NOISE-ALIGNED AUTOENCODERS: LEARNING DIFFUSABLE LATENTS WITHOUT END-TO-END TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusability measures how amenable an autoencoder's latent space is to efficient generative modeling, a property that often trades off against reconstruction fidelity. REPA-E improves the diffusability of autoencoders with an end-to-end training paradigm, but it is confined to controlled academic framework, i.e., class-conditional generation on *explicitly labeled* ImageNet. In this paper, we demonstrate that end-to-end tuning might not be essential, and the diffusability gain stems largely from representation alignment (REPA) rather than the diffusion loss. However, applying the REPA loss directly to the autoencoder can slip into a suboptimal solution that oversmooths the latent (e.g., VA-VAE), still lagging behind end-to-end tuned VAEs in the convergence speed of downstream diffusion training. Motivated by these observations, we propose ***N**oise-**A**ligned* Autoencoder (**NA-VAE**), which aligns the noise-conditioned latent with pretrained vision foundation models (VFMs) through a lightweight head with spatial inductive bias, forcing the latent to carry more low-frequency components that noise cannot drown out and thereby forming a structured latent space. NA-VAE is simple (implemented in < 4 lines of code) and efficient, requiring neither end-to-end tuning nor explicitly labeled data. Across VAE families, it consistently improves upon REPA and achieves performance comparable to or better than REPA-E on standard class-conditional and text-to-image benchmarks, while its decoupled training formulation offers potential for extension to video and audio generation.

1 INTRODUCTION

Tokenizers (Kingma & Welling, 2013) serve as the cornerstone of modern visual generative models (Rombach et al., 2022; Peebles & Xie, 2023; Ma et al., 2024; Esser et al., 2024; Blattmann et al., 2023), producing compact and abstract latent representations that obviate the prohibitive complexity of modeling high-dimensional natural signals (e.g., raw pixels) directly. The performance of tokenizers, however, is fundamentally circumscribed by an intrinsic dilemma (Yao et al., 2025) between reconstruction *fidelity* and generative *diffusability* – how amenable the induced latents are to generative modeling. Fidelity favors a high-dimensional latent space and thus rewards the preservation of high-frequency details, yet inordinate high-frequency components render it hard to diffuse.

REPA (Yu et al., 2025; Singh et al., 2026a) has emerged as a powerful technique that aligns the intermediate representations of the diffusion model with pretrained self-supervised vision foundation models (VFMs), significantly accelerating the optimization of generative modeling. REPA-E (Leng et al., 2025a) unlocks the VAE for end-to-end tuning with a latent diffusion model under representation alignment, substantially improving the diffusability of the co-trained VAE, which is released as a byproduct termed E2E-VAE. Despite their practical success, it remains underexplored how end-to-end training improves the diffusability of the VAE, and how VAE training can be scaled to industrial-grade settings that place a greater premium on generative quality while preserving the fidelity of small text (Zhang et al., 2026b) and tiny faces (Wu et al., 2025c) – a regime that lies beyond the purview of controlled academic class-conditional generation on ImageNet (Deng et al., 2009). We ask: ***Is end-to-end training indispensable to improving diffusability?*** The following quote from the REPA-E (Leng et al., 2025a) paper encapsulates the current understanding:

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.

> *"Interestingly, we observe that end-to-end tuning with REPA-E also improves the VAE itself; leading to improved latent space structure and downstream generation performance."*

In this paper, we challenge this intuitive understanding – that the end-to-end diffusion learning process is what impels the VAE towards diffusable latents more conducive to generative modeling – and set out to identify the underlying mechanism that genuinely drives diffusability. Through comprehensive architectural ablations and controlled experiments spanning diverse VAE families, VFMs, backbone scales and downstream tasks, we summarize our findings as follows:

Representation alignment rather than end-to-end tuning drives the diffusability. We isolate VAE training from the diffusion model while leaving the original alignment path intact: the latent is still propagated through the first eight SiT blocks, yet the diffusion branch is removed entirely. Contrary to prior belief, the resulting latents rival those of the corresponding E2E-VAE released by REPA-E (Leng et al., 2025a) in generative performance (i.e., diffusability) across disparate VAE families (e.g., SD-VAE (Rombach et al., 2022), VA-VAE (Yao et al., 2025) and FLUX-VAE (Black Forest Labs, 2024)), even though they are never learned end-to-end with a diffusion loss. Visualizations in Fig. 1 of the latent structure further indicate the two paradigms are almost equivalent.

Aligning the clean latent captures only a fraction of the gain. We further probe the core ingredient of diffusability by disabling the noise sampling process, which can be viewed as VA-VAE (Yao et al., 2025) with its MLP alignment head replaced by a stack of SiT (Ma et al., 2024) blocks. Latent visualizations in Fig. 1 and quantitative analyses in §3.2 reveal that the latent retains excessive high-frequency details, a suboptimal solution that manifests in inferior generative performance.

Noise and spatial inductive bias synergistically improve diffusability. Through exhaustive ablations over the noising scheme, as summarized in Tab. 1 and §4.2, we find that noise plays a pivotal role in compelling the VAE to retain more low-frequency signals in its latent, which are far less likely to be drowned out and hence remain recoverable by the denoising process, rendering the latent more amenable to generative modeling. Moreover, spatial inductive bias synergistically amplifies this effect, to the point that a lightweight three-layer convolutional head alone attains diffusability competitive with end-to-end tuning. This finding resonates with the conclusion of i-REPA (Singh et al., 2026a), which we extend from representation alignment with the diffusion model to the VAE itself.

We highlight the contributions of our study below:

- We meticulously conduct architectural ablations and reveal that the genuine source of diffusability is representation alignment imposed directly on the VAE, rather than its end-to-end coupling with the diffusion model. To the best of our knowledge, this is the first demonstration that end-to-end tuning is not indispensable to attaining diffusability comparable to that of E2E-VAE.
- Through systematic analysis and controlled experiments, we pinpoint the mechanism underlying this gain in diffusability: a synergy between noise and spatial inductive bias that prevents the optimization from collapsing into the suboptimal solution reached by clean-latent alignment.
- We propose **NA-VAE** (**N**oise-**A**ligned VAE), which aligns the noise-conditioned latent via a lightweight spatial head to match or surpass the diffusability of end-to-end tuning across diverse VAEs / downstream tasks. Our method is simple (<4 lines of code), efficient (no DiT or diffusion loss), and no explicit labels (scalable to industrial-grade settings and video / audio generation).

2 PRELIMINARIES

Given continuous target distribution $p(\boldsymbol{x})$, latent diffusion model v_θ (Rombach et al., 2022) approximates $p(\boldsymbol{z})$ using sampled data $\boldsymbol{x} \in \mathcal{X}$ within the compact latent space $\boldsymbol{z} = \mathcal{E}(\boldsymbol{x})$ to alleviate the computational overhead, which is constructed by a pretrained autoencoder $\mathcal{V} = \{\mathcal{E}, \mathcal{D}\}$. The autoencoder $\mathcal{V}$ is optimized for fidelity alone, combining reconstruction, perceptual (Zhang et al., 2018), Kullback–Leibler and adversarial terms, $\mathcal{L}_{\mathcal{E},\mathcal{D}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}$. The diffusion model is trained in isolation on the frozen latent $\boldsymbol{z}$, typically under the linear interpolant $\boldsymbol{z}_t = (1-t)\boldsymbol{z} + t\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $t \in [0,1]$, by regressing the velocity field (Ma et al., 2024), $\mathcal{L}_\theta = \mathbb{E}_{\boldsymbol{z},\boldsymbol{\epsilon},t}\big[\|v_\theta(\boldsymbol{z}_t, t) - (\boldsymbol{\epsilon} - \boldsymbol{z})\|_2^2\big]$, where $\boldsymbol{h}_t = v_\theta^{(\ell)}(\boldsymbol{z}_t, t) \in \mathbb{R}^{N\times d}$ denotes its hidden states on the N latent tokens after the first ℓ blocks. Diffusability (Yao et al., 2025; Leng

et al., 2025a; Skorokhodov et al., 2025; Yu et al., 2025; Singh et al., 2026a; Kouzelis et al., 2025a) measures how readily the latent space $\boldsymbol{z} = \mathcal{E}(\boldsymbol{x})$ obtained from $\mathcal{V}$ can be modelled by v_θ.

VA-VAE (Yao et al., 2025) identifies the dilemma between reconstruction and generation, and aligns the clean latent $\boldsymbol{z}$ with a self-supervised VFM throughout the training of the VAE,

$$\mathcal{L}_{\mathcal{E},\mathcal{D}}^{\text{VA-VAE}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{VF}}\mathcal{L}_{\text{VF}}(\boldsymbol{W}\boldsymbol{z}, \boldsymbol{y}) \tag{1}$$

where the alignment loss $\mathcal{L}_{\text{VF}}$ anchors the linear projected latent $\boldsymbol{W}\boldsymbol{z}$ to $\boldsymbol{y} = f(\boldsymbol{x})$, the patch-wise features of a frozen VFM f, under a cosine objective and its pairwise counterpart.

REPA (Yu et al., 2025) significantly accelerates the convergence of the diffusion model by aligning its intermediate hidden states $\boldsymbol{h}_t$ with the VFM features $\boldsymbol{y}$ through a trainable MLP head h_ω.

$$\mathcal{L}_{\theta,\omega}^{\text{REPA}} = \mathbb{E}_{\boldsymbol{z},\boldsymbol{\epsilon},t}\left[\|v_\theta(\boldsymbol{z}_t, t) - (\boldsymbol{\epsilon} - \boldsymbol{z})\|_2^2\right] + \lambda_{\text{REPA}}\mathcal{L}_{\text{REPA}}(h_\omega(\boldsymbol{h}_t), \boldsymbol{y}) \tag{2}$$

REPA-E (Leng et al., 2025a) unlocks VAE to enable end-to-end tuning, thereby improving diffusability, and **splits every update on a batch into two substeps**. The first opens $\mathcal{E}$ and $\mathcal{D}$ while the diffusion transformer v_θ and the head h_ω are held fixed, so that the fidelity terms $\mathcal{L}_{\mathcal{E},\mathcal{D}}$ and the alignment loss $\mathcal{L}_{\text{REPA}}$ update the autoencoder $\mathcal{V}$ alone; the second freezes $\mathcal{V}$ by a stop-gradient $\text{sg}[\cdot]$ on the latent and updates v_θ and h_ω with the diffusion loss and the alignment loss.

$$\underbrace{\mathcal{L}_{\mathcal{E},\mathcal{D}} + \lambda_{\text{REPA}}\mathcal{L}_{\text{REPA}}}_{\text{step 1: update } \mathcal{V}} + \underbrace{\mathbb{E}_{\boldsymbol{z},\boldsymbol{\epsilon},t}\left[\|v_\theta(\text{sg}[\boldsymbol{z}_t], t) - (\boldsymbol{\epsilon} - \text{sg}[\boldsymbol{z}])\|_2^2\right]}_{\text{step 2: update } v_\theta, h_\omega} \tag{3}$$

3 NA-VAE: Noise-Aligned Autoencoder

3.1 Motivation

Although REPA-E has obtained E2E-VAE with markedly improved diffusable latents, it remains confined to label-conditioned training framework, which makes it inconvenient to apply directly to industrial-grade tokenizer training. For instance, 1) labeled data at billion scale is expensive to collect and difficult to balance over a long-tailed distribution; 2) training on an existing label-conditioned dataset (e.g., ImageNet (Deng et al., 2009)) improves diffusability at the cost of degraded reconstruction in specialized domains (e.g., small text (Zhang et al., 2026b) and tiny faces (Wu et al., 2025c)); 3) transplanting REPA-E into a text-to-image framework is resource-intensive and unstable, and are exacerbated when training a unified tokenizer, whose objective already comprises other loss terms (e.g., CLIP loss (Yao et al., 2026), NTP loss (Zhang et al., 2026a)).

Is end-to-end tuning essential to acquire diffusability? At first glance the stop-gradient in equation 3 already answers in the affirmative: within a single update the encoder is driven by the fidelity terms and the alignment loss alone, $\mathcal{L}_{\mathcal{E},\mathcal{D}} + \lambda_{\text{REPA}}\mathcal{L}_{\text{REPA}}(h_\omega(\boldsymbol{h}_t), \boldsymbol{y})$. However, across updates, the two substeps remain tightly coupled: the diffusion loss in the second substep updates v_θ, whose first ℓ blocks produce the hidden states $\boldsymbol{h}_t$ through which the first substep aligns the latent to $\boldsymbol{y}$. **In short, the diffusion loss shapes the alignment of the autoencoder (the first substep at batch $n{+}1$) by updating the first ℓ SiT layers (the second substep at batch n).** Decoupling the diffusion loss from substeps therefore amounts to probing the contribution of end-to-end tuning to diffusability.

3.2 Observations

Learning diffusable latents without end-to-end tuning. We keep the alignment path of equation 3 intact – the noised latent is still forwarded through the first ℓ SiT blocks and the head h_ω – **but unfreeze those blocks and remove the diffusion branch entirely**. The autoencoder and the alignment path are thus updated within one substep, with the alignment loss as the only supervision beyond fidelity, so that no diffusion loss, no DiT training substep and no class labels enter this recipe. Across the autoencoder families (Tab. 1), surprisingly, the resulting latents rival or surpass those of the E2E-VAE released by REPA-E in generative performance under an identical downstream budget. We further investigate it and visualize their latents by projecting the latent channels onto three leading principal components and mapping them to RGB. As observed in Fig. 1, the two paradigms exhibit a highly analogous latent structure. Details of visualization protocol are provided in §A.

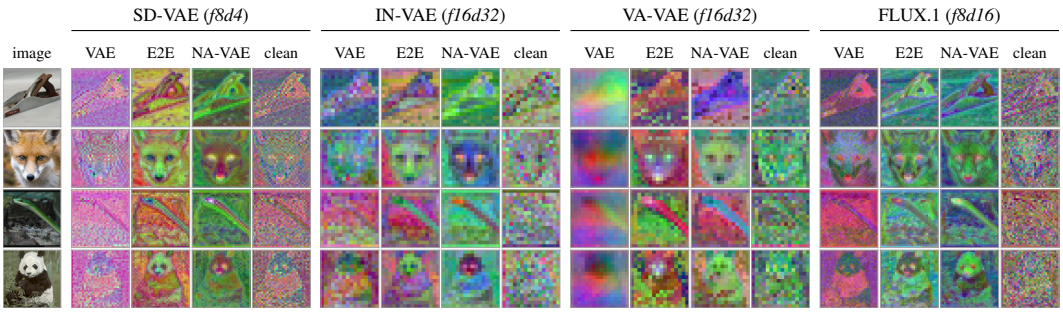


Fig. 1: **Aligning the noised latent reorganizes the latent space the way end-to-end tuning does.** Top three PCA components of the latent, mapped to RGB, for four autoencoders (blocks) on four ImageNet images (rows). Each block shows the original VAE, the E2E-VAE (end-to-end tuned) released by Leng et al. (2025a), NA-VAE (ours), and our own recipe with the noise removed (*clean*, i.e. clean-latent alignment under an identical budget). The PCA basis is fit per tokenizer, so hue is arbitrary and only spatial structure is comparable. Original VAEs spend much of their capacity on high-frequency texture; E2E-VAE and NA-VAE both reorganize them into object-level regions, while clean-latent alignment does not — Aligning the clean latent thus picks up the semantics while carrying the excessive details along with it. See §A for visualization protocol and more results.

Finding 1. *End-to-end tuning is not indispensable to diffusability; its genuine source remains the classical representation alignment.*

Tab. 1: **Quantitative results on variations of the alignment head architecture, with and without added noise, across VAE families, as a complement to Fig. 1 and Fig. 3.** One subtable per tokenizer family, each listing the released tokenizer (VAE), the end-to-end tuned E2E-VAE of Leng et al. (2025a), and NA-VAE under eight variants of the alignment path: SiT blocks of depth $8/4/1/0$, a three-layer pointwise stack (MLP) and the same stack with 3×3 kernels (conv). On the NA-VAE rows the suffix names what the head reads, the noised latent $\boldsymbol{z}_t$ or the clean one $\boldsymbol{z}$. See §C.4.

(a) SD-VAE (*f8d4*)

	gFID↓	sFID↓	IS↑	Pre.↑	Rec.↑
VAE	7.72	5.06	127.9	0.700	**0.652**
E2E-VAE	5.00	4.61	146.3	0.742	0.630
SiT-8-$\boldsymbol{z}_t$	4.92	5.62	148.5	0.754	0.624
SiT-8-$\boldsymbol{z}$	7.41	4.79	125.1	0.718	0.642
SiT-4-$\boldsymbol{z}_t$	4.53	5.27	150.9	0.758	0.621
SiT-1-$\boldsymbol{z}_t$	**4.47**	4.60	**154.9**	**0.764**	0.617
SiT-0-$\boldsymbol{z}_t$	6.64	6.08	134.5	0.745	0.608
MLP-$\boldsymbol{z}$	7.02	6.98	133.6	0.759	0.587
MLP-$\boldsymbol{z}_t$	8.18	6.14	123.3	0.733	0.605
conv-$\boldsymbol{z}_t$	5.20	**4.55**	144.9	0.759	0.618

(b) IN-VAE (*f16d32*)

	gFID↓	sFID↓	IS↑	Pre.↑	Rec.↑
VAE	7.40	4.87	124.9	0.710	**0.650**
E2E-VAE	4.06	4.47	151.6	0.749	0.636
SiT-8-$\boldsymbol{z}_t$	4.13	4.81	152.0	0.753	0.638
SiT-8-$\boldsymbol{z}$	7.31	5.28	124.6	0.714	0.644
SiT-4-$\boldsymbol{z}_t$	3.96	4.54	154.1	0.755	0.642
SiT-1-$\boldsymbol{z}_t$	**3.79**	4.66	**156.4**	0.761	0.634
SiT-0-$\boldsymbol{z}_t$	4.87	4.50	145.4	0.754	0.628
MLP-$\boldsymbol{z}$	4.74	4.48	147.6	**0.763**	0.620
MLP-$\boldsymbol{z}_t$	4.91	4.87	146.3	0.753	0.625
conv-$\boldsymbol{z}_t$	4.13	**4.35**	152.9	0.759	0.635

(c) VA-VAE (*f16d32*)

	gFID↓	sFID↓	IS↑	Pre.↑	Rec.↑
VAE	4.63	5.48	149.7	0.762	0.608
E2E-VAE	3.36	**4.20**	160.6	0.773	0.619
SiT-8-$\boldsymbol{z}_t$	**3.00**	4.57	166.6	**0.779**	**0.634**
SiT-8-$\boldsymbol{z}$	5.19	4.85	141.0	0.749	0.626
SiT-4-$\boldsymbol{z}_t$	3.10	4.40	167.5	0.778	0.624
SiT-1-$\boldsymbol{z}_t$	3.06	4.90	**170.1**	**0.779**	0.623
SiT-0-$\boldsymbol{z}_t$	3.68	5.06	160.4	0.773	0.614
MLP-$\boldsymbol{z}$	4.15	5.33	157.8	0.771	0.611
MLP-$\boldsymbol{z}_t$	3.85	5.48	158.7	0.773	0.614
conv-$\boldsymbol{z}_t$	3.25	4.90	165.0	0.773	0.632

Aligning clean latent makes a trivial solution. We further study the underlying mechanism that actually shapes diffusability. Contrasting our recipe with VA-VAE introduced in §2, once the diffusion loss is removed, a potential source of the gain is the noise: the alignment path reads a noised latent $\boldsymbol{z}_t$, whereas VA-VAE aligns the clean $\boldsymbol{z}$. We therefore switch off the noise sampling throughout training and align the clean latent instead, $\mathcal{L}_{\mathcal{E},\mathcal{D}} + \lambda_{\text{REPA}}\mathcal{L}_{\text{REPA}}(h_\omega(v_\theta^{(\ell)}(\boldsymbol{z}, 0)), \boldsymbol{y})$, where the head now reads the clean latent $\boldsymbol{z}$ rather than its noised counterpart $\boldsymbol{z}_t$ in equation 3. Quantitative results in Tab. 1 (the SiT-8-$\boldsymbol{z}_t$ / SiT-8-$\boldsymbol{z}$ pair), visualizations in Fig. 1 and Fig. 2 show that the noise plays a pivotal role in shaping the latent: aligning the clean latent does improve generation by enriching the semantics of the latent, as stated in VA-VAE (Yao et al., 2025), yet it remains a suboptimal solution, in which excessive high-frequency detail is carried along and caps the attainable diffusability.

Finding 2. *Noise plays a pivotal role in shaping diffusability: aligning the clean latent directly picks the low-hanging fruit, yet caps the ceiling it can reach.*

Noise and spatial inductive bias make diffusability. Beyond the noise, our recipe also departs from VA-VAE in the structure of the alignment path (SiT blocks vs. MLP head). To examine this, we ablate the number of SiT layers, the positional embedding and a plain convolutional stack, under

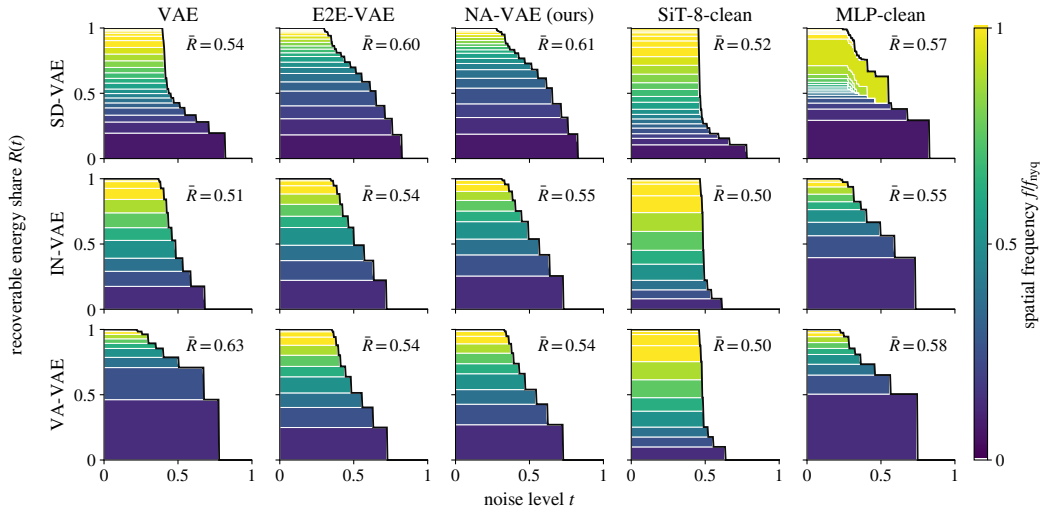


Fig. 2: **Quantitative frequency analysis of the latent space.** For each panel, the x axis is the noise level t of the linear interpolant $\boldsymbol{z}_t$, the y axis is the fraction of the latent's energy that still stands above the noise at t, and the stacked areas split that fraction by spatial frequency. Each band drops out at the t where the noise overtakes it, which shows the order in which scales are lost as it gets noisier. Detailed protocol are defined in §B.

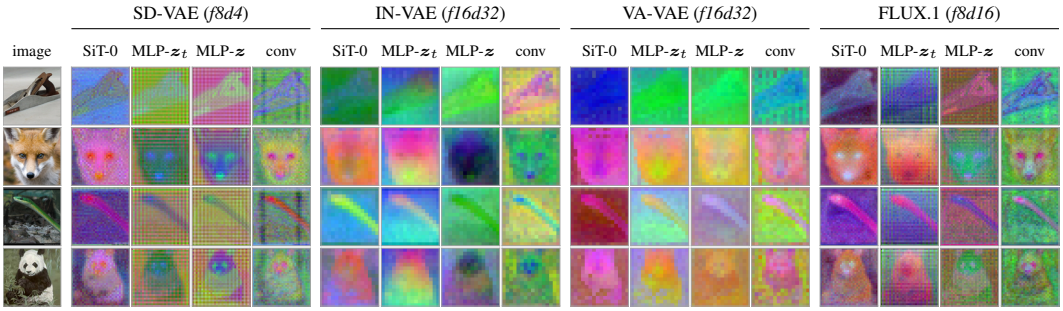


Fig. 3: **Spatial inductive bias of the align head and noise schema synergistically improve latent diffusability.** Same protocol as Fig. 1, with the depth-8 SiT head replaced by four weaker ones: a depth-0 SiT head that keeps only the positional embeddings (SiT-0), a three-layer pointwise stack reading the noised latent (MLP-$\boldsymbol{z}_t$) or the clean one (MLP-$\boldsymbol{z}$), and the same stack with 3×3 kernels (conv). Positional information, i.e. the spatial inductive bias of the head, is essential to the alignment of the noised latent with the target; under appropriate noise schema, even a plain convolutional stack yields a latent structure comparable to that of end-to-end tuning.

both noised and clean latent. As presented in Tab. 1, Fig. 2 and Fig. 3, we find that the spatial inductive bias also contributes to better diffusability. Comparing the head that retains the positional embedding against the pointwise MLP one shows that positional information removes the grid-like artifacts in the latent. This finding echoes the conclusion of i-REPA (Singh et al., 2026a), except that we extend it from representation alignment with the diffusion model to the learning of the VAE.

