuals to the input tokens after patchification of the main training forward pass. For that auxiliary forward pass, following RIN (Jabri et al., 2023), we use the same noise level σ and "no-grad" context (i.e., we do not backpropagate through the auxiliary forward pass).

DiT training details. All the DiT models are trained for 400,000 steps with 10,000 warmup steps of the learning rate from 0 to 0.0003 and then its gradual decay towards 0.00001. We used weight decay of 0.01 and AdamW (Loshchilov, 2017) optimizer with beta coefficients of 0.9 and 0.99. We used posterior sampling from the encoder distribution for VAE-based autoencoders. In contrast to the original work, we found it helpful to do learning rate decay to 0.00001 using the cosine learning rate schedule. We used the same model sizes for DiT-S (small), DiT-B (base), DiT-L (large) and DiT-XL (extra large), as the original work (Peebles & Xie, 2022):

- DiT-S: hidden dimensionality of 384, 12 transformer blocks, and 6 attention heads in the multi-head attention.

- DiT-B: hidden dimensionality of 768, 12 transformer blocks, and 12 attention heads in the multi-head attention.

- DiT-L: hidden dimensionality of 1024, 24 transformer blocks, and 16 attention heads in the multi-head attention.

- DiT-XL: hidden dimensionality of 1152, 28 transformer blocks, and 16 attention heads in the multi-head attention.

We used gradient clipping with the norm of 16 for all the DiT models. Our models were trained in the FSDP (Zhao et al., 2023) framework with the full sharding strategy on a single node of $8\times$ NVidia A100 80GB GPUs or $8\times$ NVidia H100 80GB GPUs (depending on their availability in our computational cluster).

For CV-AE, since it is considerably slower than other autoencoders, we trained LDMs on pre-extracted latents. For this, we pre-extracted them on random 17-frames clips. In essence, this reduces the total dataset size, but since we do the same procedure for the entire CogVideoX-AE family, the models are comparable between each other.

Autoencoders training details. Since none of the autoencoders had their training pipelines released, we had to develop the training recipes for each of the autoencoder baselines individually which would not be detrimental to neither their reconstruction capability nor downstream diffusion performance. To do this, we ablated multiple hyperparameters (the most important ones being learning rate and KL regularization strength) to arrive to a proper setup. We chose the KL weight in such a way that the KL penalty maintains approximately the same magnitude as the pre-trained checkpoint.

Each autoencoder is trained with AdamW (Loshchilov, 2017) optimizer, with betas of 0.9 and 0.99, and weight decay of 0.01. The learning rate was grid-searched individually for each autoencoder and is provided in Table 5. In all the cases, we used mixed precision training with BFloat16.

During training, we maintained an exponential moving average of the weights (Karras et al., 2024), initialized from the same parameters as the starting model, and having a half life of 5,000 steps.

We emphasize that, when applying our regularization strategy on top of an autoencoder baseilne, we do not alter other hyperparameters (like learning rate), except for KL regularization which we disable for SE-regularized models (even though we found it helpful in some of our explorations).

For each autoencoder, we freeze the last output layers of the decoder. The motivation is the following: they were fine-tuned with the adversarial loss, which we want to exclude from the equation without hurting the ability of an autoencoder to model textural details which FID would be sensitive to (Rombach et al., 2022) and which do not influence the latent space properties. Namely, we freeze the last normalization and output convolution layers. In each case, the amount of frozen parameters constitute a negligible amount of total parameters.

Other hyperparameters for autoencoders training are provided in Table 5.

Table 5. Hyperparameters for the autoencoders explored in the current work. We had to tweak the hyperparameters for various autoencoders to prevent the divergence of the baseline training.

Hyperparameter	FluxAE	CMS-AE$_I$	CV-AE	LTX-AE
Domain	image	image	video	video
Compression rate	8×8	16×16	$4 \times 8 \times 8$	$8 \times 32 \times 32$
Latent channels	16	16	16	32
Number of fune-tuning steps	10,000	10,000	20,000	20,000
Image batch size	32	32	64	64
Video batch size	0	0	32	32
Default KL β weight	0.001	0.0	0.001	0.0001
Learning rate	0.00001	0.0001	0.0003	0.00005
Number of parameters	83.8M	44M	211.5M	419M
Training resolution	256×256	256×256	$17 \times 256 \times 256$	$17 \times 256 \times 256$
MSE loss weight	1	1	1	5
LPIPS loss weight	1.0	1.0	1.0	1.0
Gradient clipping norm	50	50	1	50
Num upsampling blocks frozen	1	3	0	0
Is output convolution frozen?	Yes	Yes	Yes	Yes

Time shifting for rectified flow. In Section 4.2, we discussed the influence of time shifting on the DiT model. Here, we provided the details of its implementation. We adapt the time shifting schedule from Lumina-T2X (Gao et al., 2024) for rectified flow parametrization by rescaling the time steps at inference time using the following formula:

$$t' = 1 - \frac{1 - t}{s + (1 - s) \cdot (1 - t)}, \tag{3}$$

where t is the time step, going from 1 (full noise) to 0 (clean image), and s is the shifting coefficient. One can see that for $s = 1$, $t' = t$, which is the default unshifted behaviour.