> **Finding 3.** *Noise and spatial inductive bias synergistically improve latent diffusability; with a proper design, even a single transformer layer or a plain convolutional stack suffices to match the generation performance of end-to-end tuning.*

3.3 Noisy Latent Representation Alignment with Spatial Inductive Bias

Based on the observations illustrated in §3.2, we propose the **N**oise-**A**ligned autoencoder (NA-VAE), a simple and efficient method that matches or surpasses the diffusability of the VAE obtained by end-to-end tuning, while requiring no DiT training, no diffusion loss and no label-conditioned data. NA-VAE leaves the fidelity objective of the autoencoder untouched and attaches a single alignment

Algorithm 1 VA-VAE (Yao et al., 2025)

Input: $\mathcal{E}, \mathcal{D}$; frozen VFM f; linear projection $\boldsymbol{W}$

while not converged **do**
  $\boldsymbol{x} \sim p(\boldsymbol{x}), \quad \boldsymbol{y} \leftarrow f(\boldsymbol{x})$
  $\boldsymbol{z} \leftarrow \mathcal{E}(\boldsymbol{x})$
  ▷ *no noise is sampled*
  $\mathcal{L}_{\text{align}} \leftarrow \mathcal{L}_{\text{VF}}(\boldsymbol{W}\boldsymbol{z}, \boldsymbol{y})$
  $\mathcal{L} \leftarrow \mathcal{L}_{\mathcal{E},\mathcal{D}} + \lambda \mathcal{L}_{\text{align}}$
  update $\mathcal{E}, \mathcal{D}, \boldsymbol{W}$ with $\nabla \mathcal{L}$
end while

Algorithm 2 NA-VAE (ours)

Input: $\mathcal{E}, \mathcal{D}$; frozen VFM f; spatial head h_ω

while not converged **do**
  $\boldsymbol{x} \sim p(\boldsymbol{x}), \quad \boldsymbol{y} \leftarrow f(\boldsymbol{x})$
  $\boldsymbol{z} \leftarrow \mathcal{E}(\boldsymbol{x})$
  $t \sim p(t), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad \boldsymbol{z}_t \leftarrow (1-t)\boldsymbol{z} + t\boldsymbol{\epsilon}$
  $\mathcal{L}_{\text{align}} \leftarrow \mathcal{L}_{\text{REPA}}(h_\omega(\boldsymbol{z}_t, t), \boldsymbol{y})$
  $\mathcal{L} \leftarrow \mathcal{L}_{\mathcal{E},\mathcal{D}} + \lambda \mathcal{L}_{\text{align}}$
  update $\mathcal{E}, \mathcal{D}, h_\omega$ with $\nabla \mathcal{L}$
end while

term to it. At every step we draw a timestep t and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, form the interpolant $\boldsymbol{z}_t$ exactly as the diffusion model would, and ask the latent to stay alignable to the VFM features *through the noise*: a lightweight head h_ω carrying spatial inductive bias reads $\boldsymbol{z}_t$ and is matched to $\boldsymbol{y} = f(\boldsymbol{x})$,

$$\mathcal{L}^{\text{NA-VAE}}_{\mathcal{E},\mathcal{D},\omega} = \mathcal{L}_{\mathcal{E},\mathcal{D}} + \lambda_{\text{REPA}} \mathbb{E}_{\boldsymbol{\epsilon},t}\big[\mathcal{L}_{\text{REPA}}(h_\omega(\boldsymbol{z}_t, t), \boldsymbol{y})\big] \tag{4}$$

where h_ω is trained jointly with the autoencoder and discarded afterwards.

Why noise and spatial inductive bias. Under the interpolant every latent component is attenuated by $(1 - t)$ and buried under noise of scale t, and the components of low amplitude or high spatial frequency are the first to be drowned, as indicated in Fig. 2. Since the alignment has to hold in expectation over t, the encoder cannot afford to carry semantics in the channels that the noise erases, and is thus forced to spend its capacity on the low-frequency structure that survives large t – precisely the part of the signal that the diffusion model recovers first along its trajectory, which in turn accelerates the learning of the diffusion model. The spatial head amplifies this effect: reading a neighbourhood rather than a single position, it lets the encoder satisfy the alignment with spatially coherent structure, whereas a pointwise head can only ask each position to be independently legible and therefore tolerates the grid-like artifacts observed in §3.2.

Implementation. NA-VAE can be implemented in <4 lines of code on top of a VA-VAE style training loop (Alg. 1 *vs.* Alg. 2): sample t, form $\boldsymbol{z}_t$, and read it with h_ω in place of the linear projection $\boldsymbol{W}$. In this sense, NA-VAE can be viewed as a bridge between VA-VAE and REPA-E.

4 Experiments

4.1 Setup

Training. We follow the same setup as in SiT (Ma et al., 2024) and REPA (Yu et al., 2025) unless otherwise specified. For the ablations, we train on the ImageNet (Deng et al., 2009) training split under the same preprocessing protocol, and experiment with SD-VAE (*f8d4*) (Rombach et al., 2022), IN-VAE (*f16d32*) (Leng et al., 2025a) and VA-VAE (*f16d32*) (Yao et al., 2025). We adopt SiT-XL/2 and SiT-XL/1 according to the downsampling rate of the tokenizer, so that all families are modelled at the same compute cost. We use DINOv2 (Oquab et al., 2024) as the external visual features and apply the alignment to the eighth layer of the SiT model. We train NA-VAE for 20 epochs in ablations and 80 epochs for Tab. 2. Specific training hyperparameters are provided in §C.1.

Evaluation. We report the classical FID-50K on the ADM setup (Dhariwal & Nichol, 2021), including Fréchet inception distance (gFID) (Heusel et al., 2017), spatial FID (sFID) (Nash et al., 2021), inception score (IS) (Salimans et al., 2016), precision (Prec.) and recall (Rec.) (Kynkäänniemi et al., 2019), using the SDE Euler–Maruyama sampler with 250 steps. See full evaluation protocol in §C.2.

4.2 Ablation Study

Architecture of Alignment Head. As presented in Tab. 1 and Fig. 4a, we ablate the architecture of the alignment head under the same training recipe: 20 epochs of VAE training, followed by 80 epochs of downstream SiT training. We find that the spatial inductive bias of the head is crucial to improving diffusability. Even a simple convolutional head already matches or surpasses end-to-end tuning. A single SiT block is the most balanced, and we adopt it by default afterwards.

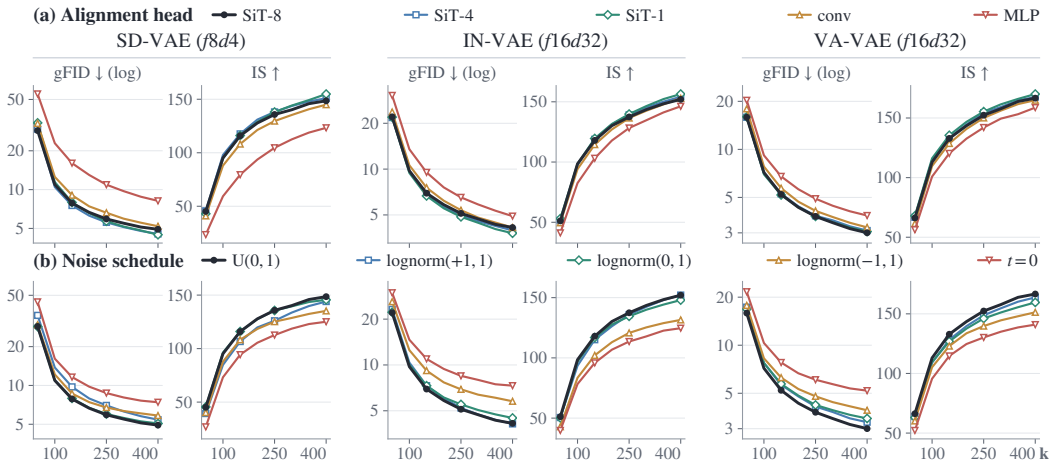


Fig. 4: **Downstream convergence under the two design axes.** gFID (log scale) and IS against downstream SiT training steps. See §C.4 and §C.5 for detailed results.

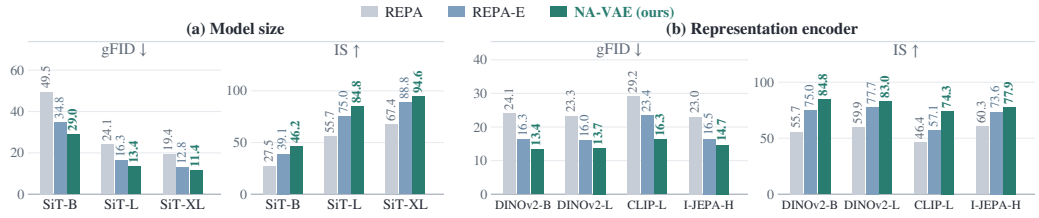


Fig. 5: **Ablation on model size and alignment encoder.** All results are obtained after 100k downstream SiT training steps. See §C.6 and §C.7 for detailed results.

Noise Schedule. We next examine how the sampling distribution of the noise level t affects downstream generation while keeping the alignment head fixed. Fig. 4b compares uniform sampling with three logit-normal schedules favoring high, intermediate, or low noise levels, as well as a no-noise baseline ($t = 0$). All noisy schedules improve generation over clean alignment, with uniform sampling achieving the best or near-best gFID and IS across all three tokenizer families. We therefore adopt $t \sim \mathcal{U}(0, 1)$ by default, avoiding additional schedule-specific hyperparameters.

Model Size. We evaluate NA-VAE (training from SD-VAE) across three downstream backbone scales, SiT-B/2, SiT-L/2, and SiT-XL/2. At a common budget of 100k downstream training steps, NA-VAE achieves lower gFID and higher IS than both REPA and REPA-E at every scale (Fig. 5a). These results show that the benefits of tokenizer-side persist across the backbone scales evaluated.

Representation Encoder. We compare four frozen representation encoders—DINOv2-B and DINOv2-L (Oquab et al., 2024), CLIP-L (Radford et al., 2021), and I-JEPA-H (Assran et al., 2023)—while initializing the tokenizer from SD-VAE and fixing the downstream backbone to SiT-L/2. At 100k downstream training steps, NA-VAE outperforms both REPA and REPA-E in gFID and IS with all four encoders (Fig. 5b). Among the four pretrained vision encoders, DINOv2-B yields the best performance, supporting its use as our default representation encoder.

Component Analysis. Fig. 6 compares denoising without alignment (*noise only*), clean-latent alignment (*align only*), and NA-VAE (*noise + align*). The two alignment variants share the same eight-block SiT head and clean-latent reconstruction path. Across all three tokenizer families, NA-VAE achieves lower gFID and higher IS than both alternatives at every evaluated downstream checkpoint. These results support learning representations that remain aligned with VFM features under noise as a key ingredient for improving diffusability, beyond latent denoising alone.

4.3 Improving Diffusability without End-to-End Tuning

Tab. 2 reports system-level generation results on ImageNet 256×256. We extend the fine-tuning of NA-VAE for 80 epochs, freeze the tokenizer, and train a downstream SiT-XL with REPA. With SD-VAE initialization, NA-VAE improves gFID from 7.90 to 4.04 and IS from 122.6 to 159.0 over

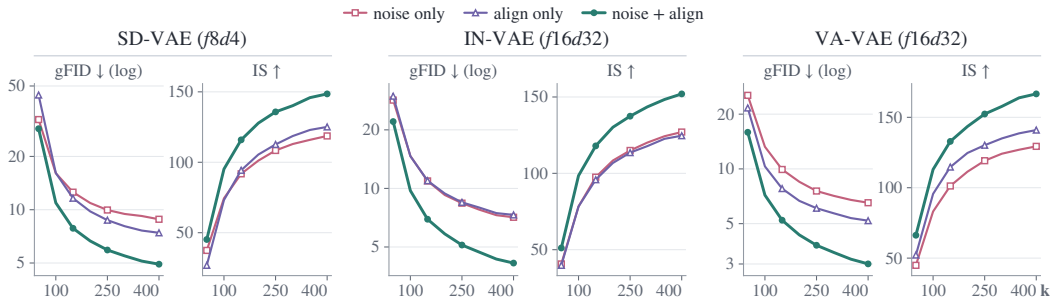


Fig. 6: **Component analysis of noise and representation alignment.** *Noise only* reconstructs images from noised latents without alignment; *align only* aligns clean latents; and *noise + align* (NA-VAE) aligns noised latents while retaining clean-latent reconstruction. See §C.8 for details.

Tab. 2: **System-level performance on ImageNet** 256×256**.** E2E-VAE$_{\text{VA-VAE}}$ and NA-VAE$_{\text{VA-VAE}}$ both start from VA-VAE (Yao et al., 2025), and NA-VAE$_{\text{SD-VAE}}$ from SD-VAE. NA-VAE is trained for 80 epochs and then kept frozen; the generator is SiT-XL trained with REPA as in §4.1.

Tokenizer	Method	Epochs	#params	rFID↓	Generation w/o CFG					Generation w/ CFG				
					gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Latent Diffusion Models (LDM)														
SD-VAE	MaskDiT (Zheng et al., 2023)	1600	675M	0.61	5.69	10.34	177.9	0.74	0.60	2.28	5.67	276.6	0.80	0.61
	DiT (Peebles & Xie, 2023)	1400	675M		9.62	6.85	121.5	0.67	0.67	2.27	4.60	278.2	**0.83**	0.57
	SiT (Ma et al., 2024)	1400	675M		8.61	6.32	131.7	0.68	0.67	2.06	4.50	270.3	0.82	0.59
	FasterDiT (Yao et al., 2024)	400	675M		7.91	5.45	131.3	0.67	**0.69**	2.03	4.63	264.0	0.81	0.60
Representation Alignment Methods														
VA-VAE	LightningDiT (Yao et al., 2025)	80	675M	**0.28**	4.29	-	-	-	-	-	-	-	-	-
SD-VAE	REPA (Yu et al., 2025)	80	675M	0.61	7.90	5.06	122.6	0.70	0.65	-	-	-	-	-
		800	675M		5.84	5.79	158.7	0.70	0.68	**1.28**	4.68	305.7	0.79	0.64
E2E-VAE$_{\text{VA-VAE}}$	REPA-E (Leng et al., 2025a)	80	675M	**0.28**	3.46	**4.17**	159.8	0.77	0.63	1.67	**4.12**	266.3	0.80	0.63
NA-VAE$_{\text{SD-VAE}}$	REPA (Yu et al., 2025)	80	675M	0.49	4.04	4.70	159.0	0.77	0.62	1.75	5.01	289.3	0.80	0.61
		480	675M		**2.53**	4.52	**196.6**	0.78	0.64	1.42	4.88	**308.9**	0.79	**0.65**
NA-VAE$_{\text{VA-VAE}}$	REPA (Yu et al., 2025)	80	675M	**0.28**	2.82	4.70	175.9	**0.79**	0.62	1.61	4.67	274.7	0.80	0.62

REPA. With VA-VAE initialization, it achieves 2.82 gFID and 175.9 IS, compared with 3.46 and 159.8 achieved by E2E-VAE. To assess whether NA-VAE scales to industrial-grade tokenizers, we train a FLUX.1-style tokenizer (*f8d16*) (Black Forest Labs, 2024) from scratch on our internal image data and compare it with their end-to-end tuned counterparts (Leng et al., 2025b) (Fig. 7 left). These results show that NA-VAE can yield diffusability competitive with end-to-end-tuned tokenizers.

4.4 Generality

Text to Image. We evaluate text-to-image generation using the NA-VAE trained from scratch on our internal image corpus in §4.3. Following the downstream training recipe of Leng et al. (2025b), we freeze each tokenizer and train a 3B DiT using Fuse-DiT (Tang et al., 2025) on BLIP-3o image–text pairs (Chen et al., 2025c). We conduct training on the 256×256 stage. Fig. 7 (right) shows that NA-VAE consistently outperforms the pretrained FLUX.1 and Qwen-Image VAE (Wu et al., 2025a) on GenEval (Ghosh et al., 2023), DPG-Bench (Hu et al., 2024), and GenAI-Bench (Li et al., 2024a). These results demonstrate competitive text-to-image alignment without end-to-end tokenizer tuning.

512×512 Resolution. We evaluate NA-VAE on ImageNet at 512×512 using SiT-L/2 and the SD-VAE backbone. Fig. 8 compares it against the original SD-VAE and E2E-VAE$_{\text{SD-VAE}}$. NA-VAE reaches the gFID at 400k steps that end-to-end tuning attains at 1M steps.

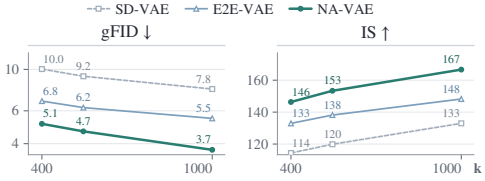


Fig. 8: **Scaling to** 512×512 **Resolution.**

Video and Audio Generation. NA-VAE can even generalize to video and audio modalities (Fig. 9). On video generation, NA-VAE$_{\text{Wan2.1}}$ reduces UCF-101 gFVD by 33.8% over Wan2.1 VAE at 300k downstream steps. On audio generation, NA-VAE$_{\text{Acoustic}}$ improves LibriSpeech-PC SIM by 0.0134 and lowers Seed-TTS-Eval WER by 22.4% over acoustic-only fine-tuning at 400k steps.

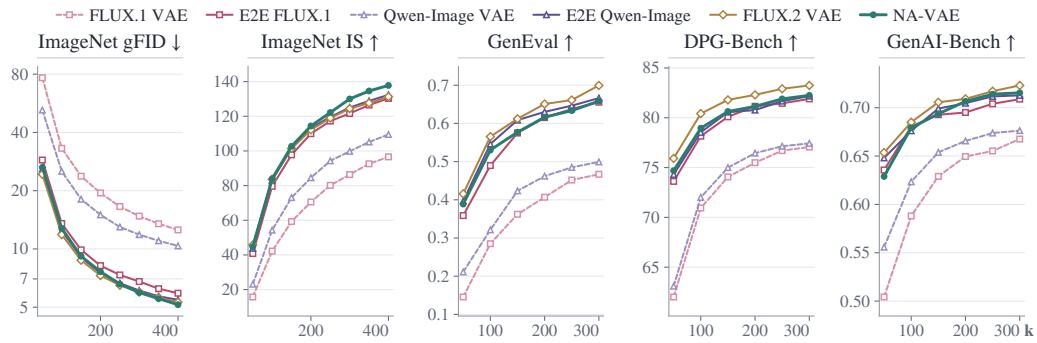


Fig. 7: **Comparison with Industrial Tokenizers.** *Left*: class-conditional generation on ImageNet 256 × 256 over 400k steps of SiT-XL. *Right*: GenEval, DPG-Bench, and GenAI-Bench over 300k steps of a 3B DiT on BLIP-3o image–text pairs. See §C.9 for more detailed settings and results.

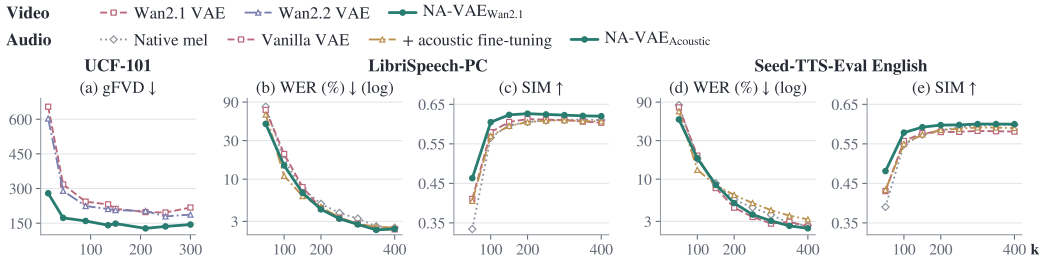


Fig. 9: **Video and audio generation with NA-VAE.** See §C.10 and §C.11 for more details.

5 Related Work

We briefly present the most relevant literature here and provide an extended discussion in §D.

Improving diffusability of visual tokenizers. VA-VAE (Yao et al., 2025) aligns clean latents with pretrained visual features. Other approaches improve latent structure through spatial equivariance (Kouzelis et al., 2025a; Zhou et al., 2025), scale-equivariant regularization (Skorokhodov et al., 2025), and reconstruction from corrupted latents (Yang et al., 2025). RAE (Zheng et al., 2025) directly uses pretrained encoder features. Subsequent work scales tokenizer pretraining with diverse objectives (Yao et al., 2026), refines semantic alignment (Zhang et al., 2026b), and combines representation autoencoders with complementary diffusion-feature alignment (Singh et al., 2026b).

Representation alignment for generation. Prior work leverages pretrained visual representations to improve diffusion-based generation (Pernias et al., 2024; Fuest et al., 2024). REPA (Yu et al., 2025) aligns diffusion features with clean-image representations, and REPA-E (Leng et al., 2025a) extends alignment to end-to-end tokenizer tuning. Joint image–feature modeling (Kouzelis et al., 2025b) provides another route, while alignment extends to video generation (Zhang et al., 2025) and 3D-consistent world modeling (Wu et al., 2025b). i-REPA (Singh et al., 2026a) further emphasizes spatial structure. Drawing inspiration from these works, we shift the focus of representation alignment from diffusion training to improving the diffusability of the tokenizer itself.

6 Conclusion

In this paper, we demonstrate that end-to-end tuning is not indispensable for improving latent diffusability. Controlled ablations highlight the complementary roles of noise and spatial inductive bias in representation alignment, beyond clean-latent alignment or denoising alone. Building on these findings, NA-VAE aligns noised latents with pretrained visual representations through a lightweight spatial head, without requiring a diffusion loss or explicit class labels for tokenizer training. Experiments across diverse settings demonstrate its competitive downstream generation relative to end-to-end-tuned tokenizers. Our approach even generalizes directly to video and audio modalities.

AI USE STATEMENT

Guided by the principles of originality, caution, and honesty, we used AI to refine the manuscript's language and presentation, organize data from experimental logs and prepare tables and figures, and implement utility functions. The scientific narrative, research ideas, and experiments are original contributions of the authors.

REPRODUCIBILITY STATEMENT

We provide all implementation details and hyperparameters in §C.1 and §C.2. We will also open-source our code, model checkpoints and analysis results.

REFERENCES

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, et al. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*, 2024.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. FlexTok: Resampling images into 1D token sequences of flexible length. In *International Conference on Machine Learning*, 2025.

Black Forest Labs. FLUX. `https://github.com/black-forest-labs/flux`, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

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 *International Conference on Machine Learning*, 2025a.

Hao Chen, Ze Wang, Xiang Li, Ximeng Sun, Fangyi Chen, Jiang Liu, Jindong Wang, Bhiksha Raj, Zicheng Liu, and Emad Barsoum. SoftVQ-VAE: Efficient 1-dimensional continuous tokenizer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025b.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025c.

Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. In *International Conference on Learning Representations*, 2025d.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, JianZhao JianZhao, Kai Yu, and Xie Chen. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6255–6271, 2025e.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, pp. 8780–8794, 2021.