C. Additional Experiments

In Section 3, we outlined the base scale equivariance strategy to regularize the spectrum of an autoencoder which has a strong advantage of being very easy to implement by a practitioner. However, it could be beneficial to possess more advanced tools for a finer-grained control over the latent space spectral properties. This section outlines them and provides the corresponding ablation.

C.1. Explicitly Chopping off High-Frequency Components

Rather than applying downsampling to produce latents and RGB targets for regualrization, it is possible to replace some ratio of high-frequency components with zeros. To do so, DCT is applied to the latents and RGB targets where a chosen set of frequency components are masked out. The modified components are then translated back to the spatial domain by inverse DCT to form the training latents and reconstruction targets.

$$\mathcal{L}_{\text{CHF}}(x) = d(x, \text{Dec}(z)) + d(D^{-1}(D(x) * \mathbf{M}), \text{Dec}(D^{-1}(D(z) * \mathbf{M})) + \mathcal{L}_{\text{reg}}, \tag{4}$$

where D and D^{-1} represent DCT and its inverse, respectively. $\mathbf{M}$ is a $B \times B$ binary mask indicating which frequencies to zero out defined as follows:

$$\mathbf{M}(u, v) = \begin{cases} 1, & \text{if } \text{zigzag}(u, v) < B^2 - N, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

N controls the frequency cutoff. We provide the ablation for this strategy in Table 6.

Table 6. Ablations for explicit high-frequency chop off for DiT-S/2 trained for 200,000 iterations on top of Flux AE with such a regularization. While it can achieve better results for some of the baselines than naive downsampling, we opt out for the latter strategy due to its simplicity. For the non-zigzag order ablation, we cut across each x and y axes independently

Stage II	Stage I	FDD$_{5K}$
DiT-S/2	FluxAE + chop off 90% (non-zigzag order)	912.4
DiT-S/2	FluxAE + chop off 70% (non-zigzag order)	915.6
DiT-S/2	FluxAE + chop off 30% (non-zigzag order)	929.7
DiT-S/2	FluxAE + chop off 10% (non-zigzag order)	916.5
DiT-S/2	FluxAE + chop off 90% (zigzag order)	935.5
DiT-S/2	FluxAE + chop off 70% (zigzag order)	932.8
DiT-S/2	FluxAE + chop off 30% (zigzag order)	962.9
DiT-S/2	FluxAE + chop off 10% (zigzag order)	930.1
DiT-S/2	FluxAE (vanilla)	992.0
DiT-S/2	FluxAE with optimal (out of 8) KL β	929.6

In Figure 5, we provided the visualizations for a FluxAE resiliense with and without such chopping high-frequency regularization for 50% HF dropout rate. In Figure 11, we provide an equivalent visualization for SE-fine-tuned FluxAE: while it is less resilient to frequency dropout than CHF, but is still noticeably better than the vanilla model.

C.2. Soft Penalty for High-Frequency Components

Instead of directly removing some of the components, which might become a too strict regularization signal, one can consider penalizing the amplitudes of high-frequency components in a soft manner. Concretely, given a $B \times B$ block, we construct the following weight penalty matrix:

$$\mathcal{W}_{uv} = (u + v)^p / B^p. \tag{6}$$

Next, the soft regularization loss itself is computed as:

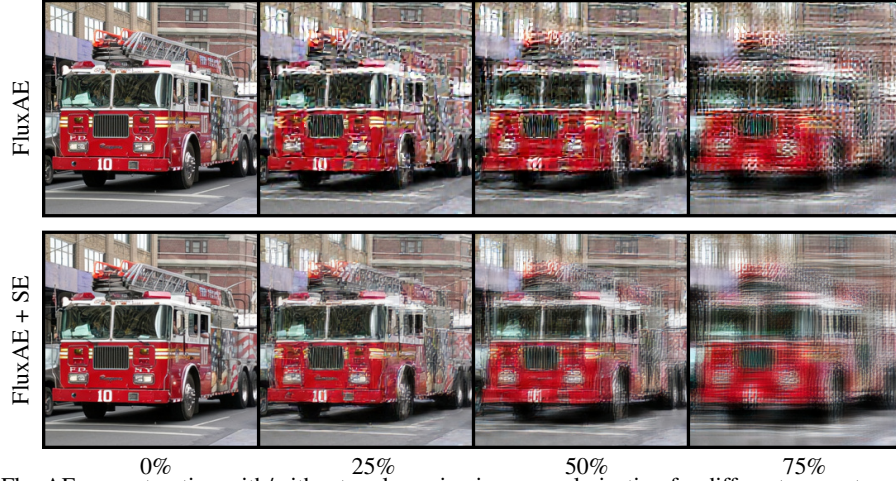$$\mathcal{L}_{\text{softreg}} = \sum_{u,v} D_{uv}(z) \cdot \mathcal{W}_{uv}. \tag{7}$$

Figure 11. RGB and FluxAE reconstruction with/without scale equivariance regularization for different percentages of chopped-off high frequency components.

During training, when enabled, we add it to the main loss with the weigh γ. We found it beneficial in some of our experiments when it is added with a small coefficient (e.g., 0.01). While it is possible to achieve higher results with more fine-grained regularization, we opt to use the simpler version since we believe it would be easier to employ by the community.

To ablate its importance, we trained DiT-B/2 model on top of FluxAE models, fine-tuned with a different strength γ. The results are presented in Table 7.

Table 7. Ablating the regularization strength γ of the soft frequency regularization $\mathcal{L}_{\text{softreg}}$.

Stage II	Stage I	FID $_{5k}$	FDD $_{5k}$
	FluxAE + FT-SE $\gamma = 0.001$	26.43	497.14
	FluxAE + FT-SE $\gamma = 0.025$	25.46	477.61
DiT-B/2	FluxAE + FT-SE $\gamma = 0.01$	26.72	487.06
	FluxAE + FT-SE $\gamma = 0.05$	24.28	458.11
	FluxAE + FT-SE $\gamma = 0.1$	25.84	461.97

C.3. ImageNet 512^2 experiments

We trained our DiT-L/2 for class-conditional 512^2 ImageNet-1K generation for 400K steps for FluxAE (Black Forest Labs, 2023), the results are presented in Table 8.

C.4. Ablating regularization strength α

To ablate the importance of the regularization strength α, we train FluxAE for 10,000 steps with a varying strength. The results are presented in Table 9.
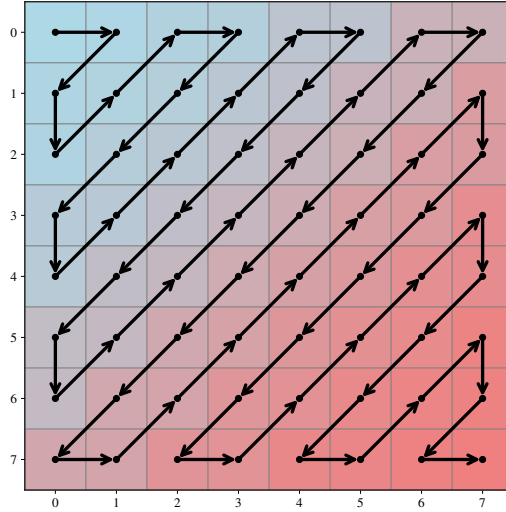
Figure 12. Illustration of the zigzag indexing order of DCT.

Table 8. Class-conditional generation results on ImageNet-1K 512^2 without guidance. The original DiT paper reports the results after 3M training steps, while we use 400K steps for our models.

Stage II	Stage I	FID	FDD
	FluxAE (vanilla)	13.13	249.4
DiT-L/2	FluxAE + FT	13.69	267.7
	FluxAE + FT-SE (ours)	11.63	203.5
DiT-XL/2 (orig) + 3M steps	SD-VAE-ft-MSE	12.03	–

Table 9. Ablating the regularization strength α of our proposed scale equivariance regularization.

Stage II	Stage I	FID $_{5k}$	FDD $_{5k}$
	FluxAE + FT-SE $\alpha = 0.01$	33.99	641.95
	FluxAE + FT-SE $\alpha = 0.05$	33.86	645.94
DiT-B/2	FluxAE + FT-SE $\alpha = 0.1$	28.62	586.91
	FluxAE + FT-SE $\alpha = 0.25$	26.84	558.36
	FluxAE + FT-SE $\alpha = 0.5$	29.63	569.92
	FluxAE + FT-SE $\alpha = 1$	33.22	612.45

C.5. From scratch training

To conduct whether our regularization provides benefits for from scratch training and when the same data is being used for both AE and LDM, we conducted additional experiments for FluxAE and CogVideoX-AE. We trained FluxAE from scratch on ImageNet 256^2 with/without our regularization and then trained DiT-B/2 on top of the latent spaces of these two autoencoders. Then, we did the same for CogVideoX-AE for Kinetics-700 17×256^2. The results of AE reconstruction performance (after 300K training steps for FluxAE and 60K training steps for CogVideoX-AE) and LDM generation performance after 400K training steps are presented in Table 10. One can observe that our regularization improves the performance in both cases.

C.6. Compute cost analysis

One might argue that the improved performance of our regularization comes from the extra computation which the autoencoder is using during training. In this section, we show that it is not the case by providing an analysis of its

Table 10. From scratch training results for FluxAE (Black Forest Labs, 2023) and CogVideoX-AE (Yang et al., 2024) See Appendix C.5 for details.

Method	PSNR ↑	LPIPS ↓	gFDD$_{5K}$ ↓	gFID$_{5K}$ ↓	gFVD$_{5K}$ ↓
FluxAE	29.39	0.059	502.44	25.43	–
+FT-SE (ours)	28.73	0.065	472.26	22.32	–
CogVideoX-AE	37.08	0.049	614.95	19.87	336.33
+FT-SE (ours)	36.61	0.053	515.68	17.40	237.28

computational cost.

We measured the FLOPs of FluxAE (for the batch size of 1 and resolution of 2562) using the popular fvcore library. The entire encoder-decoder pass has 447 GFLOPS, and is split between the encoder/decoder as 136 vs 311 GFLOPS. Our regularization reuses the encoder pass and only runs the decoder with $\times 2$ or $\times 4$ reduced resolution (the scale sampled randomly during training). This results in 77.6 or 19.4 extra GFLOPs of the decoder, which is almost exactly 1/4 or 1/16 of the decoder compute or +17% or 4.5% of the total forward pass. Since we sample $\times 2$ or $\times 4$ downsampling factor equally randomly, this results in 10.75% of total FLOPs overhead for our regularization.