Jiayu Du, Xingyu Na, Xuechen Liu, and Hui Bu. Aishell-2: Transforming mandarin asr research into industrial scale. *arXiv preprint arXiv:1808.10583*, 2018.

Sinan Du, Jiahao Guo, Bo Li, Shuhao Cui, Zhengzhuo Xu, Yifu Luo, Yongxian Wei, Kun Gai, Xinggang Wang, Kai Wu, and Chun Yuan. VQRAE: Representation quantization autoencoders for multimodal understanding, generation and reconstruction. arXiv preprint arXiv:2511.23386, 2025. URL https://arxiv.org/abs/2511.23386.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Michael Fuest, Pingchuan Ma, Ming Gui, Johannes Schusterbauer, Vincent Tao Hu, and Bjorn Ommer. Diffusion models and representation learning: A survey. *arXiv preprint arXiv:2407.00783*, 2024.

Yuan Gao, Chen Chen, Tianrong Chen, and Jiatao Gu. One layer is enough: Adapting pretrained visual encoders for image generation. *arXiv preprint arXiv:2512.07829*, 2025.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023.

Jiahao Guo, Sinan Du, Jingfeng Yao, Wenyu Liu, Bo Li, Haoxiang Cao, Kun Gai, Chun Yuan, Kai Wu, and Xinggang Wang. Visual generation tuning. arXiv preprint arXiv:2511.23469, 2025. URL https://arxiv.org/abs/2511.23469.

Minghui Guo, Shengqiong Wu, and Hao Fei. V-RAE: Rethinking video latent spaces for generation. *arXiv preprint arXiv:2608.13556*, 2026.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.

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

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. ELLA: Equip diffusion models with LLM for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

ITU-T. Wideband extension to recommendation P.862 for the assessment of wideband telephone networks and speech codecs. Recommendation P.862.2, International Telecommunication Union, November 2007. URL https://www.itu.int/rec/T-REC-P.862.2-200711-W/en.

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

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*, 2025a.

Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. *arXiv preprint arXiv:2504.16064*, 2025b.

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.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, pp. 3927–3936, 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.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for end-to-end tuning with latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18262–18272, 2025a.

Xingjian Leng, Jaskirat Singh, Ryan Murdock, Ethan Smith, Rebecca Li, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Family of End-to-End Tuned VAEs for Supercharging T2I Diffusion Transformers. `https://end2end-diffusion.github.io/repa-e-t2i/`, 2025b.

Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Tiffany Ling, Xide Xia, Pengchuan Zhang, Graham Neubig, and Deva Ramanan. GenAI-Bench: Evaluating and improving compositional text-to-visual generation. *arXiv preprint arXiv:2406.13743*, 2024a.

Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. ImageFolder: Autoregressive image generation with folded tokens. In *International Conference on Learning Representations*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. UniTok: A unified tokenizer for visual generation and understanding. In *Advances in Neural Information Processing Systems*, 2025a.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research*, 2025b.

Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, et al. JanusFlow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7739–7751, 2025c.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802, 2017.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*, 2024.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. In *International Conference on Machine Learning*, pp. 7958–7968, 2021.

Zhikang Niu, Shujie Hu, Jeongsoo Choi, Yushen Chen, Peining Chen, Pengcheng Zhu, Yunting Yang, Bowen Zhang, Jian Zhao, Chunhui Wang, et al. Semantic-vae: Semantic-alignment latent representation for better speech synthesis. *arXiv preprint arXiv:2509.22167*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

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.

Pablo Pernias, Dominic Rampas, Mats Leon Richter, Christopher Pal, and Marc Aubreville. Würstchen: An efficient architecture for large-scale text-to-image diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K. Du, Zehuan Yuan, and Xinglong Wu. TokenFlow: Unified image tokenizer for multimodal understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2545–2555, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763, 2021.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pp. 28492–28518. PMLR, 2023.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pp. 749–752. IEEE, 2001.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 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.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

Fengyuan Shi, Zhuoyan Luo, Yixiao Ge, Yujiu Yang, Ying Shan, and Limin Wang. Scalable image tokenization with index backpropagation quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16037–16046, 2025.

Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? In *The Fourteenth International Conference on Learning Representations*, 2026a.

Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved baselines with representation autoencoders. *arXiv preprint arXiv:2605.18324*, 2026b.

Hubert Siuzdak. Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 25719–25733, 2024.

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.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Transactions on audio, speech, and language processing*, 19(7):2125–2136, 2011.

Bao Tang, Jiahao Guo, Haoxiang Cao, Wenyu Liu, Changqian Yu, Kun Gai, and Xinggang Wang. StableVQ: Practical guidelines for stable vector-quantized tokenizer training. arXiv preprint arXiv:2609.26774, 2026. URL https://arxiv.org/abs/2609.26774.

Bingda Tang, Boyang Zheng, Sayak Paul, and Saining Xie. Exploring the deep fusion of large language models and diffusion transformers for text-to-image synthesis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28586–28595. IEEE, 2025.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, and Weilong Yang. Regularizing generative adversarial networks under limited data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7921–7931, 2021.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Junke Wang, Yi Jiang, Zehuan Yuan, Binyue Peng, Zuxuan Wu, and Yu-Gang Jiang. OmniTokenizer: A joint image-video tokenizer for visual generation. *arXiv preprint arXiv:2406.09399*, 2024.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16133–16142. Ieee, 2023.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3D representation for consistent world modeling. *arXiv preprint arXiv:2507.07982*, 2025b.

Junfeng Wu, Dongliang Luo, Weizhi Zhao, Zhihao Xie, Yuanhao Wang, Junyi Li, Xudong Xie, Yuliang Liu, and Xiang Bai. TokBench: Evaluating your visual tokenizer before visual generation. *arXiv preprint arXiv:2505.18142*, 2025c.

Jiawei Yang, Tianhong Li, Lijie Fan, Yonglong Tian, and Yue Wang. Latent denoising makes good tokenizers. *arXiv preprint arXiv:2507.15856*, 2025.

Jingfeng Yao, Wang Cheng, Wenyu Liu, and Xinggang Wang. FasterDiT: Towards faster diffusion transformers training without architecture modification. *arXiv preprint arXiv:2410.10356*, 2024.

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

Jingfeng Yao, Yuda Song, Yucong Zhou, and Xinggang Wang. Towards scalable pre-training of visual tokenizers for generation. In *European Conference on Computer Vision*, pp. 661–678. Springer, 2026.

Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion—tokenizer is key to visual generation. In *International Conference on Learning Representations*, 2024a.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 128940–128966, 2024b.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019.

Letian Zhang, Sucheng Ren, Yanqing Liu, Xianhang Li, Zeyu Wang, Yuyin Zhou, Huaxiu Yao, Zeyu Zheng, Weili Nie, Guilin Liu, Zhiding Yu, and Cihang Xie. OpenVision 3: A family of unified visual encoder for both understanding and generation. *arXiv preprint arXiv:2601.15369*, 2026a.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. VideoREPA: Learning physics for video generation through relational alignment with foundation models. *arXiv preprint arXiv:2505.23656*, 2025.

Zekai Zhang, Deqing Li, Kuan Cao, Yujia Wu, Chenfei Wu, Yu Wu, Liang Peng, Hao Meng, Jiahao Li, Jie Zhang, et al. Qwen-Image-VAE-2.0 technical report. *arXiv preprint arXiv:2605.13565*, 2026b.

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient GAN training. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *arXiv preprint arXiv:2306.09305*, 2023.

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

Yongxin Zhu, Bocheng Li, Yifei Xin, Zhihua Xia, and Linli Xu. Addressing representation collapse in vector quantized models with one linear layer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22968–22977, 2025.

Supplementary Material of NA-VAE

CONTENTS PAGE

A Latent PCA Visualization and Analysis 18

B Latent Frequency-Domain Analysis 21

C Detailed Experimental Setups and Results 25

C.1 Training Details 25
C.2 Evaluation Details 26
C.3 Description of Pretrained Visual Encoders 27
C.4 Architecture Ablation of NA-VAE 28
C.5 Ablation of the Noise Schedule 31
C.6 Ablation of Model Size 34
C.7 Ablation of the Representation Encoder 35
C.8 Component Analysis 36
C.9 Text-to-Image Generation 37
C.10 Video Generation 38
C.11 Audio Generation 40
C.12 Reconstruction Performance 43

D Extended Related Work 44

E Qualitative Results 46

A LATENT PCA VISUALIZATION AND ANALYSIS

We describe the exact procedure used to produce Figs. 1 and 3, and extend the visualizations to the noise-schedule and alignment-head ablations across all four tokenizer families.

Images and latent extraction. We fit PCA on a fixed set of six ImageNet images shared across all tokenizers and ablations; only a subset is displayed in each figure. Each image is read at 256×256 resolution and mapped to $[-1, 1]$. For each tokenizer, we extract the posterior mode in evaluation mode, with gradients disabled; encoding uses BF16 autocasting and the resulting latents are converted to FP32 for PCA. The latent grids are 32×32 for SD-VAE and FLUX.1, and 16×16 for IN-VAE and VA-VAE. We visualize the *clean encoder latents*, not the alignment-head outputs or teacher features: no noise is injected during visualization, even for tokenizers trained with noisy alignment.

PCA and color mapping. For each tokenizer independently, we standardize every latent channel using its mean and population standard deviation over all six images and spatial positions. These empirical statistics are distinct from the running normalization statistics used during training. We flatten the normalized latents into a matrix of spatial tokens, center its columns, and compute its singular value decomposition. The scores along the leading three principal directions are independently min–max normalized over the same token pool and mapped to RGB. The following NumPy implementation reproduces the original projection:

```
import numpy as np


def pca_rgb(latents):
    batch, channels, height, width = latents.shape
    channel_mean = latents.mean(axis=(0, 2, 3), keepdims=True)
    channel_std = latents.std(axis=(0, 2, 3), keepdims=True)
    normalized = (latents - channel_mean) / (channel_std + 1e-6)
    tokens = normalized.transpose(0, 2, 3, 1).reshape(-1, channels)
    tokens -= tokens.mean(axis=0, keepdims=True)
    _, _, directions = np.linalg.svd(tokens, full_matrices=False)
    scores = tokens @ directions[:3].T
    lower, upper = scores.min(axis=0), scores.max(axis=0)
    colors = (scores - lower) / (upper - lower + 1e-8)
    return colors.reshape(batch, height, width, 3)
```

The resulting RGB maps are reshaped to the native latent grid, quantized to 8-bit values, and enlarged with nearest-neighbor interpolation; no smoothing or additional color adjustment is applied. Crucially, *one PCA basis and one color mapping are shared by all six images of a tokenizer*, rather than fitted separately per image. After fitting, we display four images in Figs. 1 and 3, and three (fox, snake, and panda) in the appendix panels. Since different tokenizers have independent PCA bases, hues are arbitrary across tokenizers; only spatial organization should be compared.

Rendering the maps. Given a NumPy array `latents` of shape $(6, C, H, W)$ containing the six posterior-mode encodings from one tokenizer, the following snippet uses the function above to render a row of PCA maps. Each map is enlarged to 256×256 pixels without interpolation across latent cells. Repeating this procedure for each tokenizer yields the panels for comparison.

```
from PIL import Image

rgb_maps = pca_rgb(latents)
tiles = (np.clip(rgb_maps, 0, 1) * 255).round().astype(np.uint8)
row = Image.fromarray(np.concatenate(list(tiles), axis=1))
row.resize(
    (256 * len(tiles), 256), Image.Resampling.NEAREST
).save("latent_pca.png")
```

Uniform noise range. Fig. 10 extends the clean/noisy comparison in Fig. 1 to $t \sim \mathrm{U}(0, t_{\max})$, with $t_{\max} \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$, together with clean alignment ($t = 0$). All ablation checkpoints are evaluated at 100k training steps, with the SiT-8 alignment head fixed; VAE and E2E-VAE denote the original and released end-to-end-tuned tokenizers, respectively. We retain SiT-8 for this controlled sweep; the adopted NA-VAE configuration uses SiT-1. The FLUX.1 runs use no adversarial loss, whereas the other three families do, so training choices should be compared *within* each family.

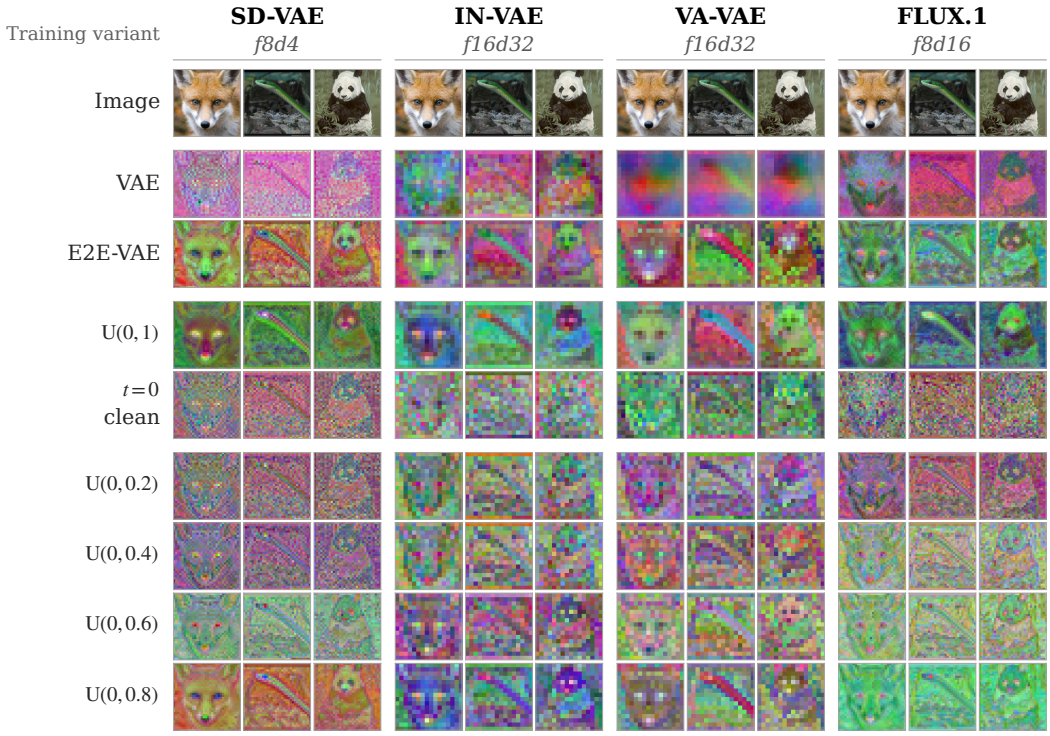


Fig. 10: **Latent PCA under different uniform noise ranges.** Each block corresponds to a tokenizer family and shows the same three images. The first three rows provide the input images, original VAE, and E2E-VAE released by Leng et al. (2025a). The remaining rows compare $t \sim \mathrm{U}(0, 1)$, clean alignment ($t = 0$), and restricted uniform ranges, all with the SiT-8 head fixed. All maps show clean posterior-mode latents using the protocol of Fig. 1; colors are not comparable across rows.

Clean alignment and restricted low-noise ranges often retain substantial fine-scale variation. Broader ranges yield more coherent object-level patterns in these examples, although the response varies across tokenizer families. These qualitative changes complement the downstream ablations rather than establishing a monotonic relationship between noise magnitude and generation quality.

Noise sampling distribution. Fig. 11 compares uniform sampling with the three logit-normal schedules used in the main-text ablation. Following the figure notation, $\mathrm{lognorm}(\mu, s)$ denotes *logit-normal* sampling of t, implemented as

$$t = \mathrm{sigmoid}(\mu + s\epsilon), \qquad \epsilon \sim \mathcal{N}(0, 1), \qquad \mu \in \{-1, 0, +1\}, \quad s = 1.$$

Thus the three settings have unit logit standard deviation and favor lower, intermediate, or higher noise levels, respectively; they do not sample an unbounded log-normal timestep. The alignment head and tokenizer training budget remain unchanged.

The high-noise-biased schedule emphasizes broader spatial regions, while the low-noise-biased schedule retains more local variation in these examples. Neither smoother maps nor closer color agreement with E2E-VAE imply better generation: PCA is a qualitative diagnostic, and the quantitative comparison is given in Fig. 4.

Alignment-head architecture. Fig. 12 expands Fig. 3 to seven head ablations, together with the SiT-8 baseline and released references.

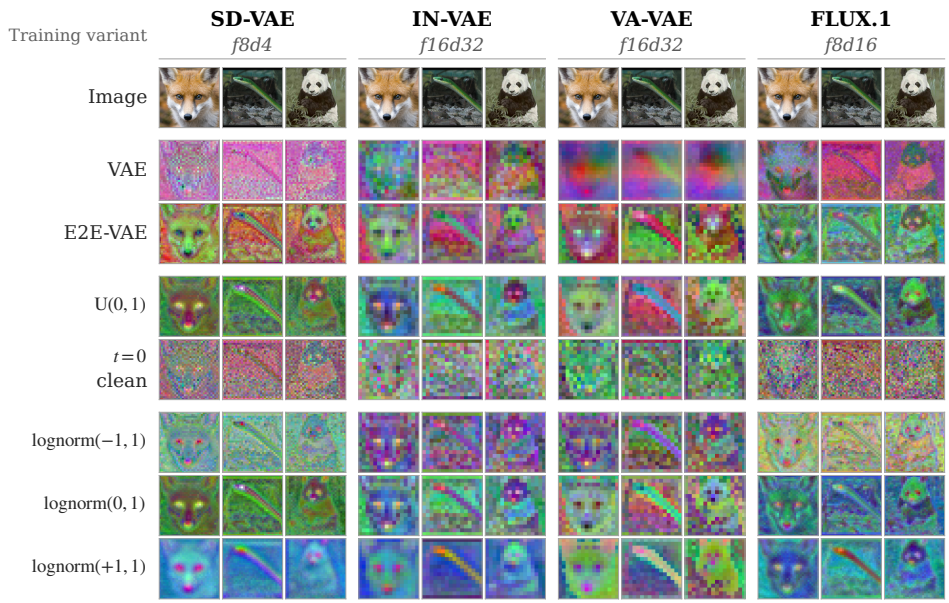


Fig. 11: **Latent PCA under different timestep sampling distributions.** Using the same images and references as Fig. 10, we compare $\mathrm{U}(0,1)$, clean alignment, and logit-normal sampling with logit means -1, 0, and $+1$ and standard deviation 1. All ablation rows use SiT-8 heads and 100k-step checkpoints. The training distribution changes, but the visualization always uses clean encoder latents and an independently fitted PCA basis for each tokenizer.

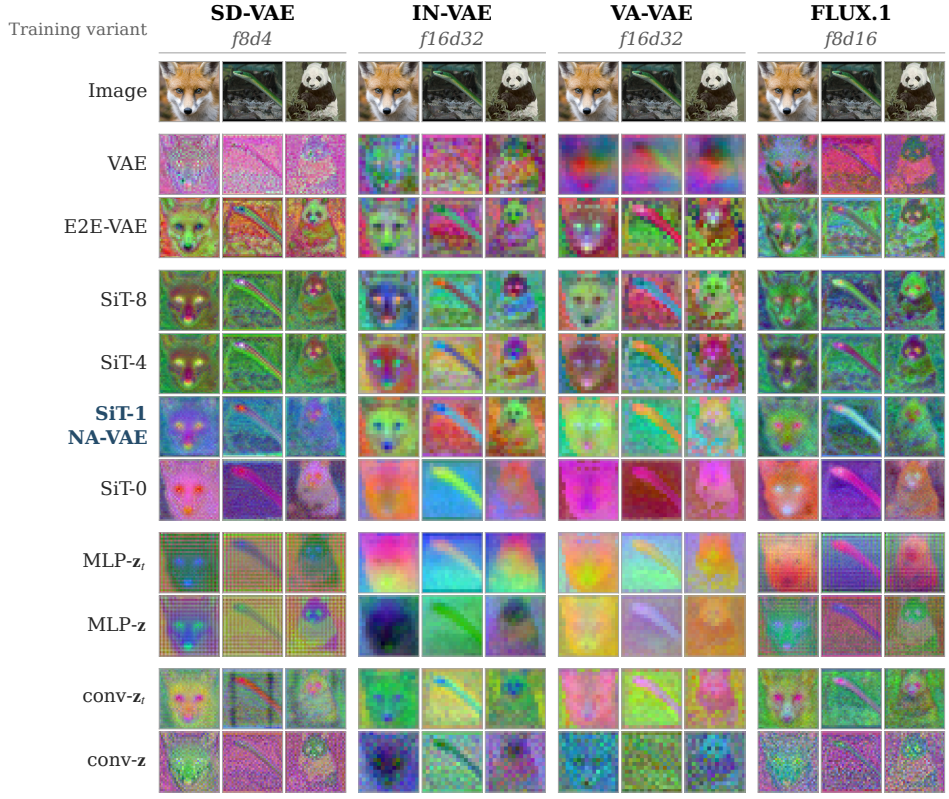


Fig. 12: **Latent PCA across alignment-head architectures.** SiT-8/4/1 denote the number of transformer blocks, with SiT-1 adopted for NA-VAE. SiT-0 retains both positional and timestep embeddings but no transformer blocks. MLP-$\boldsymbol{z}_t$ and MLP-$\boldsymbol{z}$ use pointwise heads with noisy and clean training inputs, respectively. Likewise, conv-$\boldsymbol{z}_t$ uses noisy inputs and conv-$\boldsymbol{z}$ uses clean inputs. Noisy variants use $t \sim \mathrm{U}(0,1)$; all ablation variants are evaluated at 100k steps. Pointwise heads can exhibit grid-like patterns, particularly in the higher-resolution SD-VAE and FLUX.1 grids, whereas spatially structured heads yield more coherent regions.

B LATENT FREQUENCY-DOMAIN ANALYSIS

We detail the frequency-domain diagnostic in Fig. 2 and extend it to the ablations in §A. The analysis measures how latent energy is distributed across spatial frequencies and when each band falls below the noise power of the linear interpolant.

Evaluation protocol. We select 256 ImageNet training images at evenly spaced indices in the sorted image list and use the same images for every tokenizer. We use 256×256 RGB inputs mapped to $[-1, 1]$. As in §A, we extract clean posterior-mode latents in evaluation mode with BF16 autocasting and convert them to FP32. The analysis covers SD-VAE and FLUX.1 (32×32 grids with 4 and 16 channels), and IN-VAE and VA-VAE (16×16 grids with 32 channels). Our controlled variants use 100k-step tokenizer checkpoints; the original and released E2E-VAEs serve as references. The noisy SiT-8 branch in Fig. 2 is a controlled configuration, distinct from the SiT-1 head adopted for NA-VAE.

Normalized power spectrum. For each tokenizer, we standardize each channel over the 256 images and all spatial positions using the empirical mean and population standard deviation. These statistics are recomputed for this diagnostic, not taken from the training-time normalization layer. For a standardized latent $\widetilde{\boldsymbol{z}}$ with N images, C channels, and spatial size $H \times W$, we use the orthonormal Fourier transform $F_{n,c} = \mathcal{F}(\widetilde{\boldsymbol{z}}_{n,c})/\sqrt{HW}$. Under this scaling, independent unit-variance white noise has expected power one per Fourier mode.

We group the centered Fourier grid into integer-radius rings by rounding each mode's Euclidean distance from the origin. For ring $\mathcal{B}_b$ containing n_b modes, its mean per-mode power and share of the total non-DC energy are

$$P_b = \frac{1}{NCn_b} \sum_{n,c} \sum_{(u,v)\in\mathcal{B}_b} |F_{n,c}(u,v)|^2, \qquad E_b = \frac{n_b P_b}{\sum_{b'>0} n_{b'} P_{b'}}. \tag{5}$$

The zero-frequency (DC) mode is excluded, but all other modes, including the corners of the two-dimensional spectrum, are retained. The factor n_b is essential: an outer ring generally contains more modes than an inner one. The 32×32 and 16×16 grids yield 23 and 11 rings, respectively. No PCA, spatial resampling, or windowing is applied.