To strengthen our point even further, we ran an experiment where the baseline FluxAE was fine-tuned strictly for 2 times longer (for 20K instead of 10K iterations). The resulted DiT-B/2 model achieved FID$_{5K}$ and FDD$_{5K}$ of only 33.99 and 642.7 versus the scores of 25.87 and 551.27 respectively for our regularized FluxAE+FT-SE model, fine-tuned for only 10K steps.

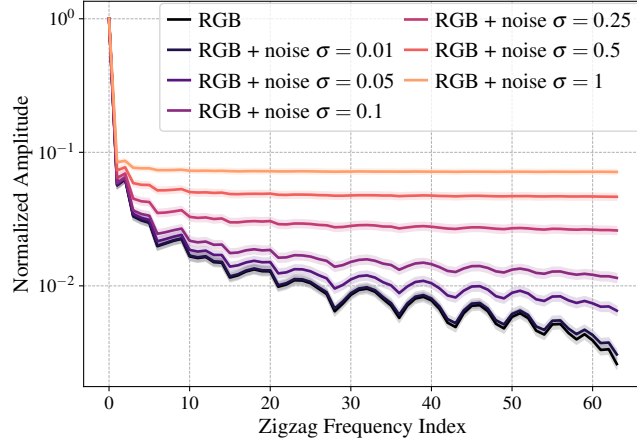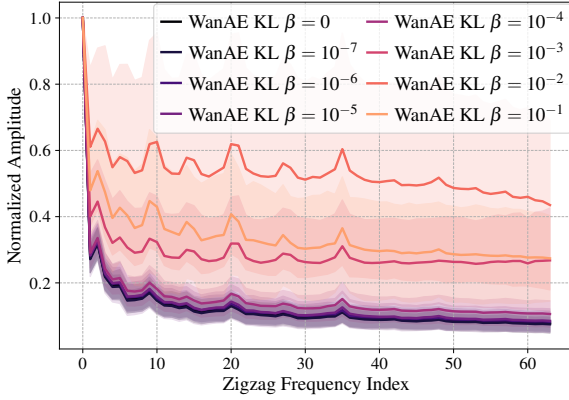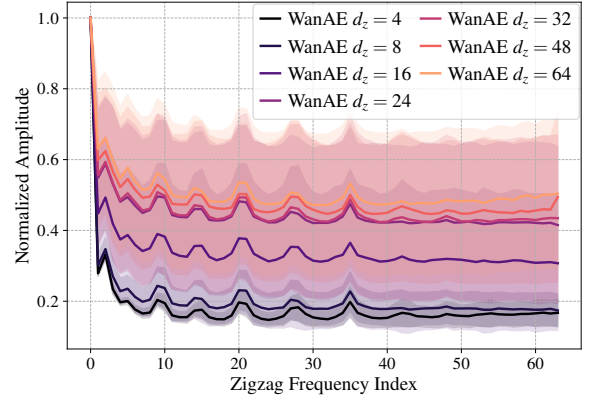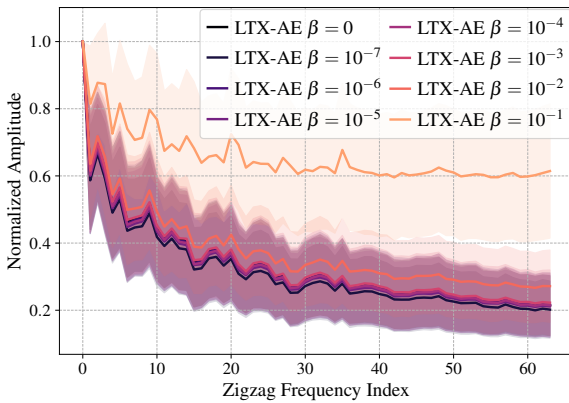# D. Additional visualizations

D.1. Spectra visualizations

Figure 13. Spectrum of RGB under different noising levels: noise inflates the high-frequency component (when normalized). This results into an undesirable side-effect of KL regularization.
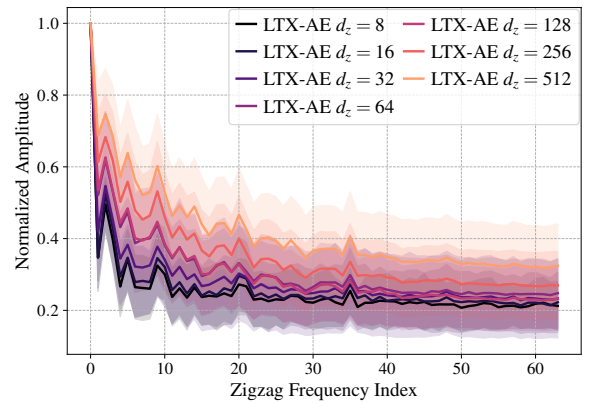

(a) WanAE, KL regularization.

(b) WanAE, channel dimensions.

(c) LTX-AE, KL regularization.

(d) LTX-AE, channel dimensions.

Figure 14. Latent spectra of WanAE (Wan et al., 2025) and LTX-AE (HaCohen et al., 2024) trained from scratch on Kinetics-700 17×256^2 for 100K steps with varying KL regularization strength β or channel size.

D.2. Samples visualizations

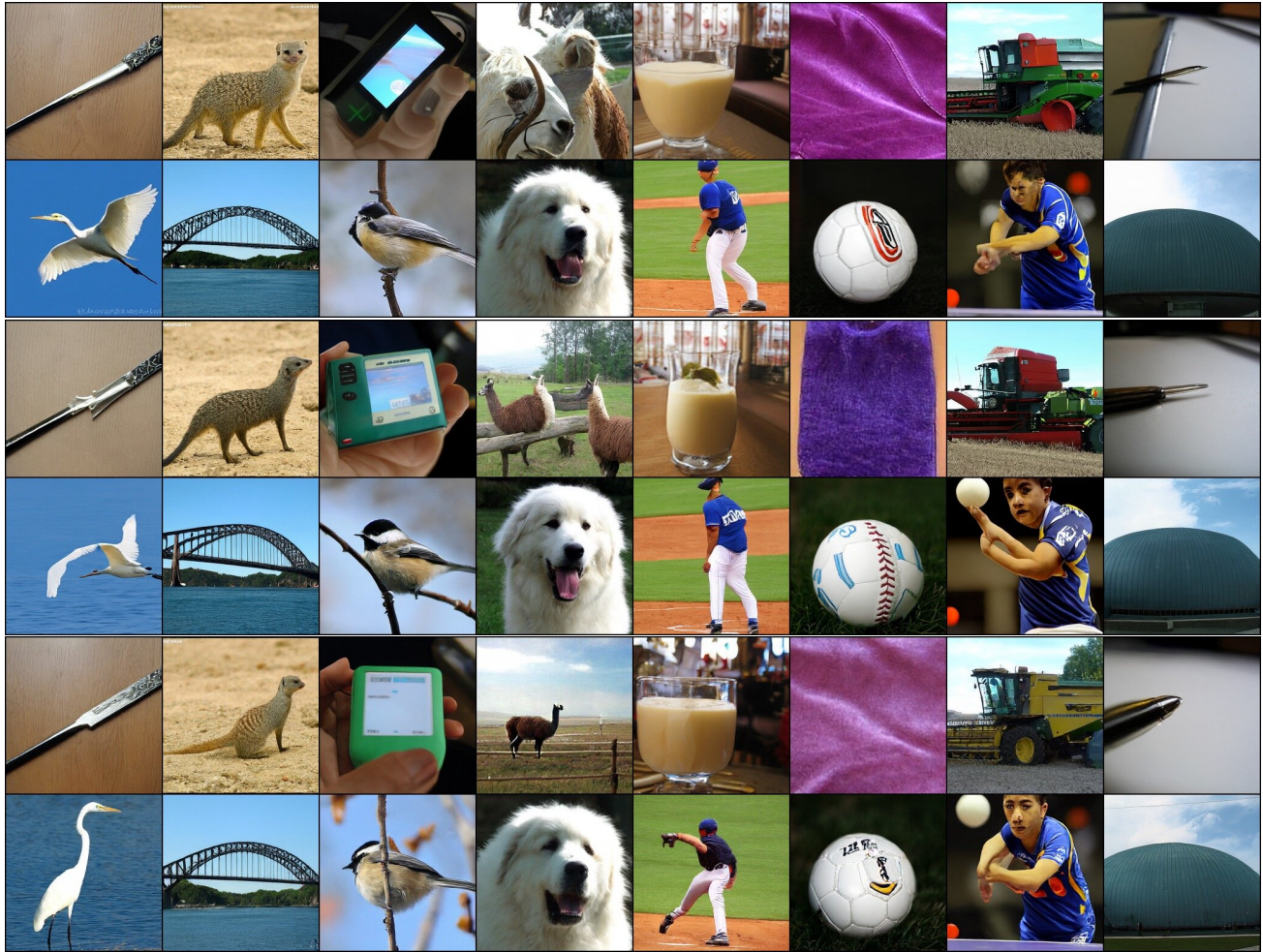*Figure 15.* Uncurated samples from DiT-XL/2 for FluxAE (top), FluxAE + FT (middle) and FluxAE + SE (bottom) on class-conditional ImageNet 256×256 for random classes. During inference, we used 256 steps with the guidance scale of 3.0.
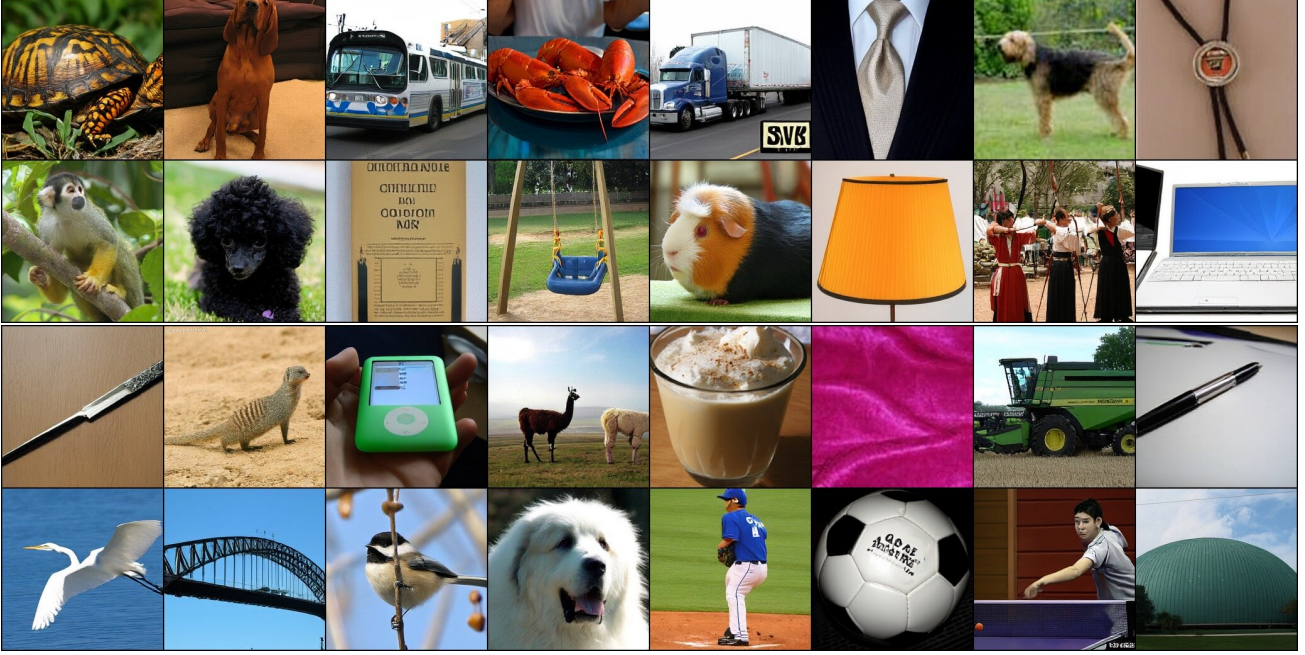
Figure 16. Uncurated samples from DiT-XL/2 trained for 1M steps on top FluxAE + SE (bottom) on class-conditional ImageNet 256×256 for random classes. During inference, we used 256 steps with the guidance scale of 3.0.