Computing the band statistics. The following function takes an array of shape (N, C, H, W) and returns the normalized radial frequencies, per-mode powers P_b, and energy shares E_b. All spectral calculations use FP64.

```
import numpy as np


def radial_spectrum(latents):
    values = np.asarray(latents, dtype=np.float64)
    height, width = values.shape[-2:]
    channel_mean = values.mean(axis=(0, 2, 3), keepdims=True)
    channel_std = values.std(axis=(0, 2, 3), keepdims=True)
    values = (values - channel_mean) / (channel_std + 1e-12)
    spectrum = np.fft.fft2(values, axes=(-2, -1))
    spectrum = np.fft.fftshift(spectrum, axes=(-2, -1))
    mode_power = (np.abs(spectrum / np.sqrt(height * width)) ** 2)
    mode_power = mode_power.mean(axis=(0, 1))
    rows, columns = np.indices((height, width))
    radius = np.rint(np.hypot(rows - height // 2,
                              columns - width // 2)).astype(int)
    bands = np.arange(1, radius.max() + 1)
    power = np.array([mode_power[radius == band].mean() for band in bands])
    counts = np.array([(radius == band).sum() for band in bands])
    energy = counts * power
    energy /= energy.sum()
    return bands / (height // 2), power, energy
```

The frequency coordinate is $f/f_{\text{nyq}} = b/(H/2)$ for these square grids. Values above one correspond to corner rings, not discarded frequencies. Because the tokenizer downsampling factors differ, identical normalized frequencies need not correspond to identical wavelengths in image space.

Surviving energy under noise. For $\boldsymbol{z}_t = (1-t)\widetilde{\boldsymbol{z}} + t\boldsymbol{\epsilon}$ with independent $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, the band-averaged signal-to-noise ratio and its unit-SNR crossing are

$$\mathrm{SNR}_b(t) = \frac{(1-t)^2 P_b}{t^2}, \qquad t_b^* = \frac{\sqrt{P_b}}{1+\sqrt{P_b}}. \tag{6}$$

We count a band's *original* non-DC energy while its mean SNR exceeds one:

$$R(t) = \sum_b E_b \mathbf{1}[t < t_b^*], \qquad \bar{R} = \int_0^1 R(t)\, dt = \sum_b E_b t_b^*. \tag{7}$$

Thus $\bar{R}$ is the area under the curve, equivalently its mean under $t \sim \mathrm{U}(0,1)$. Noise need not be sampled to compute either quantity.

```
def band_survival(power, energy):
    power, energy = np.asarray(power), np.asarray(energy)
    root_power = np.sqrt(np.maximum(power, 0.0))
    thresholds = root_power / (1.0 + root_power)
    times = np.unique(np.concatenate(([0.0], thresholds, [1.0])))
    layers = energy[:, None] * (times[None, :] < thresholds[:, None])
    average = float(np.sum(energy * thresholds))
    return times, layers, average
```

Rendering the curves. The code below stacks rings from low to high frequency, assigning frequencies above one the upper-end color. We render the curves as step functions, with each band disappearing at its analytical threshold t_b^*.

```
import matplotlib.pyplot as plt

frequency, power, energy = radial_spectrum(latents)
times, layers, average = band_survival(power, energy)
colors = plt.cm.viridis(0.08 + 0.88 * np.clip(frequency, 0, 1))
figure, axis = plt.subplots(figsize=(3, 2))
axis.stackplot(times, layers, colors=colors, step="post")
axis.step(times, layers.sum(axis=0), where="post", color="black")
axis.set(xlim=(0, 1), ylim=(0, 1), xlabel="t", ylabel="R(t)")
figure.savefig("band_survival.pdf", bbox_inches="tight")
```

Expanded ablations. We evaluate 72 checkpoints: two released references and 16 adapted variants for each of the four tokenizer families. The three figures below mirror the uniform-range, sampling-distribution, and head-architecture groups in §A. All measurements use the same 256 images, GPU BF16 encoding, and FP64 spectral analysis. Repeating the 18 configurations shared with the original frequency analysis changes $\bar{R}$ by less than 10^{-5}. The sampling sweeps fix the head to SiT-8; the architecture sweep includes SiT-1, the adopted NA-VAE configuration. The FLUX.1 adaptation runs do not use an adversarial loss, unlike the other three families, so controlled comparisons are made within each family.

Interpretation and limitations. This binary, band-averaged SNR diagnostic does not measure a denoiser's accuracy. Larger $\bar{R}$ is not necessarily better: the released VA-VAE has 0.63, versus 0.54 for its E2E and noisy SiT-8 counterparts. Radial averaging hides directional structure, and unwindowed spectra remain subject to boundary leakage.

Uniform noise range. Fig. 13 shows that low-noise-only training retains a larger high-frequency energy share. Extending the range to $t_{\max} = 0.8$ or 1 produces a broader progression of band dropout times, consistent with the PCA maps in Fig. 10.

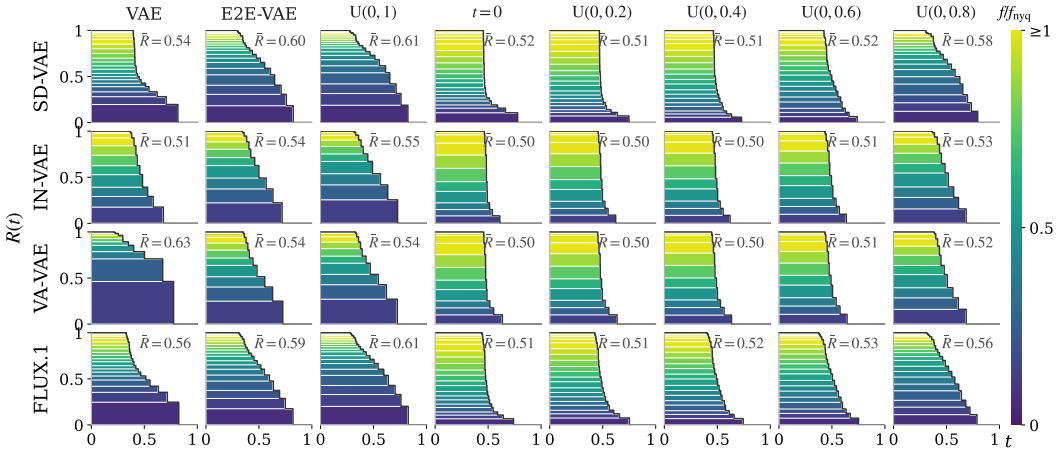


Fig. 13: **Frequency analysis of the uniform noise-range ablation.** Rows correspond to tokenizer families. Following the released VAE and E2E-VAE references, columns compare $\mathrm{U}(0, 1)$, clean alignment ($t = 0$), and restricted ranges $\mathrm{U}(0, t_{\max})$, all with SiT-8 heads. Each color denotes a radial band; the stacked height is $R(t)$ and its area is $\bar{R}$. All adapted checkpoints are evaluated at 100k steps.

Noise sampling distribution. Shifting the logit mean from -1 to $+1$ consistently concentrates more energy at low frequencies and increases $\bar{R}$ (Fig. 14). This ordering describes noise tolerance, not downstream generation quality; the uniform schedule remains competitive in the main-text ablations.

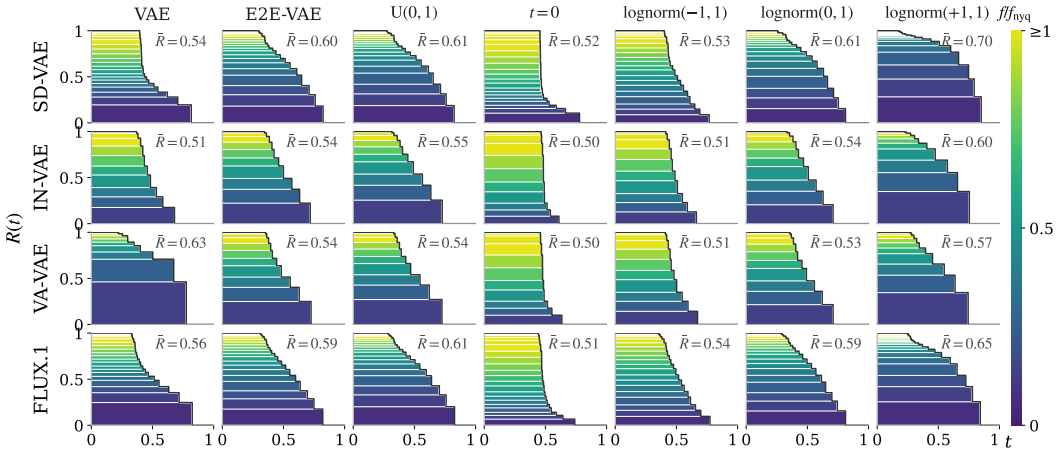


Fig. 14: **Frequency analysis of the timestep sampling distribution.** Under the same SiT-8 head and training budget, we compare uniform sampling, clean alignment, and three logit-normal schedules. As in §A, $\mathrm{lognorm}(\mu, 1)$ denotes $t = \mathrm{sigmoid}(\mu + \epsilon)$ with $\epsilon \sim \mathcal{N}(0, 1)$. All curves use the same diagnostic t axis, irrespective of the training distribution, and $\bar{R}$ always denotes a uniform-t average.

Alignment-head architecture. Fig. 15 separates transformer depth from the pointwise/convolutional head comparison. A single SiT block preserves the broadly distributed band-dropout profile without requiring a deep head. For convolutional heads, noisy alignment consistently reduces high-frequency energy relative to clean alignment: the share above $f/f_{\mathrm{nyq}} = 0.5$ falls from $0.56/0.54/0.53/0.63$ to $0.26/0.27/0.29/0.27$ for SD-VAE, IN-VAE, VA-VAE, and FLUX.1, respectively. Pointwise heads behave less uniformly; their near-Nyquist concentration on SD-VAE illustrates why the band-resolved curves matter beyond $\bar{R}$ alone.

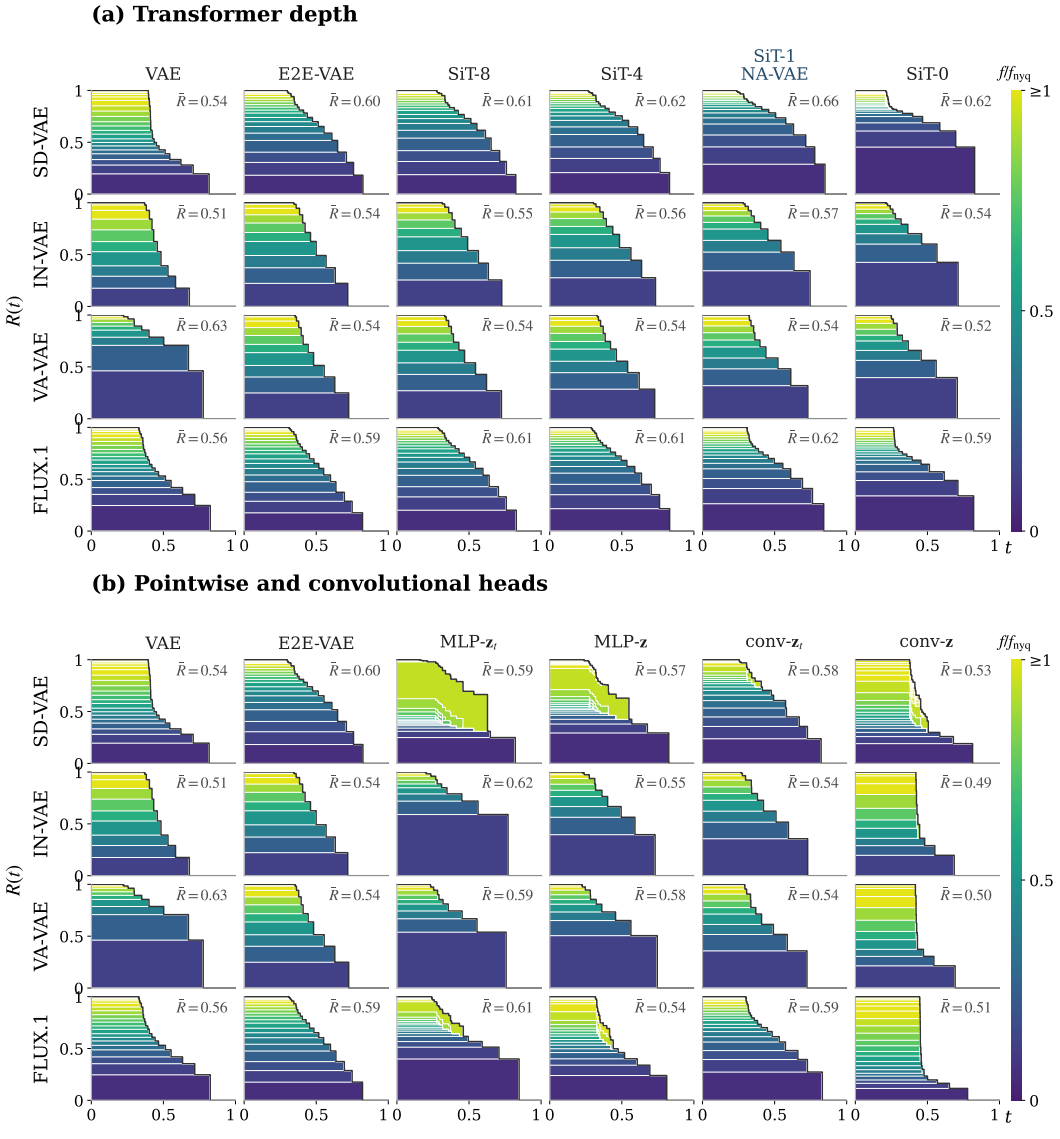


Fig. 15: **Frequency analysis of alignment-head architectures.** (a) SiT-8/4/1/0 controls, with SiT-1 adopted for NA-VAE. SiT-0 retains positional and timestep embeddings but no transformer blocks. (b) Pointwise MLP and convolutional heads trained on noisy ($\boldsymbol{z}_t$) or clean ($\boldsymbol{z}$) inputs; neither uses timestep conditioning. The released VAE and E2E-VAE references are repeated in both blocks. Noisy variants use $t \sim \mathrm{U}(0, 1)$, and all adapted checkpoints are at 100k steps. As throughout this section, the analyzed encoder latents are clean.

C Detailed Experimental Setups and Results

C.1 Training Details

We describe the ImageNet 256×256 training setup in two separate stages: NA-VAE training and downstream class-conditional SiT training with the tokenizer frozen. Tab. 3 summarizes the default configuration; training budgets are listed separately in Tab. 4.

Tab. 3: **Default training configuration on ImageNet** 256×256**.** The NA-VAE column describes the adopted single-block alignment head; ablations change the specified component. All batch sizes are global.

	NA-VAE training	**Downstream SiT training**
Architecture and representation alignment		
Transformer blocks	1 (alignment head)	28 (SiT-XL)
Hidden dimension / attention heads	1,152 / 16	1,152 / 16
Alignment readout layer	1	8
Query–key normalization	Enabled	Disabled
Projection MLP	3 linear layers, hidden width 2,048, SiLU activations	
Frozen visual encoder	DINOv2-B	DINOv2-B
Alignment gradient coefficient	1.5 (encoder), 0.5 (head)	0.5
Training objectives		
Noise-level sampling	$t \sim \mathrm{U}(0,1)$	$t \sim \mathrm{U}(0,1)$
Interpolant coefficients (α_t, σ_t)	$(1-t, t)$	$(1-t, t)$
Velocity-prediction loss weight	—	1.0
L_1 / LPIPS (VGG) weights	1.0 / 1.0	—
KL regularization weight	10^{-6}	—
Adversarial loss weight	0.1 (fixed)	—
Class-label dropout probability	— (no labels)	0.1
Optimization		
Optimizer	AdamW	AdamW
Global batch size	256	256
Learning rate	10^{-4}	10^{-4}
Learning-rate schedule / warmup	Constant / none	Constant / none
Adam (β_1, β_2) / ϵ	$(0.9, 0.999)$ / 10^{-8}	$(0.9, 0.999)$ / 10^{-8}
Weight decay	0	0
Gradient clipping (max. norm)	1.0	1.0
Mixed precision	FP16	FP16
EMA decay	—	0.9999
Random seed	42	0

Data and latent geometry. We use the ImageNet training split with the deterministic resize-and-center-crop preprocessing of SiT/REPA (Ma et al., 2024; Yu et al., 2025), without random augmentation. Tokenizer inputs are mapped to $[-1, 1]$. SD-VAE produces $32 \times 32 \times 4$ latents, whereas IN-VAE and VA-VAE produce $16 \times 16 \times 32$ latents. We use latent patch sizes of 2 and 1, respectively, for both the alignment head and the downstream transformer, yielding the same 16×16 token grid. The corresponding downstream models are SiT-XL/2 and SiT-XL/1.

NA-VAE training. We initialize each autoencoder and its PatchGAN discriminator from the corresponding pretrained checkpoints, and jointly optimize the encoder, decoder, and randomly initialized alignment head without class labels or a diffusion loss. The decoder reconstructs from a clean posterior sample; only the alignment branch adds noise after channel-wise latent normalization. The single-block head uses timestep conditioning and fixed 2D sinusoidal positional embeddings. Its projected tokens are aligned by negative cosine similarity to the final patch features of a frozen DINOv2-B encoder, evaluated on 224×224 images with ImageNet normalization. The alignment term contributes gradients with coefficients 1.5 to the VAE encoder and 0.5 to the head. Reconstruction combines mean L_1, VGG-LPIPS, KL regularization, and a fixed-weight adversarial term. The discriminator uses a hinge loss, 64 base channels, and three downsampling layers; it is updated from the first step with the same optimizer settings and clipping threshold as the autoencoder.

Downstream SiT training. We discard the alignment head and freeze the tokenizer, then train a new SiT with REPA using the frozen-tokenizer protocol of REPA-E (Leng et al., 2025a). We

cache the posterior means and standard deviations and sample latents during training. Channel-wise normalization uses the tokenizer's learned running statistics, which remain fixed throughout downstream training. The objective combines velocity prediction with REPA alignment at layer 8; only this stage uses ImageNet class labels. Sampling and evaluation settings are deferred to §C.2.

Training budgets. Unless otherwise specified, tokenizer ablations use 20 epochs, while the NA-VAE tokenizers in Tab. 2 use 80 epochs. These correspond to 100k and 400k total optimization steps, respectively, at batch size 256; epoch counts are approximate. The *Epochs* column of Tab. 2 instead denotes *downstream SiT training*, with no further tokenizer updates.

Tab. 4: **Tokenizer and downstream training budgets.** Each entry reports total steps and approximate ImageNet epochs, not additional steps after resuming. The longer downstream run reuses the same frozen 80-epoch tokenizer.

Experiment	NA-VAE training	Downstream SiT training
Ablations (unless specified)	100k / 20 epochs	400k / 80 epochs
Tab. 2: SD-VAE and VA-VAE	400k / 80 epochs	400k / 80 epochs
Tab. 2: SD-VAE, longer run	400k / 80 epochs	2.4M / 480 epochs

C.2 Evaluation Details

ImageNet generation protocol. We follow the ADM evaluation protocol (Dhariwal & Nichol, 2021) adopted by REPA-E (Leng et al., 2025a). For each checkpoint, we generate 50,000 images with exactly 50 samples per ImageNet class, following the class-balanced protocol used in REPA-E's system-level comparison. Sampling uses the SiT EMA weights, a linear interpolant, and the 250-step Euler–Maruyama SDE sampler of SiT/REPA (Ma et al., 2024; Yu et al., 2025), starting from standard Gaussian noise. We use a global random seed of 0 and enable TF32 for generation. Generated latents are denormalized using the frozen channel-wise statistics and decoded by the corresponding frozen tokenizer; the resulting images are clipped and converted to 8-bit RGB for evaluation.

Classifier-free guidance. Unless explicitly reported as guided, all results use a guidance scale of 1.0 (no CFG). Guided results apply CFG on an interval $t \in [0, t_{\max}]$, where sampling proceeds from $t = 1$ (noise) to $t = 0$ (data); outside the interval, sampling remains class-conditional without CFG. For the guided results in Tab. 2 we use a scale of 2.1 with $t_{\max} = 0.825$ for NA-VAE$_{\text{VA-VAE}}$, and scales of 2.4 (80 epochs) and 2.2 (480 epochs) with $t_{\max} = 0.70$ for NA-VAE$_{\text{SD-VAE}}$. Each setting is the search optimum of a per-checkpoint grid over the scale and the interval. The shorter interval of the SD-VAE arm reflects its architecture rather than our method: that arm decodes *f8d4* latents with SiT-XL/2, whereas the VA-VAE arm decodes *f16d32* latents with SiT-XL/1, and the published settings of REPA (Yu et al., 2025) and REPA-E (Leng et al., 2025a) track the same split. The sample count, sampler, and evaluation protocol are otherwise unchanged.

Reference data and metrics. All five generation metrics are computed on the same 50,000 generated images using ADM's Inception-v3 evaluator and resolution-matched ImageNet reference batches (256×256, or 512×512 for higher-resolution experiments). FID and sFID use the supplied reference means and covariances; precision and recall use features of the reference images contained in each batch. Following the metric overview in REPA (Yu et al., 2025), we summarize their roles:

- **Generation FID (gFID↓)** (Heusel et al., 2017) compares Gaussian approximations to the real and generated distributions in the 2,048-dimensional pooled Inception feature space. It measures distributional agreement through differences in feature means and covariances.
- **Spatial FID (sFID↓)** (Nash et al., 2021) applies the same distance to intermediate spatial Inception features instead of globally pooled features, making it sensitive to spatial structure.
- **Inception score (IS↑)** (Salimans et al., 2016) measures the combination of confident predictions for individual generated images and diversity across the generated set:

$$\text{IS} = \exp(\mathbb{E}_x \, D_{\text{KL}}[p(y \mid x) \,\|\, p(y)]) \,, \qquad p(y) = \mathbb{E}_x \, p(y \mid x),$$

  where $p(y \mid x)$ is the Inception class distribution. We follow ADM in averaging the scores over ten splits of 5,000 images. IS uses generated images only, without comparison to real images or their labels.

- **Precision and recall (Prec./Rec.↑)** (Kynkäänniemi et al., 2019) estimate fidelity and coverage using nearest-neighbor neighborhoods in Inception feature space, with $k = 3$ as in ADM. Precision is the fraction of generated features inside the estimated real-data manifold; recall is the fraction of real features inside the estimated generated-data manifold.

Reconstruction evaluation. Reconstruction FID (rFID) instead compares the feature distributions of original images and their autoencoder reconstructions; lower values indicate better distributional fidelity. We evaluate reconstruction on all 50,000 ImageNet validation images, separately from the generation protocol above. The reconstruction preprocessing and additional fidelity metrics are detailed in §C.12.

C.3 Description of Pretrained Visual Encoders

Fig. 5b compares four frozen vision foundation models (VFMs) as alignment targets for NA-VAE. Following the overview in REPA (Yu et al., 2025), we summarize their pretraining objectives and the variants used in our experiments.

- **DINOv2-B and DINOv2-L** (Oquab et al., 2024) learn visual representations through self-distillation with image-level and masked-patch objectives on the curated LVD-142M dataset. We use the ViT-B/14 and ViT-L/14 variants to examine the effect of encoder capacity within the same representation family.
- **CLIP-L** (Radford et al., 2021) learns visual representations through contrastive image–text pretraining on 400 million image–text pairs. We use its ViT-L/14 vision encoder and extract spatial features, without the text encoder or the global image–text projection.
- **I-JEPA-H** (Assran et al., 2023) learns by predicting the representations of target image blocks from visible context, with targets supplied by a momentum encoder rather than raw pixels. We use the ViT-H/14 context encoder pretrained on ImageNet-1K without labels; the pretraining predictor is not used for alignment.

Tab. 5: **Pretrained visual encoders in Fig. 5b.** All encoders receive 224×224 images during NA-VAE training.