Figure 17. Uncurated samples from DiT-XL/2 trained for 1M steps on top FluxAE + SE (bottom) on class-conditional ImageNet 256×256 for class 88. During inference, we used 256 steps with the guidance scale of 3.0.

Figure 18. Uncurated samples from DiT-XL/2 trained for 1M steps on top FluxAE + SE (bottom) on class-conditional ImageNet 256×256 for class 130. During inference, we used 256 steps with the guidance scale of 3.0.

Figure 19. Uncurated samples from DiT-XL/2 trained for 1M steps on top FluxAE + SE (bottom) on class-conditional ImageNet 256×256 for class 279. During inference, we used 256 steps with the guidance scale of 3.0.

Figure 20. Uncurated samples from DiT-XL/2 trained for 1M steps on top FluxAE + SE (bottom) on class-conditional ImageNet 256×256 for class 555. During inference, we used 256 steps with the guidance scale of 3.0.

Figure 21. Uncurated samples from DiT-XL/2 trained for 400K steps on top FluxAE + SE (bottom) on class-conditional ImageNet 512×512 for random classes. During inference, we used 256 steps with the guidance scale of 3.0.
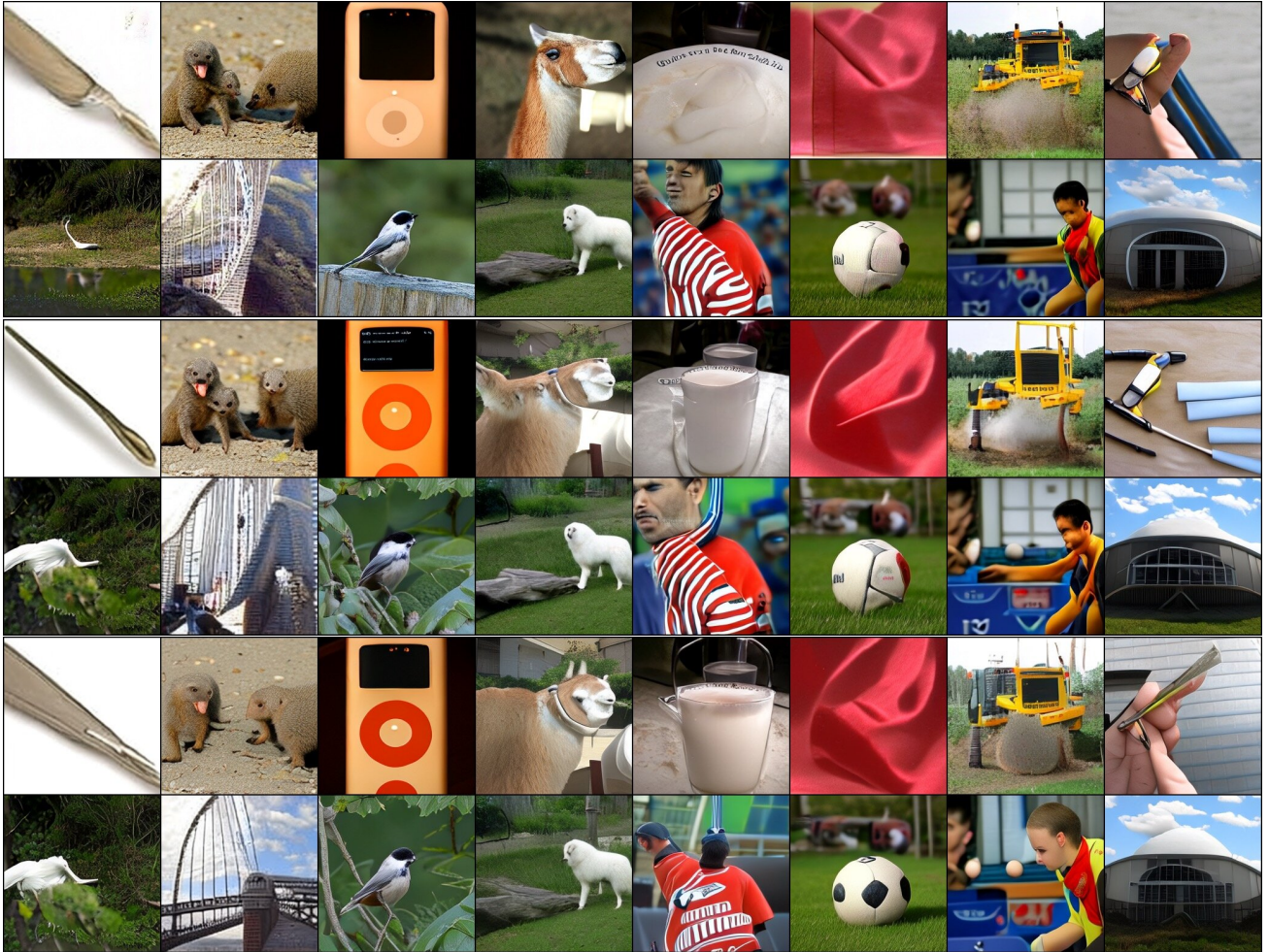
Figure 22. Uncurated samples from DiT-B/1 for CMS-AE$_I$ (top), CMS-AE$_I$ + FT (middle) and CMS-AE$_I$ + SE (bottom) on class-conditional ImageNet 256×256. During inference, we used 256 steps with the guidance scale of 1.5.

Figure 23. Uncurated samples from DiT-XL/2 for CogVideoX-AE (top), CogVideoX-AE + FT (middle) and CogVideoX-AE + SE (bottom) on class-conditional Kinetics $17 \times 256 \times 256$. During inference, we used 256 steps with the guidance scale of 3.0.

Figure 24. Uncurated samples from DiT-B/2 for CogVideoX-AE (top), CogVideoX-AE + FT (middle) and CogVideoX-AE + SE (bottom) on class-conditional Kinetics $17 \times 256 \times 256$. During inference, we used 256 steps with the guidance scale of 3.0.
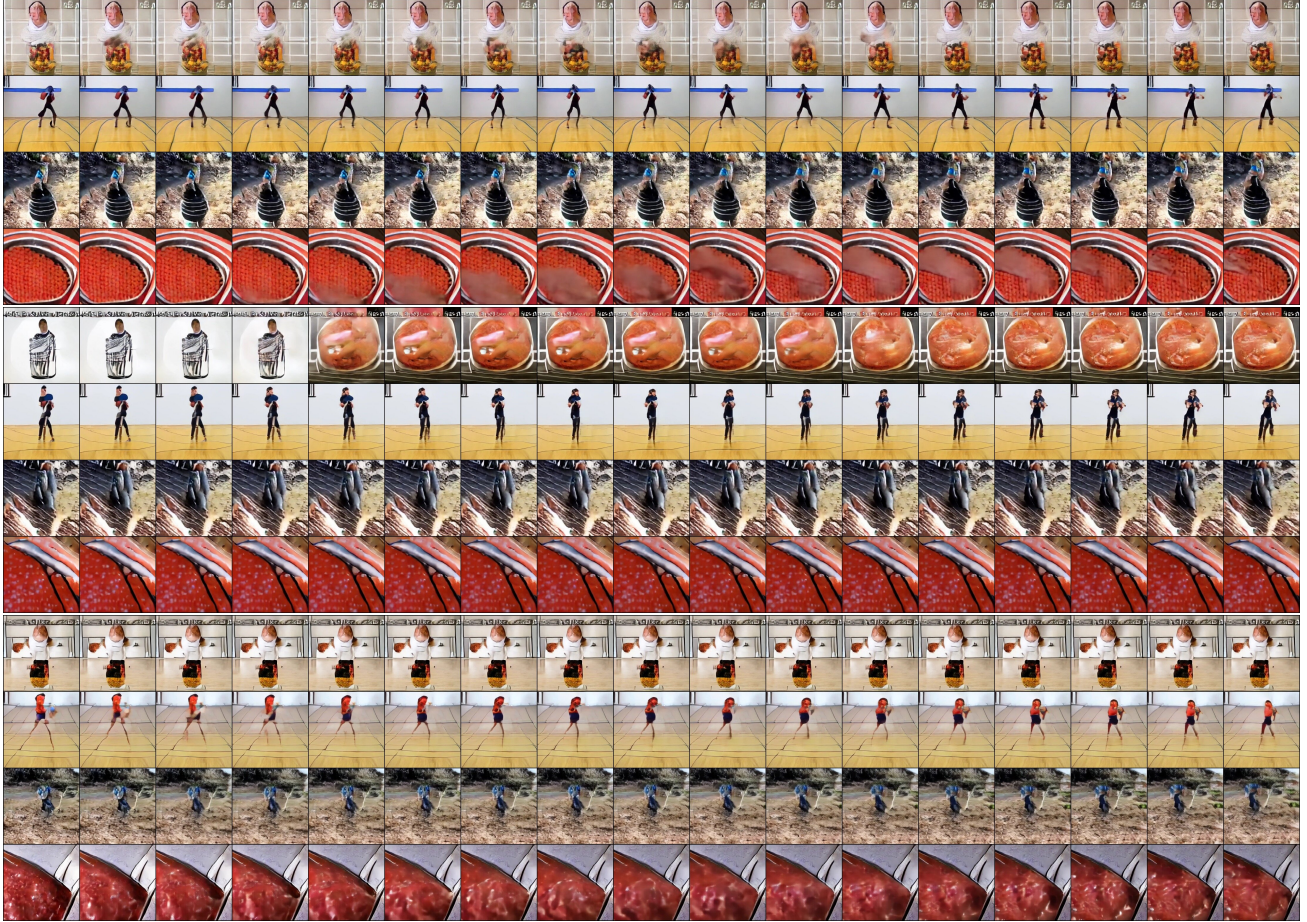
Figure 25. Uncurated samples from DiT-B/1 for LTX-AE (top), LTX-AE + FT (middle) and LTX-AE + SE (bottom) on class-conditional Kinetics $17 \times 256 \times 256$. During inference, we used 256 steps with the guidance scale of 3.0.