Encoder	Backbone	Pretraining objective	Pretraining data
DINOv2-B	ViT-B/14	Self-distillation	LVD-142M
DINOv2-L	ViT-L/14	Self-distillation	LVD-142M
CLIP-L	ViT-L/14	Image–text contrastive	WIT-400M
I-JEPA-H	ViT-H/14	Masked feature prediction	ImageNet-1K

Feature extraction. All VFMs remain frozen and in evaluation mode throughout tokenizer training. We resize images to 224×224 and apply the encoder-specific normalization: ImageNet statistics for DINOv2 and I-JEPA, and CLIP statistics for CLIP. Resizing uses bilinear interpolation for the DINOv2-B baseline and bicubic interpolation for the other three variants. We retain the final-layer spatial tokens and discard the class token when present. With a patch size of 14, each encoder provides a 16×16 feature grid, matching the patchified SD-VAE latents without spatial interpolation. The alignment head projects each latent token to the corresponding feature dimension, and alignment uses negative cosine similarity as in §C.1. No labels or captions are supplied during NA-VAE training, including when using the language-pretrained CLIP encoder.

Scope of the encoder ablation. For our runs in Fig. 5b, the VFM is varied only during tokenizer training. Downstream SiT-L/2 training always uses DINOv2-B for REPA alignment at layer 8. This keeps downstream supervision fixed when comparing the resulting tokenizers; extended results are provided in §C.7.

C.4 Architecture Ablation of NA-VAE

We supplement the *Architecture of Alignment Head* study in §4.2. Tabs. 6–8 extend Tab. 1 and Fig. 4a to all five metrics at eight checkpoints. Tokenizers use 100k training steps (20 epochs), then remain frozen for 400k downstream steps (80 epochs), with DINOv2-B and no CFG (§C.1 and §C.2).

Head variants. SiT-d has d blocks; SiT-0 retains position/time embeddings. MLP and conv use three-layer pointwise/3×3 projectors without these embeddings; conv+t adds time conditioning. Suffixes $\boldsymbol{z}_t$ and $\boldsymbol{z}$ denote noisy ($t \sim \mathrm{U}(0,1)$) and clean alignment. Shading marks NA-VAE (SiT-1).

Tab. 6: **Detailed alignment-head ablations: SD-VAE (*f8d4*).** Columns are downstream SiT-XL/2 steps; bold marks the best displayed value per metric and checkpoint.

Alignment head	Metric	50k	100k	150k	200k	250k	300k	350k	400k
SiT-8-$\boldsymbol{z}_t$	gFID↓	**28.69**	10.93	7.85	6.66	5.92	5.49	5.13	4.92
	sFID↓	**6.84**	5.41	5.46	5.55	5.50	5.59	5.60	5.62
	IS↑	45.2	95.2	115.9	127.8	135.8	140.2	145.9	148.5
	Prec.↑	**0.622**	0.732	0.746	0.752	0.753	0.753	0.754	0.754
	Rec.↑	0.518	0.575	0.592	0.600	0.613	0.616	0.623	0.624
SiT-8-$\boldsymbol{z}$	gFID↓	44.45	16.13	11.61	9.78	8.73	8.07	7.63	7.41
	sFID↓	28.42	5.65	**4.75**	4.71	4.69	4.72	4.77	4.79
	IS↑	26.8	73.2	94.3	105.5	112.7	118.6	123.0	125.1
	Prec.↑	0.474	0.675	0.706	0.712	0.717	0.719	0.718	0.718
	Rec.↑	**0.549**	**0.597**	**0.608**	**0.619**	**0.625**	**0.631**	**0.645**	**0.642**
SiT-4-$\boldsymbol{z}_t$	gFID↓	29.42	**10.47**	**7.53**	**6.28**	**5.57**	**5.12**	**4.78**	4.53
	sFID↓	8.99	5.57	5.38	5.35	5.30	5.26	5.27	5.27
	IS↑	**45.7**	**97.9**	**117.6**	**131.0**	**138.1**	143.7	148.0	150.9
	Prec.↑	0.609	0.734	0.754	0.756	0.757	0.759	0.757	0.758
	Rec.↑	0.517	0.580	0.593	0.599	0.608	0.614	0.618	0.621
SiT-1-$\boldsymbol{z}_t$	gFID↓	32.91	11.44	7.96	6.56	5.69	5.17	4.83	**4.47**
NA-VAE	sFID↓	9.61	5.38	4.80	4.70	**4.60**	4.61	4.59	4.60
	IS↑	43.0	94.6	116.1	128.8	**138.1**	**144.1**	**149.2**	**154.9**
	Prec.↑	0.597	0.740	0.753	**0.759**	**0.762**	0.763	0.764	**0.764**
	Rec.↑	0.461	0.556	0.576	0.592	0.605	0.604	0.612	0.617
SiT-0-$\boldsymbol{z}_t$	gFID↓	52.36	19.41	13.16	10.53	8.84	7.79	7.18	6.64
	sFID↓	31.04	9.57	7.43	6.99	6.37	6.14	6.13	6.08
	IS↑	26.9	66.4	88.8	103.8	115.1	123.7	129.4	134.5
	Prec.↑	0.439	0.673	0.709	0.724	0.732	0.736	0.742	0.745
	Rec.↑	0.407	0.553	0.570	0.580	0.592	0.610	0.603	0.608
MLP-$\boldsymbol{z}_t$	gFID↓	55.20	22.90	16.04	12.98	10.92	9.73	8.81	8.18
	sFID↓	11.59	7.41	6.84	6.68	6.44	6.30	6.20	6.14
	IS↑	23.5	59.3	79.5	93.6	104.6	112.5	118.9	123.3
	Prec.↑	0.459	0.660	0.694	0.711	0.720	0.725	0.728	0.733
	Rec.↑	0.411	0.547	0.570	0.576	0.586	0.591	0.591	0.605
MLP-$\boldsymbol{z}$	gFID↓	37.69	17.23	12.56	10.35	8.98	8.09	7.51	7.02
	sFID↓	23.22	9.43	8.28	7.80	7.39	7.31	7.08	6.98
	IS↑	33.9	73.4	93.3	106.6	115.6	122.4	128.4	133.6
	Prec.↑	0.597	0.715	0.736	0.744	0.749	0.752	0.755	0.759
	Rec.↑	0.440	0.527	0.549	0.567	0.572	0.571	0.581	0.587
conv-$\boldsymbol{z}_t$	gFID↓	32.63	12.54	9.05	7.42	6.62	5.95	5.54	5.20
	sFID↓	12.69	**5.31**	4.83	**4.69**	4.61	**4.59**	**4.56**	**4.55**
	IS↑	41.0	88.0	108.1	121.3	129.6	135.3	140.7	144.9
	Prec.↑	0.595	0.734	0.749	0.754	0.757	0.759	0.756	0.759
	Rec.↑	0.471	0.563	0.579	0.591	0.595	0.610	0.613	0.618
conv-$\boldsymbol{z}$	gFID↓	30.02	10.81	7.94	6.59	5.94	5.39	5.11	4.81
	sFID↓	12.33	5.66	5.40	5.45	5.47	5.46	5.43	5.42
	IS↑	40.9	94.7	114.8	126.5	133.6	139.5	143.5	147.1
	Prec.↑	0.614	**0.747**	**0.757**	0.758	**0.762**	**0.764**	**0.766**	0.763
	Rec.↑	0.499	0.561	0.581	0.591	0.595	0.607	0.608	0.605
conv+t-$\boldsymbol{z}_t$	gFID↓	32.69	12.36	8.83	7.18	6.38	5.83	5.40	5.05
	sFID↓	15.74	5.49	4.99	4.89	4.87	4.83	4.85	4.85
	IS↑	39.6	88.9	110.0	124.0	131.0	137.7	142.4	147.8
	Prec.↑	0.587	0.732	0.747	0.757	0.759	0.762	0.760	0.761
	Rec.↑	0.488	0.555	0.577	0.592	0.597	0.603	0.612	0.615

Tab. 7: **Detailed alignment-head ablations: IN-VAE (*f16d32*).** Columns are downstream SiT-XL/1 steps; bold marks the best displayed value per metric and checkpoint.

Alignment head	Metric	50k	100k	150k	200k	250k	300k	350k	400k
SiT-8-z_t	gFID↓	22.06	9.74	6.94	5.84	5.12	4.71	4.33	4.13
	sFID↓	10.00	5.49	5.05	4.93	4.86	4.82	4.81	4.81
	IS↑	51.1	**98.5**	118.0	130.2	137.4	143.5	148.4	152.0
	Prec.↑	0.645	0.729	0.741	0.748	0.753	0.749	0.751	0.753
	Rec.↑	0.581	0.600	0.615	0.618	0.631	0.627	**0.637**	0.638
SiT-8-z	gFID↓	29.80	14.65	10.96	9.42	8.48	7.98	7.47	7.31
	sFID↓	10.52	5.72	5.27	5.24	5.24	5.22	5.26	5.28
	IS↑	39.8	78.3	95.8	106.8	113.6	117.8	122.6	124.6
	Prec.↑	0.562	0.678	0.699	0.707	0.713	0.712	0.716	0.714
	Rec.↑	**0.614**	**0.620**	**0.626**	**0.633**	**0.634**	**0.643**	0.635	**0.644**
SiT-4-z_t	gFID↓	**21.82**	9.74	6.88	5.64	4.98	4.55	4.23	3.96
	sFID↓	8.99	5.18	4.77	4.65	4.63	4.52	4.52	4.54
	IS↑	51.7	97.6	118.9	131.2	139.2	145.2	150.1	154.1
	Prec.↑	0.647	0.733	**0.745**	0.750	0.754	0.754	0.755	0.755
	Rec.↑	0.586	0.605	0.620	0.625	0.628	0.628	0.634	0.642
SiT-1-z_t	gFID↓	21.95	**9.44**	**6.67**	**5.53**	**4.86**	**4.44**	**4.04**	**3.79**
NA-VAE	sFID↓	11.28	5.49	5.03	4.90	4.76	4.70	4.68	4.66
	IS↑	**53.1**	**98.5**	**119.5**	**131.6**	**139.8**	**146.4**	**152.3**	**156.4**
	Prec.↑	**0.653**	**0.734**	**0.745**	**0.753**	**0.755**	0.756	0.756	0.761
	Rec.↑	0.563	0.605	0.609	0.614	0.629	0.631	0.632	0.634
SiT-0-z_t	gFID↓	28.80	13.01	9.23	7.50	6.38	5.68	5.21	4.87
	sFID↓	10.56	5.18	4.90	4.72	4.65	4.54	4.53	4.50
	IS↑	42.5	83.3	104.6	117.8	127.9	134.9	141.9	145.4
	Prec.↑	0.599	0.705	0.730	0.737	0.745	0.750	0.752	0.754
	Rec.↑	0.548	0.594	0.598	0.617	0.621	0.620	0.621	0.628
MLP-z_t	gFID↓	30.49	13.51	9.52	7.58	6.52	5.86	5.33	4.91
	sFID↓	10.97	5.82	5.47	5.21	5.03	4.96	4.98	4.87
	IS↑	41.2	82.5	103.1	117.9	128.2	134.7	141.2	146.3
	Prec.↑	0.602	0.707	0.732	0.739	0.745	0.745	0.749	0.753
	Rec.↑	0.554	0.586	0.602	0.610	0.617	0.615	0.629	0.625
MLP-z	gFID↓	25.51	12.31	8.78	7.18	6.32	5.66	5.12	4.74
	sFID↓	**7.07**	5.00	4.77	4.67	4.59	4.53	4.50	4.48
	IS↑	46.5	86.3	107.7	120.4	128.3	137.7	142.5	147.6
	Prec.↑	0.642	0.727	0.742	0.746	0.754	**0.757**	0.758	**0.763**
	Rec.↑	0.559	0.579	0.591	0.605	0.611	0.614	0.613	0.620
conv-z_t	gFID↓	23.74	10.59	7.55	6.20	5.37	4.80	4.46	4.13
	sFID↓	9.99	5.04	4.75	4.54	4.44	4.40	4.38	4.35
	IS↑	49.6	94.1	114.5	127.1	136.5	142.6	147.9	152.9
	Prec.↑	0.643	0.723	0.742	0.745	0.751	0.753	0.754	0.759
	Rec.↑	0.565	0.602	0.618	0.623	0.628	0.637	0.633	0.635
conv-z	gFID↓	24.74	11.22	8.15	6.68	5.84	5.38	4.98	4.60
	sFID↓	7.93	**4.61**	**4.40**	**4.35**	**4.31**	**4.27**	**4.27**	**4.25**
	IS↑	47.0	90.3	110.6	121.2	130.2	135.8	140.8	145.3
	Prec.↑	0.629	0.721	0.734	0.746	0.749	0.752	0.753	0.758
	Rec.↑	0.591	0.608	0.607	0.623	0.622	0.635	0.627	0.633
conv+t-z_t	gFID↓	24.37	10.61	7.46	6.20	5.37	4.80	4.40	4.09
	sFID↓	10.30	4.97	4.69	4.56	4.48	4.42	4.40	4.39
	IS↑	48.2	92.9	114.1	125.7	135.3	140.4	147.4	151.5
	Prec.↑	0.639	0.725	0.741	0.750	**0.755**	0.756	**0.761**	0.759
	Rec.↑	0.563	0.607	0.611	0.616	0.631	0.625	0.629	0.634

Tab. 8: **Detailed alignment-head ablations: VA-VAE (*f16d32*).** Columns are downstream SiT-XL/1 steps; bold marks the best displayed value per metric and checkpoint.

Alignment head	Metric	50k	100k	150k	200k	250k	300k	350k	400k
SiT-8-z_t	gFID↓	**15.90**	7.17	5.22	**4.34**	**3.80**	**3.47**	**3.19**	**3.00**
	sFID↓	7.50	4.88	4.67	4.63	4.59	4.57	4.53	4.57
	IS↑	66.2	113.0	132.9	143.5	152.4	157.8	163.8	166.6
	Prec.↑	**0.707**	0.765	0.770	**0.774**	0.776	0.777	0.776	**0.779**
	Rec.↑	0.567	0.586	0.605	0.613	0.613	0.621	0.625	**0.634**
SiT-8-z	gFID↓	21.63	10.32	7.79	6.67	6.08	5.70	5.36	5.19
	sFID↓	**6.41**	5.02	4.82	4.80	4.81	4.84	4.86	4.85
	IS↑	52.2	95.6	114.7	124.7	130.2	135.0	138.6	141.0
	Prec.↑	0.661	0.736	0.747	0.749	0.749	0.749	0.752	0.749
	Rec.↑	**0.587**	0.590	0.609	**0.616**	0.621	**0.627**	**0.633**	0.626
SiT-4-z_t	gFID↓	**15.90**	7.09	5.27	4.40	3.84	3.59	3.30	3.10
	sFID↓	8.10	4.84	4.59	**4.46**	**4.46**	**4.41**	**4.40**	**4.40**
	IS↑	66.7	114.8	133.4	144.6	154.4	159.2	164.9	167.5
	Prec.↑	0.699	0.765	**0.771**	0.773	0.775	0.776	0.775	0.778
	Rec.↑	0.555	0.582	0.592	0.607	0.611	0.613	0.621	0.624
SiT-1-z_t	gFID↓	16.01	**6.98**	**5.18**	**4.34**	**3.80**	3.50	3.24	3.06
NA-VAE	sFID↓	9.89	5.47	5.15	5.03	5.01	4.87	4.89	4.90
	IS↑	**67.9**	**116.2**	**135.5**	**146.8**	**155.4**	**161.4**	**166.0**	**170.1**
	Prec.↑	0.697	**0.766**	0.769	**0.774**	**0.779**	**0.779**	**0.779**	**0.779**
	Rec.↑	0.557	0.585	0.598	0.606	0.613	0.618	0.621	0.623
SiT-0-z_t	gFID↓	20.14	8.75	6.41	5.33	4.63	4.27	3.91	3.68
	sFID↓	10.07	5.64	5.31	5.20	5.15	5.12	5.08	5.06
	IS↑	56.7	104.7	123.4	135.4	143.4	150.9	156.9	160.4
	Prec.↑	0.655	0.750	0.761	0.765	0.769	0.773	0.770	0.773
	Rec.↑	0.540	0.582	0.599	0.605	0.611	0.613	0.615	0.614
MLP-z_t	gFID↓	20.30	9.18	6.77	5.63	4.91	4.44	4.10	3.85
	sFID↓	10.23	6.20	5.90	5.76	5.70	5.58	5.45	5.48
	IS↑	56.3	100.9	120.3	132.7	142.0	149.5	153.2	158.7
	Prec.↑	0.668	0.746	0.759	0.766	0.767	0.769	0.771	0.773
	Rec.↑	0.536	0.574	0.592	0.596	0.601	0.610	0.611	0.614
MLP-z	gFID↓	19.61	9.05	6.78	5.79	5.11	4.70	4.35	4.15
	sFID↓	9.41	5.86	5.58	5.52	5.51	5.38	5.35	5.33
	IS↑	59.1	104.0	122.7	133.5	142.6	148.9	155.1	157.8
	Prec.↑	0.668	0.751	0.761	0.765	0.769	0.767	0.772	0.771
	Rec.↑	0.540	0.569	0.584	0.587	0.591	0.596	0.595	0.611
conv-z_t	gFID↓	17.96	7.81	5.72	4.70	4.12	3.77	3.47	3.25
	sFID↓	11.12	5.54	5.27	5.16	5.00	4.93	4.92	4.90
	IS↑	61.5	109.7	128.7	140.8	150.1	156.2	161.6	165.0
	Prec.↑	0.675	0.755	0.765	0.769	0.769	0.770	0.768	0.773
	Rec.↑	0.561	0.594	**0.611**	0.610	0.616	0.623	0.627	0.632
conv-z	gFID↓	19.07	8.37	6.20	5.11	4.52	4.14	3.80	3.60
	sFID↓	7.81	**4.74**	**4.53**	4.52	4.48	4.45	4.50	4.50
	IS↑	56.8	105.1	123.4	135.8	142.6	149.0	154.4	158.8
	Prec.↑	0.672	0.751	0.764	0.768	0.770	0.768	0.770	0.772
	Rec.↑	0.575	**0.597**	0.603	0.608	0.615	0.619	0.624	0.626
conv+t-z_t	gFID↓	17.64	7.85	5.66	4.67	4.15	3.74	3.42	3.20
	sFID↓	9.95	5.48	5.18	5.02	4.93	4.87	4.86	4.84
	IS↑	63.0	110.0	129.4	141.4	149.9	156.0	162.3	167.0
	Prec.↑	0.686	0.753	0.763	0.768	0.770	0.771	0.772	0.772
	Rec.↑	0.556	**0.597**	0.604	0.611	**0.624**	0.620	0.623	0.629

C.5 Ablation of the Noise Schedule

We supplement the *Noise Schedule* study in §4.2 and Fig. 4b. Tabs. 9–11 report all five metrics at eight checkpoints, including additional truncated-uniform variants. We fix the head to SiT-8 and train each tokenizer for 100k steps (20 epochs), then freeze it for 400k downstream steps (80 epochs), with DINOv2-B and no CFG (§C.1 and §C.2).

Noise sampling. Uniform variants use $t \sim \mathrm{U}(0, t_{\max})$ with $t_{\max} \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$; $t = 0$ is clean alignment. As in Fig. 4b, $\mathrm{lognorm}(m, s)$ denotes a *logit-normal* distribution over t, defined by

$$t = \mathrm{sigmoid}(m + s\epsilon), \qquad \epsilon \sim \mathcal{N}(0, 1), \qquad m \in \{-1, 0, 1\}, \quad s = 1.$$

Here m and s are the mean and standard deviation before the sigmoid; positive/negative m favors higher/lower noise. Shading highlights the adopted $\mathrm{U}(0, 1)$ schedule, not a change in head architecture.

Tab. 9: **Detailed noise-schedule ablations: SD-VAE (*f8d4*).** All variants use SiT-8. Columns are downstream SiT-XL/2 steps; bold marks the best displayed value per metric and checkpoint.

Noise schedule	Metric	50k	100k	150k	200k	250k	300k	350k	400k
$\mathrm{U}(0, 1)$	gFID↓	28.69	**10.93**	**7.85**	**6.66**	**5.92**	**5.49**	**5.13**	**4.92**
	sFID↓	**6.84**	5.41	5.46	5.55	5.50	5.59	5.60	5.62
	IS↑	45.2	95.2	115.9	**127.8**	**135.8**	**140.2**	**145.9**	**148.5**
	Prec.↑	**0.622**	0.732	0.746	0.752	0.753	0.753	0.754	0.754
	Rec.↑	0.518	0.575	0.592	0.600	0.613	0.616	0.623	0.624
$\mathrm{lognorm}(+1, 1)$	gFID↓	34.94	13.77	9.75	7.92	7.00	6.29	5.78	5.43
	sFID↓	9.29	5.67	5.24	5.18	4.99	4.90	4.88	4.84
	IS↑	39.3	84.8	106.6	119.8	126.0	133.7	139.8	144.0
	Prec.↑	0.602	0.732	0.746	**0.756**	**0.759**	**0.758**	**0.763**	**0.765**
	Rec.↑	0.452	0.544	0.572	0.577	0.590	0.600	0.604	0.597
$\mathrm{lognorm}(0, 1)$	gFID↓	**28.58**	10.94	7.95	6.71	6.01	5.60	5.27	5.07
	sFID↓	7.07	**4.89**	4.87	4.94	4.97	4.97	4.98	5.01
	IS↑	**45.5**	**96.3**	**116.0**	127.6	135.6	139.5	143.6	145.6
	Prec.↑	0.620	**0.739**	**0.749**	0.754	0.753	0.756	0.754	0.757
	Rec.↑	0.513	0.570	0.588	0.599	0.615	0.619	0.613	0.616
$\mathrm{lognorm}(-1, 1)$	gFID↓	29.57	12.18	8.65	7.42	6.74	6.34	6.07	5.84
	sFID↓	14.02	5.40	4.88	4.76	4.79	4.79	4.83	4.86
	IS↑	40.2	87.8	108.3	118.4	125.2	128.7	132.4	135.3
	Prec.↑	0.609	0.728	0.739	0.746	0.746	0.745	0.744	0.743
	Rec.↑	0.538	0.582	0.599	0.610	0.612	0.620	0.624	0.635
$t = 0$	gFID↓	44.45	16.13	11.61	9.78	8.73	8.07	7.63	7.41
	sFID↓	28.42	5.65	4.75	4.71	4.69	4.72	4.77	4.79
	IS↑	26.8	73.2	94.3	105.5	112.7	118.6	123.0	125.1
	Prec.↑	0.474	0.675	0.706	0.712	0.717	0.719	0.718	0.718
	Rec.↑	0.549	**0.597**	0.608	**0.619**	**0.625**	**0.631**	**0.645**	**0.642**
$\mathrm{U}(0, 0.2)$	gFID↓	37.47	15.92	11.54	9.86	8.85	8.21	7.91	7.61
	sFID↓	18.43	5.68	4.82	4.74	4.72	4.72	4.75	4.77
	IS↑	31.6	72.8	93.3	103.5	110.6	114.9	118.0	120.6
	Prec.↑	0.526	0.680	0.706	0.711	0.716	0.718	0.719	0.716
	Rec.↑	**0.562**	**0.597**	0.601	0.618	**0.625**	0.630	0.625	0.634
$\mathrm{U}(0, 0.4)$	gFID↓	36.70	16.82	12.37	10.50	9.49	8.94	8.58	8.25
	sFID↓	16.40	5.83	4.91	4.76	4.70	4.72	4.74	4.79
	IS↑	31.4	70.0	88.5	99.7	105.2	109.5	112.6	115.8
	Prec.↑	0.543	0.680	0.705	0.710	0.714	0.717	0.714	0.714
	Rec.↑	**0.562**	0.588	**0.611**	0.611	0.618	0.618	0.625	0.629
$\mathrm{U}(0, 0.6)$	gFID↓	33.09	13.83	10.18	8.72	7.95	7.36	7.09	6.85
	sFID↓	22.06	5.93	**4.71**	**4.54**	**4.53**	**4.53**	**4.55**	**4.55**
	IS↑	35.4	79.8	99.1	109.7	115.5	120.5	122.8	125.3
	Prec.↑	0.573	0.714	0.726	0.728	0.730	0.731	0.732	0.731
	Rec.↑	0.537	0.581	0.598	0.605	0.611	0.619	0.627	0.628
$\mathrm{U}(0, 0.8)$	gFID↓	29.26	11.77	8.54	7.13	6.41	6.01	5.62	5.41
	sFID↓	8.55	5.31	5.10	5.11	5.12	5.07	5.10	5.08
	IS↑	42.7	90.6	111.0	122.0	128.6	133.9	137.1	140.5
	Prec.↑	**0.622**	0.732	0.742	0.747	0.751	0.750	0.747	0.748
	Rec.↑	0.521	0.574	0.595	0.605	0.608	0.610	0.623	0.622

Tab. 10: **Detailed noise-schedule ablations: IN-VAE (*f16d32*).** All variants use SiT-8. Columns are downstream SiT-XL/1 steps; bold marks the best displayed value per metric and checkpoint.

Noise schedule	Metric	50k	100k	150k	200k	250k	300k	350k	400k
$\mathrm{U}(0,1)$	gFID↓	**22.06**	**9.74**	**6.94**	**5.84**	**5.12**	**4.71**	**4.33**	4.13
	sFID↓	10.00	5.49	5.05	4.93	4.86	4.82	4.81	4.81
	IS↑	**51.1**	**98.5**	**118.0**	**130.2**	**137.4**	**143.5**	**148.4**	152.0
	Prec.↑	0.645	0.729	0.741	**0.748**	0.753	0.749	0.751	0.753
	Rec.↑	0.581	0.600	0.615	0.618	0.631	0.627	0.637	0.638
$\mathrm{lognorm}(+1,1)$	gFID↓	23.17	10.37	7.31	6.04	5.21	4.73	4.36	**4.10**
	sFID↓	8.71	5.08	4.74	4.58	4.53	4.47	4.46	**4.41**
	IS↑	50.5	93.5	115.1	126.2	135.8	142.4	148.0	**152.1**
	Prec.↑	0.645	0.728	**0.746**	0.747	**0.755**	**0.758**	**0.759**	**0.760**
	Rec.↑	0.572	0.604	0.609	0.621	0.617	0.623	0.626	0.625
$\mathrm{lognorm}(0,1)$	gFID↓	22.21	10.04	7.30	6.12	5.50	5.06	4.73	4.48
	sFID↓	9.58	5.39	5.05	4.95	4.89	4.87	4.84	4.84
	IS↑	50.2	96.8	116.9	127.7	134.5	140.0	144.2	148.0
	Prec.↑	**0.647**	0.730	0.743	0.745	0.747	0.747	0.751	0.749
	Rec.↑	0.584	0.602	0.612	0.618	0.619	0.626	0.633	0.634
$\mathrm{lognorm}(-1,1)$	gFID↓	26.05	12.51	9.18	7.68	6.91	6.38	6.09	5.77
	sFID↓	**7.90**	5.17	4.76	4.70	4.68	4.64	4.69	4.67
	IS↑	42.9	83.4	102.1	113.0	120.7	125.3	128.8	131.5
	Prec.↑	0.615	0.710	0.726	0.729	0.735	0.736	0.738	0.736
	Rec.↑	0.607	0.601	0.609	0.614	0.621	0.628	0.627	0.636
$t=0$	gFID↓	29.80	14.65	10.96	9.42	8.48	7.98	7.47	7.31
	sFID↓	10.52	5.72	5.27	5.24	5.24	5.22	5.26	5.28
	IS↑	39.8	78.3	95.8	106.8	113.6	117.8	122.6	124.6
	Prec.↑	0.562	0.678	0.699	0.707	0.713	0.712	0.716	0.714
	Rec.↑	**0.614**	**0.620**	**0.626**	**0.633**	0.634	**0.643**	0.635	**0.644**
$\mathrm{U}(0,0.2)$	gFID↓	30.10	14.41	10.76	9.13	8.34	7.84	7.64	7.25
	sFID↓	9.64	5.28	4.96	4.91	4.94	4.93	5.00	4.99
	IS↑	39.0	77.8	95.9	106.1	112.0	116.7	119.4	123.6
	Prec.↑	0.579	0.687	0.706	0.711	0.715	0.717	0.712	0.714
	Rec.↑	0.602	0.619	0.622	0.620	**0.637**	0.641	**0.642**	**0.644**
$\mathrm{U}(0,0.4)$	gFID↓	31.61	15.50	11.41	9.59	8.62	8.03	7.65	7.37
	sFID↓	8.87	5.49	5.15	5.07	5.04	5.03	5.09	5.10
	IS↑	36.7	74.0	91.7	102.0	108.5	113.4	118.0	119.3
	Prec.↑	0.571	0.687	0.709	0.717	0.717	0.720	0.720	0.719
	Rec.↑	0.605	0.610	0.619	0.620	0.623	0.621	0.629	0.632
$\mathrm{U}(0,0.6)$	gFID↓	29.45	14.16	10.40	8.71	7.93	7.35	6.99	6.74
	sFID↓	10.39	5.19	4.83	4.81	4.80	4.84	4.87	4.83
	IS↑	38.1	77.1	96.5	107.2	113.5	118.2	121.8	123.4
	Prec.↑	0.590	0.702	0.721	0.725	0.729	0.729	0.732	0.730
	Rec.↑	0.597	0.598	0.609	0.616	0.621	0.628	0.627	0.631
$\mathrm{U}(0,0.8)$	gFID↓	23.13	10.79	7.98	6.66	6.04	5.56	5.15	4.95
	sFID↓	8.96	**4.89**	**4.59**	**4.46**	**4.45**	**4.42**	**4.42**	**4.41**
	IS↑	48.0	91.9	110.7	120.6	128.2	133.4	138.1	140.8
	Prec.↑	0.646	**0.733**	0.742	0.747	0.747	0.745	0.749	0.751
	Rec.↑	0.586	0.596	0.607	0.613	0.617	0.628	0.628	0.637

Tab. 11: **Detailed noise-schedule ablations: VA-VAE (*f16d32*).** All variants use SiT-8. Columns are downstream SiT-XL/1 steps; bold marks the best displayed value per metric and checkpoint.

Noise schedule	Metric	50k	100k	150k	200k	250k	300k	350k	400k
$\mathrm{U}(0,1)$	gFID↓	**15.90**	**7.17**	**5.22**	**4.34**	**3.80**	**3.47**	**3.19**	**3.00**
	sFID↓	7.50	4.88	4.67	4.63	4.59	4.57	4.53	4.57
	IS↑	**66.2**	**113.0**	**132.9**	**143.5**	**152.4**	**157.8**	**163.8**	**166.6**
	Prec.↑	**0.707**	**0.765**	0.770	0.774	0.776	**0.777**	0.776	0.779
	Rec.↑	0.567	0.586	0.605	0.613	0.613	0.621	0.625	**0.634**
$\mathrm{lognorm}(+1,1)$	gFID↓	17.32	7.73	5.65	4.78	4.13	3.82	3.48	3.29
	sFID↓	9.40	5.19	4.84	4.74	4.69	4.62	4.56	4.59
	IS↑	63.9	109.1	128.3	139.7	148.8	154.7	160.2	163.8
	Prec.↑	0.690	0.761	0.769	0.773	0.776	0.775	0.776	**0.780**
	Rec.↑	0.550	0.579	0.595	0.602	0.608	0.609	0.614	0.614
$\mathrm{lognorm}(0,1)$	gFID↓	16.87	7.75	5.72	4.84	4.22	3.92	3.66	3.48
	sFID↓	8.41	5.23	4.93	4.79	4.76	4.71	4.68	4.69
	IS↑	63.0	109.6	126.8	137.6	146.1	151.1	155.6	159.6
	Prec.↑	0.702	0.762	0.771	0.773	**0.779**	0.775	**0.778**	0.777
	Rec.↑	0.564	0.583	0.598	0.602	0.608	0.612	0.614	0.618
$\mathrm{lognorm}(-1,1)$	gFID↓	17.77	8.28	6.27	5.32	4.78	4.40	4.14	3.91
	sFID↓	7.38	**4.75**	4.55	4.44	4.43	4.39	4.37	4.39
	IS↑	60.2	105.2	123.0	133.7	139.7	144.5	148.0	151.3
	Prec.↑	0.699	**0.765**	0.770	**0.775**	0.773	0.769	0.773	0.773
	Rec.↑	0.573	0.587	0.598	0.603	0.611	0.611	0.614	0.620
$t=0$	gFID↓	21.63	10.32	7.79	6.67	6.08	5.70	5.36	5.19
	sFID↓	6.41	5.02	4.82	4.80	4.81	4.84	4.86	4.85
	IS↑	52.2	95.6	114.7	124.7	130.2	135.0	138.6	141.0
	Prec.↑	0.661	0.736	0.747	0.749	0.749	0.749	0.752	0.749
	Rec.↑	0.587	**0.590**	**0.609**	**0.616**	**0.621**	**0.627**	**0.633**	0.626
$\mathrm{U}(0,0.2)$	gFID↓	20.73	10.06	7.65	6.61	6.01	5.65	5.43	5.25
	sFID↓	**6.23**	4.99	4.79	4.72	4.75	4.72	4.69	4.74
	IS↑	53.8	96.4	113.2	123.4	128.9	133.0	136.5	138.3
	Prec.↑	0.671	0.736	0.747	0.748	0.752	0.749	0.750	0.749
	Rec.↑	0.584	0.581	0.601	0.606	0.614	0.618	0.627	0.626
$\mathrm{U}(0,0.4)$	gFID↓	21.81	10.74	7.98	6.90	6.20	5.80	5.55	5.25
	sFID↓	7.52	5.25	4.95	4.87	4.89	4.86	4.83	4.84
	IS↑	50.3	91.6	109.2	118.6	126.0	130.2	132.8	136.0
	Prec.↑	0.664	0.745	0.753	0.756	0.755	0.754	0.755	0.752
	Rec.↑	**0.591**	0.583	0.597	0.605	0.607	0.611	0.622	0.618
$\mathrm{U}(0,0.6)$	gFID↓	20.78	10.10	7.48	6.45	5.78	5.43	5.16	4.82
	sFID↓	7.96	5.03	4.73	4.66	4.62	4.61	4.59	4.60
	IS↑	52.3	94.4	112.5	121.3	128.7	132.7	136.0	139.8
	Prec.↑	0.679	0.753	0.760	0.762	0.763	0.761	0.759	0.760
	Rec.↑	0.578	0.579	0.589	0.595	0.605	0.611	0.618	0.620
$\mathrm{U}(0,0.8)$	gFID↓	17.05	8.27	6.15	5.17	4.59	4.26	3.98	3.77
	sFID↓	7.66	4.80	**4.50**	**4.37**	**4.33**	**4.32**	**4.29**	**4.29**
	IS↑	62.0	104.7	123.2	135.2	140.8	146.3	151.4	154.6
	Prec.↑	0.705	**0.765**	**0.772**	0.773	0.777	0.775	0.776	0.777
	Rec.↑	0.563	0.583	0.591	0.595	0.604	0.618	0.610	0.618

C.6 Ablation of Model Size

We supplement the *Model Size* study in §4.2 and Fig. 5a with SiT-S/2 and detailed training trajectories for all four backbone sizes. All runs reuse the same SD-VAE-based NA-VAE (*f8d4*), trained for 100k steps (20 epochs) with the adopted SiT-1 alignment head and then frozen. Only the downstream backbone size changes: SiT-S/2, SiT-B/2, SiT-L/2, or SiT-XL/2. Each is trained for 400k steps (80 epochs) on ImageNet 256×256, with DINOv2-B alignment at layer 8 and a REPA coefficient of 0.5. We evaluate 50,000 samples per checkpoint without CFG (§C.1 and §C.2).

Tab. 12 reports all five metrics at eight checkpoints. Larger backbones consistently achieve lower gFID and higher IS at matched training steps. Relative to REPA with the original SD-VAE, NA-VAE reduces 400k-step gFID by 45.4% for SiT-L/2 and 43.5% for SiT-XL/2, while improving IS and precision; recall is lower in both comparisons. These results show that a single frozen NA-VAE supports downstream models of different capacities without tokenizer retraining.

Tab. 12: **Detailed model-size ablations on ImageNet** 256×256**.** Columns are downstream training steps; shading marks NA-VAE. REPA results are from Yu et al. (2025); — denotes unreported checkpoints. Bold marks the best displayed value per metric at the same backbone size and checkpoint, only where both methods are available.

Model / method	Metric	50k	100k	150k	200k	250k	300k	350k	400k
SiT-S/2	gFID↓	88.47	56.77	45.35	39.93	36.48	33.97	32.17	30.80
NA-VAE	sFID↓	14.13	9.65	8.71	8.37	8.16	7.85	7.68	7.52
	IS↑	14.5	22.6	29.4	34.3	38.4	41.3	43.6	45.8
	Prec.↑	0.316	0.482	0.529	0.555	0.571	0.581	0.587	0.594
	Rec.↑	0.302	0.481	0.532	0.548	0.561	0.573	0.572	0.566
SiT-B/2	gFID↓	57.28	28.96	22.87	19.86	17.71	16.28	15.33	14.51
NA-VAE	sFID↓	9.42	6.12	5.88	5.74	5.54	5.40	5.36	5.32
	IS↑	22.4	46.2	58.1	66.6	72.9	78.6	81.5	85.2
	Prec.↑	0.451	0.615	0.645	0.658	0.667	0.678	0.679	0.687
	Rec.↑	0.452	0.563	0.582	0.592	0.596	0.593	0.597	0.599
SiT-L/2	gFID↓	55.40	24.10	—	14.00	—	—	—	10.00
REPA	sFID↓	24.00	6.25	—	5.18	—	—	—	5.20
	IS↑	23.0	55.7	—	86.5	—	—	—	109.2
	Prec.↑	0.430	0.620	—	0.670	—	—	—	0.690
	Rec.↑	**0.530**	**0.600**	—	**0.640**	—	—	—	**0.650**
SiT-L/2	gFID↓	**35.15**	**13.36**	9.61	**7.84**	6.86	6.27	5.83	**5.46**
NA-VAE	sFID↓	**9.08**	**5.06**	4.79	**4.72**	4.64	4.63	4.59	**4.60**
	IS↑	**39.0**	**84.8**	103.8	**117.6**	126.3	132.2	137.8	**141.0**
	Prec.↑	**0.575**	**0.721**	0.738	**0.746**	0.751	0.755	0.753	**0.757**
	Rec.↑	0.472	0.565	0.585	0.592	0.599	0.606	0.609	0.615
SiT-XL/2	gFID↓	52.30	19.40	—	11.10	—	—	—	7.90
REPA	sFID↓	31.24	6.06	—	5.05	—	—	—	5.06
	IS↑	24.3	67.4	—	100.4	—	—	—	122.6
	Prec.↑	0.450	0.640	—	0.690	—	—	—	0.700
	Rec.↑	**0.530**	**0.610**	—	**0.640**	—	—	—	**0.650**
SiT-XL/2	gFID↓	**32.91**	**11.44**	7.96	**6.56**	5.69	5.17	4.83	**4.47**
NA-VAE	sFID↓	**9.61**	**5.38**	4.80	**4.70**	4.60	4.61	4.59	**4.60**
	IS↑	**43.0**	**94.6**	116.1	**128.8**	138.1	144.1	149.2	**154.9**
	Prec.↑	**0.597**	**0.740**	0.753	**0.759**	0.762	0.763	0.764	**0.764**
	Rec.↑	0.461	0.556	0.576	0.592	0.605	0.604	0.612	0.617

C.7 Ablation of the Representation Encoder

We supplement the *Representation Encoder* study in §4.2 and Fig. 5b. Four SD-VAE-based tokenizers (*f8d4*) are trained for 100k steps (20 epochs) with SiT-1 and $t \sim \mathrm{U}(0, 1)$, varying only the tokenizer-stage VFM and its preprocessing (§C.3). Each is frozen for 400k-step (80-epoch) SiT-L/2 and SiT-XL/2 training on ImageNet 256×256; downstream REPA always uses DINOv2-B at layer 8 with coefficient 0.5. Evaluation uses 50,000 samples without CFG (§C.1 and §C.2).

Tabs. 13 and 14 report all checkpoints; the 100k SiT-L/2 results correspond to Fig. 5b. DINOv2-B gives the lowest gFID and highest IS from 100k steps onward at both scales, whereas DINOv2-L gives the lowest final sFID. Shading marks DINOv2-B; bold marks the best displayed value per metric and checkpoint.

Tab. 13: **Representation-encoder ablations: SiT-L/2.** Columns are downstream steps.

Tokenizer VFM	Metric	50k	100k	150k	200k	250k	300k	350k	400k
DINOv2-B	gFID↓	35.15	**13.36**	**9.61**	**7.84**	**6.86**	**6.27**	**5.83**	**5.46**
	sFID↓	**9.08**	**5.06**	**4.79**	4.72	4.64	**4.63**	4.59	4.60
	IS↑	**39.0**	**84.8**	**103.8**	**117.6**	**126.3**	**132.2**	**137.8**	**141.0**
	Prec.↑	0.575	**0.721**	**0.738**	**0.746**	**0.751**	**0.755**	**0.753**	**0.757**
	Rec.↑	0.472	0.565	0.585	0.592	0.599	0.606	0.609	0.615
DINOv2-L	gFID↓	37.04	13.70	9.92	8.12	7.11	6.48	5.95	5.61
	sFID↓	10.17	5.14	**4.79**	**4.71**	**4.61**	**4.63**	**4.56**	**4.55**
	IS↑	37.2	83.0	103.6	115.8	123.9	130.6	135.8	139.8
	Prec.↑	0.558	0.715	0.735	0.745	0.747	0.751	0.752	0.753
	Rec.↑	0.484	0.560	0.590	0.600	0.605	0.613	0.617	0.614
CLIP-L	gFID↓	42.55	16.25	11.75	9.69	8.57	7.79	7.34	7.00
	sFID↓	10.79	5.75	5.57	5.57	5.60	5.62	5.62	5.62
	IS↑	32.0	74.3	94.2	106.4	114.1	121.2	124.9	128.0
	Prec.↑	0.509	0.684	0.705	0.715	0.722	0.725	0.726	0.728
	Rec.↑	0.503	**0.580**	**0.596**	0.607	**0.611**	**0.617**	**0.623**	**0.627**
I-JEPA-H	gFID↓	**32.30**	14.69	10.93	9.08	7.95	7.12	6.61	6.30
	sFID↓	11.29	5.47	5.16	4.99	4.94	4.92	4.88	4.87
	IS↑	38.7	77.9	96.5	107.5	116.3	123.8	129.1	131.4
	Prec.↑	**0.611**	0.712	0.726	0.733	0.736	0.743	0.744	0.746
	Rec.↑	**0.518**	0.567	0.591	**0.611**	0.606	0.616	**0.623**	0.620

Tab. 14: **Representation-encoder ablations: SiT-XL/2.** Columns are downstream steps.

Tokenizer VFM	Metric	50k	100k	150k	200k	250k	300k	350k	400k
DINOv2-B	gFID↓	32.91	**11.44**	**7.96**	**6.56**	**5.69**	**5.17**	**4.83**	**4.47**
	sFID↓	**9.61**	**5.38**	4.80	4.70	4.60	4.61	4.59	4.60
	IS↑	43.0	**94.6**	**116.1**	**128.8**	**138.1**	**144.1**	**149.2**	**154.9**
	Prec.↑	0.597	**0.740**	**0.753**	**0.759**	**0.762**	**0.763**	**0.764**	**0.764**
	Rec.↑	0.461	0.556	0.576	0.592	0.605	0.604	0.612	0.617
DINOv2-L	gFID↓	36.87	11.81	8.15	6.66	5.90	5.38	4.90	4.66
	sFID↓	10.89	5.47	**4.79**	**4.66**	**4.57**	**4.53**	**4.51**	**4.52**
	IS↑	39.9	92.7	115.0	127.9	136.4	142.0	147.9	151.7
	Prec.↑	0.561	0.735	**0.753**	0.758	0.757	0.760	0.763	**0.764**
	Rec.↑	0.448	0.568	0.581	0.601	0.605	0.610	0.622	**0.624**
CLIP-L	gFID↓	41.15	13.22	9.38	7.80	6.95	6.46	5.92	5.62
	sFID↓	12.22	5.63	5.33	5.33	5.39	5.41	5.44	5.42
	IS↑	35.4	86.7	108.2	120.0	127.4	133.0	138.5	142.6
	Prec.↑	0.519	0.703	0.724	0.732	0.736	0.740	0.739	0.740
	Rec.↑	0.483	**0.581**	**0.594**	0.602	**0.614**	**0.618**	**0.625**	0.622
I-JEPA-H	gFID↓	**27.68**	11.64	8.52	7.08	6.31	5.72	5.34	4.98
	sFID↓	12.00	5.65	5.13	4.99	4.91	4.85	4.87	4.84
	IS↑	**46.1**	90.9	110.5	123.4	130.8	137.7	142.5	146.8
	Prec.↑	**0.637**	0.731	0.744	0.747	0.750	0.752	0.754	0.755
	Rec.↑	**0.512**	0.572	0.587	**0.605**	0.608	0.616	0.612	0.618

C.8 Component Analysis

We supplement the *Component Analysis* study in §4.2 and Fig. 6 with all five metrics in Tabs. 15–17. Tokenizers use 100k training steps (20 epochs) and remain frozen for 400k downstream steps (80 epochs) on ImageNet 256×256. Downstream REPA uses DINOv2-B; evaluation uses 50,000 samples without CFG (§C.1 and §C.2).

Three variants. *Noise only* reconstructs clean images from noised latents, without a tokenizer alignment head. *Align only* aligns clean latents; *noise + align* aligns noised latents while retaining clean-latent decoding. The two alignment variants use SiT-8, distinct from the adopted SiT-1 head. For *noise only*, the decoder input is

$$\boldsymbol{z}_t^{\mathrm{dec}} = (1-t)\boldsymbol{z} + t(\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}), \qquad t \sim \mathrm{U}(0,1), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

where $\boldsymbol{z}$ is the unnormalized latent and $\boldsymbol{\mu}, \boldsymbol{\sigma}$ are the running channel-wise mean and standard deviation. Noise is added in normalized latent space before restoring the original latent scale. This corruption is disabled for evaluation and latent caching.

Results. Noise + align achieves lower gFID and higher IS than either alternative at every checkpoint, reducing final gFID over clean alignment by 33.5–43.5%. The advantage is not uniform across sFID and recall.

Tab. 15: **Detailed component analysis: SD-VAE (*f8d4*).** Columns are downstream SiT-XL/2 steps. Shading marks noise + align; bold marks the best displayed value per metric and checkpoint.

Variant	Metric	50k	100k	150k	200k	250k	300k	350k	400k
Noise only	gFID↓	32.31	16.07	12.56	10.92	9.96	9.46	9.21	8.83
	sFID↓	7.97	6.45	6.73	6.82	6.91	7.08	7.11	7.22
	IS↑	37.4	74.4	91.8	101.2	108.4	113.0	115.9	118.7
	Prec.↑	0.580	0.692	0.711	0.715	0.716	0.717	0.717	0.718
	Rec.↑	**0.588**	0.590	0.606	0.603	0.612	0.621	0.628	0.630
Align only	gFID↓	44.45	16.13	11.61	9.78	8.73	8.07	7.63	7.41
	sFID↓	28.42	5.65	**4.75**	**4.71**	**4.69**	**4.72**	**4.77**	**4.79**
	IS↑	26.8	73.2	94.3	105.5	112.7	118.6	123.0	125.1
	Prec.↑	0.474	0.675	0.706	0.712	0.717	0.719	0.718	0.718
	Rec.↑	0.549	**0.597**	**0.608**	**0.619**	**0.625**	**0.631**	**0.645**	**0.642**
Noise + align	gFID↓	**28.69**	**10.93**	**7.85**	**6.66**	**5.92**	**5.49**	**5.13**	**4.92**
	sFID↓	**6.84**	**5.41**	5.46	5.55	5.50	5.59	5.60	5.62
	IS↑	**45.2**	**95.2**	**115.9**	**127.8**	**135.8**	**140.2**	**145.9**	**148.5**
	Prec.↑	**0.622**	**0.732**	**0.746**	**0.752**	**0.753**	**0.753**	**0.754**	**0.754**
	Rec.↑	0.518	0.575	0.592	0.600	0.613	0.616	0.623	0.624

Tab. 16: **Detailed component analysis: IN-VAE (*f16d32*).** Columns are downstream SiT-XL/1 steps. Shading marks noise + align; bold marks the best displayed value per metric and checkpoint.