E. Failed experiments

Over the course of the project, we explored several other ideas to regularize the latent space of autoencoders. While they did not pan out that well, we still discuss them in this section to spur potential future exploration.

MiniLDM-regulzation training. LSGM (Vahdat et al., 2021) showed that the correct objective of LDM training optimizes both autoencoder and LDM at the same time:

$$
\begin{aligned}
\mathcal{L}(\mathbf{x}, \phi, \theta, \psi) &= \mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x})}\left[-\log p_\psi(\mathbf{x}|\mathbf{z}_0)\right] + \mathrm{KL}(q_\phi(\mathbf{z}_0|\mathbf{x})\|p_\theta(\mathbf{z}_0)) \\
&= \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x})}\left[-\log p_\psi(\mathbf{x}|\mathbf{z}_0)\right]}_{\text{reconstruction term}} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x})}\left[\log q_\phi(\mathbf{z}_0|\mathbf{x})\right]}_{\text{negative encoder entropy}} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x})}\left[-\log p_\theta(\mathbf{z}_0)\right]}_{\text{cross entropy}},
\end{aligned}
\tag{8}
$$

where $\mathcal{L}(\mathbf{x}, \phi, \theta, \psi)$ is the ELBO objective, q_ϕ, p_ψ are encoder/decoder, and p_θ is the LDM (or any other latent generative model). This motivated us to explore training a small LDM model together with the autoencoder expecting that it would make its latent space more diffusable. However, even with extensive hyperparameter search, it was not outperforming the vanilla baseline.

Lipszhitz regularization. Dao et al. (2023) derived the following upper bound on the Wasserstein distance $\mathcal{W}_2^2(p_0, \hat{p}_0)$ between the ground truth $p_0(\mathbf{x})$ and recovered $\hat{p}_0(\mathbf{x})$ distributions, when the distribution is learned in the latent space of an autoencoder with encoder f_ψ and decoder g_τ:

$$
\mathcal{W}_2^2(p_0, \hat{p}_0) \leq \|\Delta_{f_\phi, g_\tau}(\mathbf{x})\|^2 + L_{g_\tau}^2 e^{1+2\hat{L}} \int_0^1 \int_{\mathbb{R}^{d/h}} \|v_t(\mathbf{z}_t, t) - \hat{v}_t(\mathbf{z}_t, t)\|^2 q_t^\phi \, d\mathbf{z}dt,
\tag{9}
$$

where v_t and $\hat{v}_t$ are the ground-truth and recovered velocity estimators in the rectified flow framework (Liu et al., 2022), L_{g_τ} is the Lipschitz constant of the decoder and $\hat{L}$ is the Lipschitz constant of the learned velocity estimator $\hat{v}_t$. This upper bound inspired us to minimize the Lipschitz constant of the decoder. To do this, we used R_1-regularization (Mescheder et al., 2018) from the GAN literature (Goodfellow et al., 2014). This was yielding promising initial results and working almost on par with the scale equivariance regularization, but second-order differentiation for R_1 regularization was entailing much slower training speed and engineering struggles (it has poor compatibility with FSDP). This is why we proceeded with scale equivariance which was performing slightly better and much simpler conceptually.

Temporal scale equivariance. We briefly explored temporal scale-equivariance regularization, but surprisingly, it was not leading to improved results. We hypothesize that the temporal domain is of different nature compared to the spatial one since the temporal high-frequency components are more noticeable by human eye than the low-frequency ones.