Variant	Metric	50k	100k	150k	200k	250k	300k	350k	400k
Noise only	gFID↓	28.44	14.69	10.93	9.25	8.40	7.78	7.29	7.11
	sFID↓	**6.42**	5.80	5.98	6.06	6.15	6.09	6.19	6.19
	IS↑	40.7	78.3	97.5	108.4	115.0	120.1	124.3	127.0
	Prec.↑	0.607	0.694	0.716	0.723	0.723	0.725	0.730	0.726
	Rec.↑	0.594	0.603	0.601	0.614	0.622	0.618	0.629	0.627
Align only	gFID↓	29.80	14.65	10.96	9.42	8.48	7.98	7.47	7.31
	sFID↓	10.52	5.72	5.27	5.24	5.24	5.22	5.26	5.28
	IS↑	39.8	78.3	95.8	106.8	113.6	117.8	122.6	124.6
	Prec.↑	0.562	0.678	0.699	0.707	0.713	0.712	0.716	0.714
	Rec.↑	**0.614**	**0.620**	**0.626**	**0.633**	**0.634**	**0.643**	0.635	**0.644**
Noise + align	gFID↓	**22.06**	**9.74**	**6.94**	**5.84**	**5.12**	**4.71**	**4.33**	**4.13**
	sFID↓	10.00	**5.49**	**5.05**	**4.93**	**4.86**	**4.82**	**4.81**	**4.81**
	IS↑	**51.1**	**98.5**	**118.0**	**130.2**	**137.4**	**143.5**	**148.4**	**152.0**
	Prec.↑	**0.645**	**0.729**	**0.741**	**0.748**	**0.753**	**0.749**	**0.751**	**0.753**
	Rec.↑	0.581	0.600	0.615	0.618	0.631	0.627	**0.637**	0.638

Tab. 17: **Detailed component analysis: VA-VAE (*f16d32*).** Columns are downstream SiT-XL/1 steps. Shading marks noise + align; bold marks the best displayed value per metric and checkpoint.

Variant	Metric	50k	100k	150k	200k	250k	300k	350k	400k
Noise only	gFID↓	25.38	13.23	9.93	8.50	7.56	7.11	6.75	6.52
	sFID↓	**6.08**	**4.84**	4.89	4.92	4.93	4.95	4.96	4.95
	IS↑	44.9	83.1	101.2	111.4	119.2	124.1	127.0	129.4
	Prec.↑	0.641	0.717	0.730	0.739	0.741	0.742	0.743	0.742
	Rec.↑	0.584	**0.596**	0.601	0.605	0.608	0.608	0.612	0.619
Align only	gFID↓	21.63	10.32	7.79	6.67	6.08	5.70	5.36	5.19
	sFID↓	6.41	5.02	4.82	4.80	4.81	4.84	4.86	4.85
	IS↑	52.2	95.6	114.7	124.7	130.2	135.0	138.6	141.0
	Prec.↑	0.661	0.736	0.747	0.749	0.749	0.749	0.752	0.749
	Rec.↑	**0.587**	0.590	**0.609**	**0.616**	**0.621**	**0.627**	**0.633**	0.626
Noise + align	gFID↓	**15.90**	**7.17**	**5.22**	**4.34**	**3.80**	**3.47**	**3.19**	**3.00**
	sFID↓	7.50	4.88	**4.67**	**4.63**	**4.59**	**4.57**	**4.53**	**4.57**
	IS↑	**66.2**	**113.0**	**132.9**	**143.5**	**152.4**	**157.8**	**163.8**	**166.6**
	Prec.↑	**0.707**	**0.765**	**0.770**	**0.774**	**0.776**	**0.777**	**0.776**	**0.779**
	Rec.↑	0.567	0.586	0.605	0.613	0.613	0.621	0.625	**0.634**

C.9 Text-to-Image Generation

We supplement the *Text to Image* study in §4.4 and Fig. 7 (right), using the 50k-step checkpoint of the FLUX.1-style NA-VAE trained from scratch on our internal corpus (§4.3). All tokenizers remain frozen during downstream training; E2E denotes previously end-to-end-tuned tokenizers (Leng et al., 2025b).

Training. We use the self-attention 3B DiT implementation of Fuse-DiT (Tang et al., 2025), with 30 blocks, width 2,048, and a frozen Gemma-2B text encoder. Training uses BLIP-3o long- and short-caption data (Chen et al., 2025c), 256×256 center crops, a 256-token text limit, and caption dropout of 0.1. We use AdamW with global batch size 1,024, constant learning rate 10^{-4}, weight decay 10^{-4}, and BF16 precision. Flow matching uses logit-normal $(0, 1)$ timestep sampling and EMA decay 0.9999. We report 50k–300k checkpoints from runs configured for 500k steps, without higher-resolution fine-tuning.

Evaluation. We sample with EMA weights, 25 Euler steps, CFG 6.0, and seed 42. Tab. 18 reports only overall GenEval (Ghosh et al., 2023), DPG-Bench (Hu et al., 2024), and GenAI-Bench (Li et al., 2024a) scores; all GenAI-Bench evaluations cover 1,600 prompts. FLUX.2 uses 128 latent values per DiT token versus 64 for the other tokenizers, and is included as a higher-capacity reference.

Tab. 18: **Text-to-image results for the tokenizers in Fig. 7.** Columns are downstream training steps; higher is better. Shading marks NA-VAE. Bold compares the five *f8d16* tokenizers, excluding the higher-capacity FLUX.2 reference (†).

Tokenizer	Benchmark	50k	100k	150k	200k	250k	300k
FLUX.1 VAE	GenEval	0.1456	0.2851	0.3621	0.4068	0.4517	0.4668
	DPG-Bench	62.01	70.94	74.06	75.49	76.70	77.06
	GenAI-Bench	0.5043	0.5881	0.6290	0.6496	0.6552	0.6676
E2E FLUX.1	GenEval	0.3587	0.4896	0.5748	0.6152	0.6376	0.6555
	DPG-Bench	73.61	78.15	80.10	**81.15**	81.45	81.90
	GenAI-Bench	0.6356	**0.6805**	0.6927	0.6951	0.7040	0.7088
Qwen-Image VAE	GenEval	0.2111	0.3214	0.4234	0.4619	0.4849	0.4991
	DPG-Bench	63.09	72.01	75.00	76.45	77.14	77.43
	GenAI-Bench	0.5558	0.6234	0.6539	0.6656	0.6738	0.6763
E2E Qwen-Image	GenEval	**0.3969**	**0.5461**	**0.6085**	**0.6302**	**0.6464**	**0.6664**
	DPG-Bench	74.31	78.54	80.56	80.77	81.67	82.17
	GenAI-Bench	**0.6481**	0.6761	**0.6991**	0.7047	0.7117	0.7127
NA-VAE	GenEval	0.3888	0.5303	0.5770	0.6163	0.6335	0.6584
	DPG-Bench	**74.69**	**78.95**	**80.63**	81.14	**81.90**	**82.25**
	GenAI-Bench	0.6289	0.6792	0.6936	**0.7066**	**0.7140**	**0.7154**
FLUX.2 VAE†	GenEval	0.4157	0.5654	0.6120	0.6508	0.6611	0.6994
	DPG-Bench	75.92	80.42	81.80	82.30	82.91	83.25
	GenAI-Bench	0.6536	0.6850	0.7056	0.7091	0.7171	0.7230

C.10 VIDEO GENERATION

We extend NA-VAE to class-conditional video generation on UCF-101 (Soomro et al., 2012), following the downstream training and evaluation framework of V-RAE (Guo et al., 2026). We compare the released Wan2.1 and Wan2.2 VAEs (Wan et al., 2025) with NA-VAE$_{\text{Wan2.1}}$, obtained by fine-tuning Wan2.1 for 100k tokenizer updates. We retain the original autoencoder architecture and compression ratio. All three tokenizers remain frozen during downstream training; tokenizer-specific adaptations and evaluation details are given below.

Tokenizers and matched sequence length. Tab. 19 summarizes the tokenizer geometry. Both Wan families are causal and compress time by a factor of 4: a 17-frame input produces five latent frames. We use spatial DiT patch size of 1 for Wan2.2 and 2 for the Wan2.1, yielding 1,280 tokens in each case. This matches sequence length, while tokenizer encoding costs can still differ.

Tab. 19: **Video tokenizers and downstream input geometry.** Compression is temporal $\times$ spatial $\times$ spatial; the latent shape is $T' \times H' \times W' \times C$ for a $17 \times 256 \times 256$ clip. Steps count tokenizer updates in this study; — denotes released weights. Shading marks NA-VAE.

Tokenizer	Params	Steps	Compression	Latent shape	Patch	Tokens
Wan2.1 VAE	126.9M	—	$4 \times 8 \times 8$	$5 \times 32 \times 32 \times 16$	2	1,280
Wan2.2 VAE	704.7M	—	$4 \times 16 \times 16$	$5 \times 16 \times 16 \times 48$	1	1,280
NA-VAE$_{\text{Wan2.1}}$	126.9M	100k	$4 \times 8 \times 8$	$5 \times 32 \times 32 \times 16$	2	1,280

Data and temporal alignment. Both training stages use UCF-101 (Soomro et al., 2012) split 1 training videos, whose manifest contains 9,537 records. Tokenizer training draws 64 consecutive frames at 256×256 with random horizontal flips; 9,364 videos satisfy this length requirement. The frozen V-JEPA2 ViT-L teacher (Assran et al., 2025) (`vjepa2-vitl-fpc64-256`) receives the full 64-frame window, while the VAE reconstructs its 17 frames at zero-based indices $6, 9, \ldots, 54$. The teacher's $32 \times 16 \times 16$ tubelet grid is pooled temporally into five groups (tubelet indices $\{3\}$, 4–9, 10–15, 16–21, and 22–27) and bilinearly upsampled to the VAE's $5 \times 32 \times 32$ grid. Downstream training uses 17-frame clips with temporal stride 3, random spatial crops, and random flips.

Video NA-VAE training. We jointly fine-tune the encoder, decoder, and alignment head without class labels or a diffusion loss. The head is a single spatio-temporal transformer block with width 1,152, 16 attention heads, query–key normalization, timestep conditioning, and fixed 3D sinusoidal positional embeddings. It uses unit patches and a three-layer MLP projector with hidden and output width 1,024 to match the V-JEPA2 features. As in Eq. 4, only the alignment branch receives the interpolant $\boldsymbol{z}_t = (1-t)\boldsymbol{z} + t\boldsymbol{\epsilon}$, with $t \sim \mathrm{U}(0,1)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$; here $\boldsymbol{z}$ denotes the normalized posterior sample. Normalization uses an affine-free channel-wise BatchNorm with momentum 0.1. The decoder reconstructs from the posterior sample before this alignment noise is added. The cosine-alignment loss contributes gradients with coefficients 1.5 to the encoder and 0.5 to the head. Reconstruction uses equally weighted L_1 and LPIPS (Zhang et al., 2018) evaluated on four frames per clip, plus KL and adversarial terms. Both training stages use AdamW (Loshchilov & Hutter, 2019), with optimization settings summarized in Tab. 20. The 100k training steps process 3.2M sampled windows, approximately 342 epochs over the 9,364 eligible videos. Tokenizer training ends at this budget while reconstruction rFVD is still decreasing.

Tab. 20: **Video training settings.** The tokenizer column applies only to NA-VAE; downstream settings are shared by all three variants. Batch sizes are global, with no gradient accumulation.

Setting	NA-VAE training	Downstream DiT training
Optimizer	AdamW	AdamW
Global batch / GPUs	32 / 8	64 / 8
Learning rate	2×10^{-5}	$10^{-4} \rightarrow 5 \times 10^{-5}$
Schedule	Constant	Linear decay over 447k steps
Warmup	None	1,490 steps from zero
Adam (β_1, β_2)	$(0.9, 0.999)$	$(0.9, 0.95)$
Adam ϵ / weight decay	10^{-8} / 0	10^{-8} / 0
Gradient clipping (max. norm)	1.0	1.0
Precision	BF16	BF16
DiT EMA decay	—	0.9995
Training steps reported	100k	Up to 300k

KL reduction for video latents. For posterior parameters of shape $B \times C \times T' \times H' \times W'$, we sum KL over channels and spatial positions and average over the batch and latent frames:

$$\mathcal{L}_{\mathrm{KL}}^{\mathrm{frame}} = \frac{1}{BT'} \sum_{b=1}^{B} \sum_{\tau=1}^{T'} \sum_{c,h,w} \frac{1}{2} \left(\mu_{bc\tau hw}^2 + \sigma_{bc\tau hw}^2 - 1 - \log \sigma_{bc\tau hw}^2 \right), \qquad \lambda_{\mathrm{KL}} = 10^{-6}, \tag{8}$$

where μ and σ denote the posterior mean and standard deviation, respectively.

Adversarial reconstruction. We enable a VideoMAE-based discriminator (Tong et al., 2022) for the final 40k of the 100k tokenizer updates. It reads features at depths 2, 5, 8, and 11, with kernel-size-9 heads, batch normalization, and spectral normalization (Miyato et al., 2018). Generator and discriminator loss weights are both 0.3; we use a non-saturating generator loss, a smoothed discriminator loss, LeCam regularization (Tseng et al., 2021) of weight 10^{-3}, DiffAugment (Zhao et al., 2020) with probability 1 and cutout fraction 0.2, and discriminator updates every five steps. The discriminator uses AdamW at 2×10^{-5} with $\beta = (0.9, 0.95)$ and weight decay 10^{-4}.

Downstream generator and compute. We use V-RAE's class-conditional video DiT (Guo et al., 2026), whose main encoder and output decoder have depths $(28, 2)$, widths $(1536, 2048)$, and attention heads $(24, 16)$, respectively. The MLP ratio is 4, and class dropout is 0.1 over 101 classes. Following RAEv2 (Singh et al., 2026b), an auxiliary clean-latent prediction head is attached after encoder block 8. The rectified-flow objective (Liu et al., 2023) uses clean-latent prediction, logit-normal $(0, 1)$ time sampling, a velocity-denominator clamp $t_\epsilon = 0.05$, and auxiliary loss coefficient 1.0. Following V-RAE's temporal extension of RAE's dimension-dependent noise schedule (Zheng et al., 2025), the time-shift factor uses the unpatched latent size, $\sqrt{T'H'W'C/4096}$: $\sqrt{20}$ for the Wan2.1 family and $\sqrt{15}$ for Wan2.2. Each tokenizer has precomputed latent normalization statistics, fixed during DiT training. Gradient checkpointing is disabled. The two Wan2.1 systems differ only in tokenizer weights and normalization statistics; generator architecture, data, seed, batch size, and optimization are shared. All runs use eight GPUs and the same 447k-step schedule (3,000 epochs, 149 steps per epoch). After 1,490 warmup steps, the learning rate decays linearly from 10^{-4} to 5×10^{-5} at 447k steps, rather than to zero. The Wan2.1 and NA-VAE runs complete this schedule, whereas Wan2.2 stops at 300k steps. We restrict comparisons to their shared checkpoints up to 300k.

Evaluation protocols. We compute Fréchet Video Distance (FVD) (Unterthiner et al., 2018) using the same I3D feature extractor (Carreira & Zisserman, 2017) for all three systems. Both reconstruction and generation use raw tokenizer weights, since neither the released tokenizers nor our tokenizer training maintain an EMA copy; this differs from V-RAE's stage-1 EMA evaluation. Reconstruction FVD (rFVD) compares original and reconstructed clips from a dedicated 17-frame population of the UCF-101 split 1 *test* set: 3,769 valid clips after excluding 14 of the 3,783 videos for insufficient length. We take 17 frames starting at frame 0 with stride 3, resize with antialiased bilinear interpolation, and center-crop to 256×256. This protocol respects the causal $T = 4n + 1$ geometry; we compare rFVD within this matched population rather than against V-RAE's 16-frame reconstruction results. Reconstruction evaluation uses FP32, global batch size 64, and sample covariance. PSNR is averaged over frames, whereas $\mathrm{PSNR}_{\mathrm{global}}$ is computed from the aggregate pixel MSE; LPIPS and SSIM (Wang et al., 2004) use the same reconstructed population.

Generation FVD (gFVD) uses 2,048 generated clips, following Latte (Ma et al., 2025b), and a real-feature reference drawn from the UCF-101 split 1 *training* set, as in V-RAE's evaluation implementation. We sample from DiT EMA weights with 100 sampling steps and seed 3407. Five latent frames decode to 17 video frames (`model_frames`=17), all scored at 256×256, with real-frame stride 3, generated-frame stride 1, center-square Lanczos preprocessing of real clips, and an MP4 round trip at 8 fps for generated clips. Generation uses BF16 and population covariance. All evaluation configs set global batch size 64; the runtime generation batch is 64 except for Wan2.1 at 15k, 45k, 90k, 150k, 210k, and 250k steps, which use batches of 8. **Classifier-free guidance is disabled** (CFG scale 1.0); internal guidance remains enabled at scale 1.3 over $t \in [0.1, 1.0]$, combining predictions as $\mathrm{base} + 1.3(\mathrm{full} - \mathrm{base})$ without an unconditional pass. Sampling steps, precision, guidance, and metric computation are shared across systems. Our absolute gFVD values should not be compared with published results using tuned CFG or different evaluation protocols.

Generation results. Tab. 21 reports gFVD at eight matched training checkpoints. NA-VAE outperforms both released tokenizers at every checkpoint. To account for checkpoint variation, we summarize the six shared checkpoints from 90k to 300k steps. The unweighted mean gFVD is 142.78 for NA-VAE, versus 216.41 for Wan2.1 and 201.88 for Wan2.2, corresponding to reductions

of 34.0% and 29.3%, respectively. Their checkpoint ranges are 127.54–159.66, 196.26–243.22, and 179.39–224.12, respectively; these describe within-run variation, not uncertainty across independent seeds. The persistent margin throughout this window supports the transfer of noise-aligned tokenizer training to video generation without joint tokenizer–generator optimization.

Tab. 21: **Class-conditional video generation on UCF-101.** Lower gFVD is better. All DiTs are trained on eight GPUs and evaluated with 2,048 generated clips, CFG 1.0, internal guidance 1.3, and DiT EMA weights. Tokenizers use frozen raw weights: released Wan2.1/Wan2.2 checkpoints and the 100k NA-VAE checkpoint. Shading marks NA-VAE.

Tokenizer / DiT steps	15k	45k	90k	135k	150k	210k	250k	300k
Wan2.1 VAE	654.07	317.86	243.22	231.40	212.01	197.94	196.26	217.64
Wan2.2 VAE	602.79	290.11	224.12	211.25	206.50	202.95	179.39	187.11
NA-VAE$_{\text{Wan2.1}}$	**279.46**	**172.94**	**159.66**	**141.27**	**148.18**	**127.54**	**135.96**	**144.07**

Reconstruction and adversarial ablation. Tab. 22 shows that NA-VAE also improves held-out reconstruction rFVD from 4.5645 to 3.7073 (18.8%) and lowers LPIPS, while sacrificing 0.67 dB of frame-averaged PSNR and about 0.005 SSIM. Wan2.2 has slightly better LPIPS, PSNR, and SSIM than Wan2.1 but higher rFVD (9.5707 versus 4.5645), illustrating that paired fidelity metrics alone do not determine distributional reconstruction quality or downstream generation.

Tab. 22: **Video reconstruction on the UCF-101 test split.** All rows use the same 3,769 clips and 17-frame protocol. Bold marks the best displayed value for each metric.

Tokenizer	Steps	rFVD↓	LPIPS↓	PSNR↑	PSNR$_{\text{global}}$↑	SSIM↑
Wan2.1 VAE	—	4.5645	0.066594	34.0218	**32.2015**	0.940428
Wan2.2 VAE	—	9.5707	0.065495	**34.0907**	32.1862	**0.941788**
NA-VAE$_{\text{Wan2.1}}$	100k	**3.7073**	**0.063464**	33.3541	31.7303	0.935845

C.11 Audio Generation

We extend NA-VAE to English zero-shot text-to-speech (TTS), using F5-TTS (Chen et al., 2025e) as the downstream generator, trained on LibriTTS (Zen et al., 2019). We compare four systems: native mel spectrograms, the released acoustic *Vanilla VAE*, *Vanilla VAE + acoustic fine-tuning*, and NA-VAE$_{\text{Acoustic}}$. The two adapted VAEs start from the same released checkpoint and receive 100k additional updates. Vanilla VAE + acoustic fine-tuning retains only the original acoustic objective, while NA-VAE$_{\text{Acoustic}}$ adds noisy semantic alignment. The subscript identifies the acoustic VAE used for initialization. All VAEs remain frozen throughout training of the separately initialized F5-TTS models. The acoustic fine-tuning control distinguishes the additional noisy-alignment objective from the effects of continued acoustic training.

Audio representations. The three VAE systems share the architecture and initialization of the released `acoustic_vae_dim64` checkpoint (Niu et al., 2025), with a DAC-style convolutional encoder (Kumar et al., 2023), a Gaussian posterior (Kingma & Welling, 2013), and a BigVGAN-based waveform decoder (Lee et al., 2022). They encode 16 kHz audio into 64-dimensional latents at 40 frames/s. Both adapted VAEs retain this architecture and compression ratio. The native-mel system instead uses 100 mel channels at 93.75 frames/s and the released 24 kHz Vocos decoder (Siuzdak, 2024). Tab. 23 separates representation geometry from the additional VAE training budget.

Tab. 23: **Audio representations.** VAE updates count additional training in this study. Native mel does not use a VAE, while Vanilla VAE serves as the common initialization for both VAE fine-tuning experiments. All four F5-TTS models are trained separately for 400k steps.

Representation	VAE updates	Channels	Frames/s	Audio rate
Native mel	—	100	93.75	24 kHz
Vanilla VAE	—	64	40	16 kHz
Vanilla VAE + acoustic fine-tuning	100k	64	40	16 kHz
NA-VAE$_{\text{Acoustic}}$	100k	64	40	16 kHz

Data and temporal alignment. Downstream training uses the same 354,780-utterance manifest from the `train-clean-100`, `train-clean-360`, and `train-other-500` splits of LibriTTS. VAE adaptation uses the 230,774 utterances of at least 3.01 seconds (approximately 492.2 hours), drawing random three-second crops at 16 kHz. This crop-length restriction is not applied to

the downstream manifest. For NA-VAE$_{\text{Acoustic}}$, we use the 1,024-dimensional representations from layer 23 of a frozen WavLM-Large teacher (Chen et al., 2022) as alignment targets. We normalize each valid input waveform using the teacher's feature extractor and linearly interpolate its valid feature sequence to the VAE frame count. Interpolation excludes padding, and alignment averages only over valid frames.

Audio NA-VAE training. We jointly update the encoder, decoder, and a randomly initialized 1-Dimensional alignment head, while keeping WavLM frozen. The head follows the time-conditioned transformer design of DiT/SiT (Peebles & Xie, 2023; Ma et al., 2024), with eight blocks, width 768, 12 attention heads, MLP ratio of 2, query–key normalization, timestep conditioning, and fixed sinusoidal temporal positional embeddings. A three-layer SiLU projector with hidden width 2,048 maps its output to the 1,024-dimensional teacher features. This temporal head replaces the spatial head used for images; no downstream F5-TTS model or diffusion loss is used during VAE adaptation.

Let $\boldsymbol{z}_b$ be a posterior sample and $\widetilde{\boldsymbol{z}}_b$ its channel-wise normalized version. Normalization is affine-free, with synchronized, detached running statistics, momentum 0.1, and numerical constant 10^{-4}. Only the alignment branch receives $\widetilde{\boldsymbol{z}}_{b,t} = (1-t_b)\widetilde{\boldsymbol{z}}_b + t_b\boldsymbol{\epsilon}_b$, where $t_b \sim \mathrm{U}(0,1)$ and $\boldsymbol{\epsilon}_b \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. For T_b valid latent frames and temporally aligned teacher features $\boldsymbol{y}_{b,\tau}$, the additional objective is

$$\mathcal{L}_{\text{align}} = -\frac{1}{B}\sum_{b=1}^{B}\frac{1}{T_b}\sum_{\tau=1}^{T_b}\cos(h_\omega(\widetilde{\boldsymbol{z}}_{b,t}, t_b)_\tau, \mathrm{sg}[\boldsymbol{y}_{b,\tau}])\,. \tag{9}$$

The decoder reconstructs from $\boldsymbol{z}_b$ before this extra noise and normalization. We preserve the acoustic objective of the released VAE (Niu et al., 2025):

$$\mathcal{L}_{\text{audio}} = 15\mathcal{L}_{\text{mel}} + 2\mathcal{L}_{\text{FM}} + \mathcal{L}_{\text{adv}} + 0.01\mathcal{L}_{\text{KL}}, \tag{10}$$

$$\mathcal{L}_{\text{step}} = \mathcal{L}_{\text{audio}} + 0.5\mathcal{L}_{\text{align}}. \tag{11}$$

Here $\mathcal{L}_{\text{mel}}$ uses window lengths $32, 64, 128, 256, 512, 1024, 2048$ with $5, 10, 20, 40, 80, 160, 320$ mel bins, respectively; $\mathcal{L}_{\text{FM}}$ matches discriminator features, and $\mathcal{L}_{\text{adv}}$ uses least-squares adversarial training (Mao et al., 2017) with the released multi-period and multi-band STFT discriminators (Kumar et al., 2023). KL is summed over the 64 posterior channels and averaged over batch and time. As in §C.1, the alignment gradient contributes with coefficient 1.5 to the encoder and 0.5 to the head: we multiply the encoder-side alignment gradient by three before applying Eq. 11. Vanilla VAE + acoustic fine-tuning uses only $\mathcal{L}_{\text{audio}}$, with the same starting VAE, data, crop sampling, acoustic optimizer, and number of updates. Adaptation uses seed 42 for data ordering and head initialization, with separate posterior and alignment-noise seeds 1234 and 5678.

Tab. 24: **Audio training settings.** The VAE column describes the two 100k continuations; head-specific entries apply only to NA-VAE$_{\text{Acoustic}}$. F5 settings are shared unless the representation is specified.

Setting	VAE adaptation	Downstream F5-TTS
Optimizer	AdamW	AdamW
GPUs / gradient accumulation	8 / 1	8 / 1
Batch size	64 three-second crops	Dynamic; at most 64 utterances/GPU
Frame budget per GPU	—	16,384 (VAE); 38,400 (mel)
Learning rate	2×10^{-5} (AE/D); 10^{-4} (head)	7.5×10^{-5}
Warmup steps	500	20,000
Schedule after warmup	Constant	Linear decay; fixed 686-epoch horizon
Adam (β_1, β_2)	$(0.8, 0.99)$ (AE/D); $(0.9, 0.999)$ (head)	$(0.9, 0.999)$
Adam ϵ	10^{-8}	10^{-8}
Weight decay	0.01 (AE/D); 0 (head)	0.01
Gradient clipping	10 (AE/D); 1 (head)	1
Precision	FP32	FP16
VAE EMA decay	0.9999	Frozen VAE
Training steps reported	100k	400k

Downstream F5-TTS training. After adaptation, we discard the alignment head and freeze the VAE. Each VAE-based system trains on cached posterior samples at the native codec scale; the alignment normalization is not applied to F5-TTS inputs. All systems use F5-TTS Small (Chen et al., 2025e): 18 transformer blocks, width 768, 12 attention heads, MLP ratio 2, text width 512, and four text ConvNeXt V2 blocks (Woo et al., 2023). They share the transcript manifest and the F5 example character vocabulary, without an additional downstream duration filter. Each F5 model is randomly initialized. Training uses conditional flow matching (Lipman et al., 2022) with uniform

time sampling and masks 70–100% of each target sequence. The F5 conditioning-drop parameters are 0.3 for audio and 0.2 for joint text/audio dropout. Both training stages use AdamW (Loshchilov & Hutter, 2019), with settings summarized in Tab. 24.

Dynamic batches are shuffled with seed 666. The two frame budgets each correspond to an upper bound of 409.6 seconds of audio per GPU, before packing and the utterance-count cap; native mel nevertheless has longer sequences and a different decoder, so its computation is not matched to the VAE systems. All F5 runs retain the same learning-rate trajectory and stop at 400k checkpoint steps.

Tab. 25: **Audio generation over the full 400k-step F5 budget.** Columns report F5 training steps; all representations remain frozen. Vanilla VAE + acoustic fine-tuning and NA-VAE$_{\text{Acoustic}}$ each receive 100k VAE adaptation updates. WER is an utterance-mean percentage; higher SIM is better. Bold marks the best displayed value among the four systems for each test set, metric, and checkpoint; shading marks NA-VAE$_{\text{Acoustic}}$. Native mel remains FP16, with Flash attention after 200k.

Representation	Metric	50k	100k	150k	200k	250k	300k	350k	400k
LibriSpeech-PC test-clean ($n = 1,127$)									
Native mel	WER%↓	79.0296	17.1723	7.6551	4.9399	3.8228	3.2524	2.6094	2.5056
	SIM↑	0.3343	0.5661	0.5966	0.6053	0.6090	0.6109	0.6114	0.6102
Vanilla VAE	WER%↓	71.9379	20.5207	7.9877	4.3153	3.3188	2.8261	2.4263	**2.3905**
	SIM↑	0.4107	0.5796	0.6055	0.6127	0.6112	0.6104	0.6063	0.6036
Vanilla VAE + acoustic fine-tuning	WER%↓	63.1557	**10.9880**	**6.1676**	4.5278	3.3500	2.8056	2.5690	2.5102
	SIM↑	0.4057	0.5700	0.5955	0.6050	0.6088	0.6097	0.6077	0.6066
NA-VAE$_{\text{Acoustic}}$	WER%↓	**48.4135**	14.7759	6.7663	**4.2309**	**3.2356**	**2.7542**	**2.3551**	2.4178
	SIM↑	**0.4632**	**0.6049**	**0.6234**	**0.6264**	**0.6245**	**0.6227**	**0.6209**	**0.6200**
Seed-TTS-Eval English ($n = 1,088$)									
Native mel	WER%↓	82.7524	17.7769	8.8991	5.5699	4.4542	3.6312	2.8572	2.6655
	SIM↑	0.3903	0.5485	0.5734	0.5843	0.5921	0.5955	0.5982	0.5981
Vanilla VAE	WER%↓	77.5611	19.5540	**7.7641**	**4.4358**	**3.3980**	**2.8164**	2.9697	2.5977
	SIM↑	0.4304	0.5573	0.5771	0.5798	0.5804	0.5828	0.5819	0.5810
Vanilla VAE + acoustic fine-tuning	WER%↓	69.3110	**13.0523**	8.4538	6.3679	5.0628	4.1362	3.5297	3.1754
	SIM↑	0.4336	0.5498	0.5737	0.5850	0.5882	0.5914	0.5912	0.5900
NA-VAE$_{\text{Acoustic}}$	WER%↓	**54.8978**	18.1562	8.5315	5.0562	3.6438	3.0444	**2.6417**	**2.4627**
	SIM↑	**0.4811**	**0.5785**	**0.5922**	**0.5975**	**0.5980**	**0.6001**	**0.6001**	**0.5998**

Generation evaluation. We evaluate LibriSpeech-PC test-clean (Panayotov et al., 2015; Chen et al., 2025e) (1,127 cross-sentence prompt–target pairs) and Seed-TTS-Eval English (Anastassiou et al., 2024) (1,088 pairs). Sampling uses F5 EMA weights in FP32, 32 Euler steps, and classifier-free guidance (Ho & Salimans, 2022) with the F5 parameter `cfg_strength`=2. We use sway sampling (Chen et al., 2025e) with coefficient -1 and seed 1. VAE prompts use posterior means, whereas native mel uses mel prompts. We follow the F5 prompt RMS normalization and UTF-8 text-length duration rule (Chen et al., 2025e), including its trailing-space convention, without using the target waveform's duration. Native-mel outputs are decoded at 24 kHz and resampled to 16 kHz for scoring; all other outputs are native 16 kHz.

WER is measured with Whisper large-v3 (Radford et al., 2023) through faster-whisper (English, beam size 5, FP16 ASR inference), using the F5 punctuation/case normalization and averaging word-error rates over utterances. SIM is cosine similarity between generated and prompt speaker embeddings from the WavLM-Large (Chen et al., 2022) / ECAPA-TDNN (Desplanques et al., 2020) evaluator used by F5-TTS. Each checkpoint is evaluated on both complete test sets; the same prompts, target texts, and scoring code are used throughout.

Generation results. Tab. 25 compares all four systems at 50k-step intervals from 50k to 400k F5 training steps, with final results in the last column. NA-VAE$_{\text{Acoustic}}$ has the highest SIM at every evaluated checkpoint on both test sets. At 400k, it improves SIM over Vanilla VAE + acoustic fine-tuning by 0.0134 on LibriSpeech-PC and 0.0099 on Seed English, while reducing Seed WER from 3.1754% to 2.4627% (22.4% relative). Its final LibriSpeech WER is close to that of the released Vanilla VAE (2.4178% versus 2.3905%).

Reconstruction evaluation. We evaluate the three waveform autoencoders on complete utterances from LibriSpeech test-clean (Panayotov et al., 2015) (2,620), AISHELL-2 test iOS (Du et al., 2018) (5,000), and LibriTTS test-other (Zen et al., 2019) (5,120). Inputs are converted to mono 16 kHz audio, reconstructed in FP32 from posterior samples with fixed utterance-dependent seed 1, and trimmed to the original length. We use the released non-EMA Vanilla weights and EMA weights for both 100k continuations. The alignment head and its additional noise are absent at reconstruction time. We report PESQ (Rix et al., 2001) in wideband mode at 16 kHz, following ITU-T P.862.2 (ITU-T, 2007), and in narrowband mode after resampling to 8 kHz, together with standard STOI (Taal et al., 2011). All three metrics are averaged per utterance. All LibriSpeech and AISHELL-2 samples have valid scores. On LibriTTS, the same five short utterances have undefined PESQ for all three models; PESQ uses the remaining 5,115, while STOI includes all 5,120. The Chinese set evaluates reconstruction only, not Chinese TTS.

Tab. 26: **Full-utterance audio reconstruction.** Higher is better for all three metrics. Bold compares models within each test set; shading marks NA-VAE$_{\text{Acoustic}}$. Native mel is a downstream reference and is not included in this VAE reconstruction comparison.

Test set	VAE	PESQ-WB↑	PESQ-NB↑	STOI↑
LibriSpeech test-clean	Vanilla VAE	3.9721	4.1253	0.980870
	Vanilla VAE + acoustic fine-tuning	**3.9840**	**4.1317**	0.981157
	NA-VAE$_{\text{Acoustic}}$	3.9820	4.1307	**0.981177**
AISHELL-2 test iOS	Vanilla VAE	3.5321	3.8710	0.964119
	Vanilla VAE + acoustic fine-tuning	3.5607	**3.8753**	**0.964936**
	NA-VAE$_{\text{Acoustic}}$	**3.5614**	3.8748	0.964836
LibriTTS test-other	Vanilla VAE	3.7775	4.0304	0.965889
	Vanilla VAE + acoustic fine-tuning	3.8347	4.0634	0.967881
	NA-VAE$_{\text{Acoustic}}$	**3.8367**	**4.0636**	**0.967924**

Reconstruction and comparison scope. Both continuations improve the displayed reconstruction metrics over the released Vanilla VAE. NA-VAE$_{\text{Acoustic}}$ remains close to Vanilla VAE + acoustic fine-tuning: their largest PESQ-WB difference is about 0.0021 and their largest STOI difference is about 10^{-4}. This comparable reconstruction accompanies a clearer separation in downstream speaker similarity, supporting the benefit of noisy semantic alignment beyond continued acoustic training alone.

C.12 Reconstruction Performance

We complement the reconstruction results in Tab. 2 (§4.3) with a common-protocol comparison of NA-VAE and the released SD-VAE, IN-VAE, and VA-VAE tokenizers. The SD-VAE- and VA-VAE-initialized models use the 400k-step checkpoints (80 epochs) evaluated in the main table. For IN-VAE, we use the 100k-step checkpoint (20 epochs); its shorter training budget is reported explicitly.

Tab. 27: **Reconstruction performance under a common evaluation protocol.** Steps denote NA-VAE tokenizer updates, not downstream training; — denotes released checkpoints. Shading marks NA-VAE, and bold marks the best displayed value within each VAE family.

Tokenizer	Latent	Steps	PSNR↑	SSIM↑	LPIPS↓	rFID↓
SD-VAE	*f8d4*	—	**26.9038**	**0.7598**	**0.1305**	0.6937
NA-VAE$_{\text{SD-VAE}}$	*f8d4*	400k	25.7555	0.7373	0.1342	**0.4923**
IN-VAE	*f16d32*	—	**28.9155**	**0.8313**	**0.0850**	**0.2617**
NA-VAE$_{\text{IN-VAE}}$	*f16d32*	100k	28.1893	0.8171	0.0937	0.2957
VA-VAE	*f16d32*	—	**27.6339**	**0.7977**	**0.0980**	0.2803
NA-VAE$_{\text{VA-VAE}}$	*f16d32*	400k	27.4147	0.7893	0.1025	**0.2775**

Evaluation protocol. We reconstruct all 50,000 ImageNet validation images after resizing the shorter side to 256 pixels with antialiased bilinear interpolation and taking a 256×256 center crop. All models use non-EMA weights and deterministic posterior means, with no latent noise or alignment head at inference. The autoencoder forward pass uses BF16; image-level metrics use FP32. PSNR and SSIM are averaged per image on clipped $[0, 1]$ pixels, while LPIPS (Zhang et al., 2018) uses VGG features on clipped $[-1, 1]$ images. For rFID, both originals and reconstructions are rounded to 8-bit RGB before computing the Fréchet distance between their 2,048-dimensional

Inception-v3 feature distributions. All six rows use the same PyTorch FID evaluator; scores from other evaluation implementations are not mixed into this comparison.

Results. Tab. 27 shows lower PSNR and SSIM and higher LPIPS for NA-VAE than for the corresponding released tokenizers. The rFID trade-off is family-dependent: SD-VAE improves from 0.6937 to 0.4923, VA-VAE remains close (0.2803 versus 0.2775), and IN-VAE increases from 0.2617 to 0.2957. Thus, the downstream improvements in the main experiments do not require uniformly better reconstruction scores.

D Extended Related Work

Visual Tokenizers. Visual tokenizers provide a compact interface between images or videos and generative models. Variational autoencoders (Kingma & Welling, 2013) learn continuous latent variables, while VQ-VAE (van den Oord et al., 2017) introduces a learned discrete codebook. VQ-VAE-2 (Razavi et al., 2019) extends this formulation with hierarchical codes, and VQGAN (Esser et al., 2021) combines perceptual and adversarial objectives to learn visually faithful tokens for transformer-based generation. Latent diffusion models (Rombach et al., 2022) establish continuous autoencoder latents as an efficient domain for image synthesis, subsequently adopted by transformer-based generators such as DiT (Peebles & Xie, 2023) and SiT (Ma et al., 2024). Alternative quantizers include finite scalar quantization (Mentzer et al., 2024), which discretizes individual latent dimensions, and MAGVIT-v2's lookup-free quantization (Yu et al., 2024a). TiTok (Yu et al., 2024b) replaces dense two-dimensional grids with compact one-dimensional sequences, while OmniTokenizer (Wang et al., 2024) develops a shared tokenizer for images and videos through joint spatiotemporal modeling. FLUX (Black Forest Labs, 2024) uses continuous autoencoder latents for large-scale image generation. Subsequent work advances both compression and optimization: DC-AE (Chen et al., 2025d) increases spatial compression through residual autoencoding and staged high-resolution adaptation; SimVQ (Zhu et al., 2025) reparameterizes codes with a shared linear transformation; and IBQ (Shi et al., 2025) propagates gradients to all codebook entries. SoftVQ-VAE (Chen et al., 2025b) uses soft combinations of codewords to construct compact continuous tokens, while FlexTok (Bachmann et al., 2025) supports variable-length, coarse-to-fine token sequences. Qwen-Image (Wu et al., 2025a) further demonstrates continuous tokenization in large-scale image generation. StableVQ (Tang et al., 2026) addresses training instability through a dynamic straight-through estimator, a region-based VQ loss, and separate optimization schedules for the encoder–decoder and codebook.

Improving diffusability of visual tokenizers. A compact latent space with accurate reconstruction need not be easy to model, motivating objectives that explicitly improve its generative utility (Li et al., 2025; Yao et al., 2025; Chen et al., 2025a). ImageFolder (Li et al., 2025) separates semantic and pixel-detail information into complementary quantization branches, while TokenFlow (Qu et al., 2025) connects semantic and pixel-level codebooks through shared indices for unified understanding and generation. VA-VAE (Yao et al., 2025) aligns clean latents with vision-foundation-model features, and MAETok (Chen et al., 2025a) uses masked modeling to learn semantically structured representations. EQ-VAE (Kouzelis et al., 2025a) imposes equivariance through reconstruction from spatially transformed latents, while AF-LDM (Zhou et al., 2025) targets fractional-shift equivariance in both the autoencoder and diffusion model. Skorokhodov et al. (2025) connect poor diffusability to excessive high-frequency latent information and introduce scale-equivariant regularization to encourage a more favorable spectral structure. UniTok (Ma et al., 2025a) combines reconstruction and contrastive learning with multi-codebook quantization, while l-DeTok (Yang et al., 2025) combines image-patch masking with interpolative latent noise. RAE (Zheng et al., 2025) directly pairs a frozen visual encoder with a learned image decoder. VQRAE (Du et al., 2025) quantizes pretrained visual features and fine-tunes the encoder with self-distillation, retaining continuous representations for understanding alongside discrete tokens for generation and reconstruction. Visual Generation Tuning (Guo et al., 2025) adapts VLM encoders through semantic-preserving reconstruction, followed by latent normalization and noise regularization, to support continuous autoregressive generation. FAE (Gao et al., 2025) compresses pretrained features into lower-dimensional latents and reconstructs both features and images. More recent work broadens these directions: VTP (Yao et al., 2026) studies scalable tokenizer pretraining by combining reconstruction, self-supervised learning, and image–text contrastive objectives; Qwen-Image-VAE-2.0 (Zhang et al., 2026b) combines semantic alignment with high-compression training and fine-grained reconstruction, including text;

and RAEv2 (Singh et al., 2026b) shows that representation-based latent spaces remain complementary to downstream feature alignment.

Representation alignment for generation. Würstchen (Pernias et al., 2024) generates compact semantic maps to guide image synthesis, and representation-conditioned generation (RCG) (Li et al., 2024b) first samples self-supervised representations and then conditions an image generator on them. Fuest et al. (2024) survey the broader connections between diffusion and representation learning. REPA (Yu et al., 2025) instead supplies an auxiliary training target, aligning intermediate features of a diffusion transformer processing noisy latents with clean-image features from a frozen visual encoder. JanusFlow (Ma et al., 2025c) aligns understanding and generation representations within a unified multimodal model. REPA-E (Leng et al., 2025a) extends alignment to joint optimization of the autoencoder and latent diffusion model; its text-to-image extension (Leng et al., 2025b) further studies end-to-end-tuned tokenizers for large-scale generation. Other methods exploit representations through different objectives. Kouzelis et al. (2025b) jointly generate low-level VAE latents and high-level visual features with a diffusion model, allowing the generated semantic features to guide image synthesis. VideoREPA (Zhang et al., 2025) distills spatiotemporal token relations from video foundation models to improve physical plausibility, while Geometry Forcing (Wu et al., 2025b) aligns video-diffusion features with geometry-aware representations for more consistent world modeling. i-REPA (Singh et al., 2026a) subsequently identifies spatial structure as a stronger indicator of alignment effectiveness than global discriminative ability and introduces spatial regularization to improve generation.

E QUALITATIVE RESULTS

This section shows class-conditional samples from the models behind Tab. 2: SiT-XL/2 trained with REPA on a *frozen* NA-VAE$_{\text{SD-VAE}}$, at 256×256 and at 512×512. Figs. 16–23 present samples generated at both resolutions.

Sampling setting. All figures use classifier-free guidance with $w = 4.0$ over the full trajectory, 250 Euler–Maruyama steps, and the EMA weights.



Fig. 16: **Uncurated** samples from one frozen **NA-VAE**: 512×512 (top 2 rows) and 256×256 (bottom row). Classifier-free guidance with $w = 4.0$. Class label = "giant panda" (388).



Fig. 17: **Uncurated** samples from one frozen **NA-VAE**: 512×512 (top 2 rows) and 256×256 (bottom row). Classifier-free guidance with $w = 4.0$. Class label = "balloon" (417).



Fig. 18: **Uncurated** samples from one frozen **NA-VAE**: 512×512 (top 2 rows) and 256×256 (bottom row). Classifier-free guidance with $w = 4.0$. Class label = "castle" (483).



Fig. 19: **Uncurated** samples from one frozen **NA-VAE**: 512×512 (top 2 rows) and 256×256 (bottom row). Classifier-free guidance with $w = 4.0$. Class label = "cheeseburger" (933).

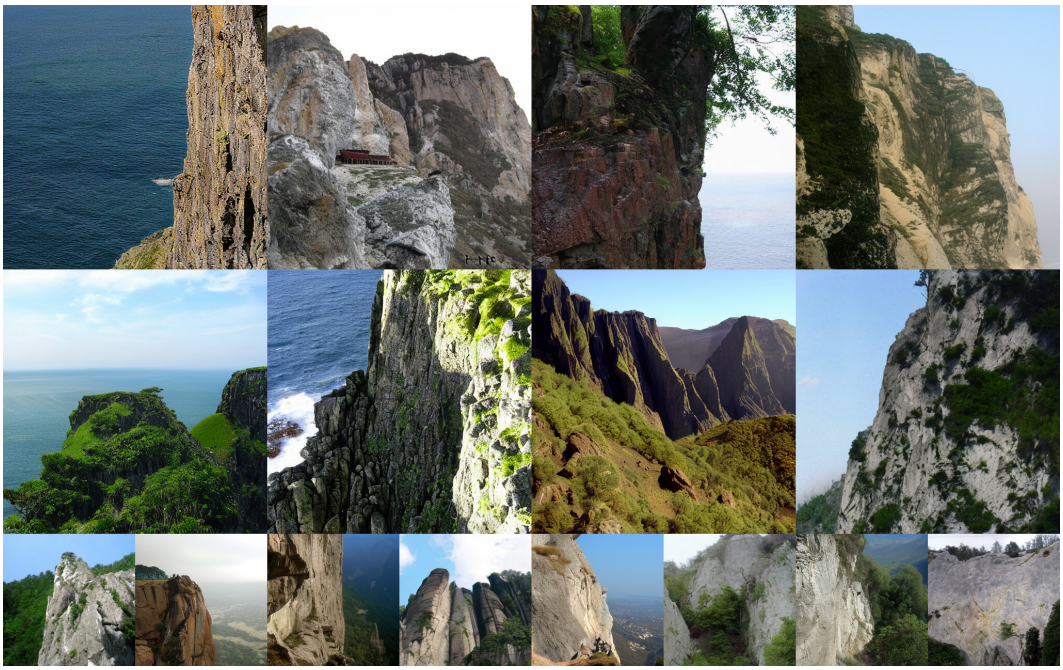


Fig. 20: **Uncurated** samples from one frozen **NA-VAE**: 512×512 (top 2 rows) and 256×256 (bottom row). Classifier-free guidance with $w = 4.0$. Class label = "cliff" (972).



Fig. 21: **Uncurated** samples from one frozen **NA-VAE**: 512×512 (top 2 rows) and 256×256 (bottom row). Classifier-free guidance with $w = 4.0$. Class label = "coral reef" (973).



Fig. 22: **Uncurated** samples from one frozen **NA-VAE**: 512×512 (top 2 rows) and 256×256 (bottom row). Classifier-free guidance with $w = 4.0$. Class label = "lakeside" (975).



Fig. 23: **Uncurated** samples from one frozen **NA-VAE**: 512×512 (top 2 rows) and 256×256 (bottom row). Classifier-free guidance with $w = 4.0$. Class label = "volcano" (980).