

DIFFUSION TRANSFORMERS WITH REPRESENTATION AUTOENCODERS

Boyang Zheng Nanye Ma Shengbang Tong Saining Xie
New York University

ABSTRACT

Latent generative modeling, where a pretrained autoencoder maps pixels into a latent space for the diffusion process, has become the standard strategy for Diffusion Transformers (DiT); however, the autoencoder component has barely evolved. Most DiTs continue to rely on the original VAE encoder, which introduces several limitations: outdated backbones that compromise architectural simplicity, low-dimensional latent spaces that restrict information capacity, and weak representations that result from purely reconstruction-based training and ultimately limit generative quality. In this work, we explore replacing the VAE with pretrained representation encoders (e.g., DINO, SigLIP, MAE) paired with trained decoders, forming what we term Representation Autoencoders (RAEs). These models provide both high-quality reconstructions and semantically rich latent spaces, while allowing for a scalable transformer-based architecture. Since these latent spaces are typically high-dimensional, a key challenge is enabling diffusion transformers to operate effectively within them. We analyze the sources of this difficulty, propose theoretically motivated solutions, and validate them empirically. Our approach achieves faster convergence without auxiliary representation alignment losses. Using a DiT variant equipped with a lightweight, wide DDT head, we achieve strong image generation results on ImageNet: 1.51 FID at 256×256 (no guidance) and 1.13 at both 256×256 and 512×512 (with guidance). RAE offers clear advantages and should be the new default for diffusion transformer training.

Project page: rae-dit.github.io

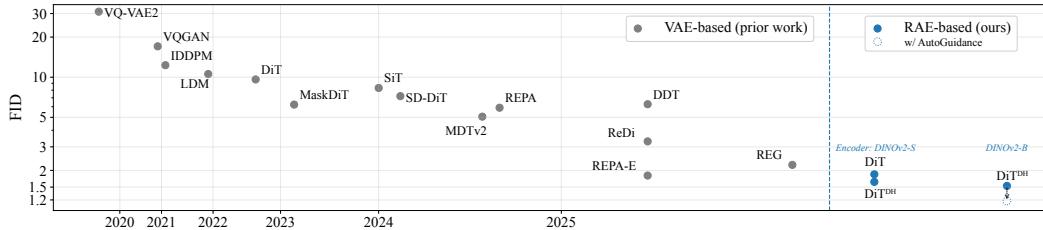


Figure 1: *Representation Autoencoder* (RAE) uses frozen pretrained representations as the encoder with a lightweight decoder to reconstruct input images without compression. RAE enables faster convergence and higher-quality samples in latent diffusion training compared to VAE-based models.

1 INTRODUCTION

The evolution of generative modeling has been driven by a continual redefinition of *where* and *how* models learn to represent data. Early pixel-space models sought to directly capture image statistics, but the emergence of latent diffusion (Vahdat et al., 2021; Rombach et al., 2022) reframed generation as a process operating within a learned, compact representation space. By diffusing in this space rather than in raw pixels, models such as Latent Diffusion Models (LDM) (Rombach et al., 2022) and Diffusion Transformers (DiT) (Peebles & Xie, 2023; Ma et al., 2024) achieve higher visual fidelity and efficiency, powering the most capable image and video generators of today.

Despite progress in diffusion backbones, the autoencoder defining the latent space remains largely unchanged. The widely used SD-VAE (Rombach et al., 2022) still relies on heavy channel-wise

compression and a reconstruction-only objective, producing low-capacity latents that capture local appearance but lack global semantic structure crucial for generalization and generative performance of diffusion models (Song et al., 2025). In addition, SD-VAE, built on a legacy convolutional design, remains computationally inefficient (see Fig. 2).

Meanwhile, visual representation learning has undergone a rapid transformation. Self-supervised and multimodal encoders such as DINO (Oquab et al., 2023), MAE (He et al., 2021), JEPA (Assran et al., 2023) and CLIP / SigLIP (Radford et al., 2021; Tschannen et al., 2025) learn semantically structured visual features that generalize across tasks and scales and provide a natural basis for visual understanding. However, latent diffusion remains largely isolated from this progress, continuing to diffuse in reconstruction-trained VAE spaces rather than semantically meaningful representational ones. Recent work attempts to improve latent quality *indirectly* through REPA-style (Yu et al., 2025; Yao et al., 2025; Leng et al., 2025) alignment with external encoders, but these methods introduce extra training stages, auxiliary losses, and tuning complexity.

This separation stems from long-standing assumptions about the *incompatibility* between semantic and generative objectives. It is widely believed that encoders trained to capture semantics are *unsuited* for faithful reconstruction, since they “*focus on high-level information and can only reconstruct an image with high-level semantic similarities.*” (Yu et al., 2024b) In addition, diffusion models are believed to perform poorly in high-dimensional latent spaces (Skorokhodov et al., 2025; Yao et al., 2025; Esser et al., 2024; Liu et al., 2024), leading practitioners to favor low-dimensional VAE latents over the typically much higher-dimensional representations of semantic encoders. In this work, we show that *both assumptions might be wrong*. We demonstrate that *frozen* representation encoders, even those explicitly optimized for semantics over reconstruction, can be repurposed into powerful autoencoders for generation, yielding reconstructions superior to SD-VAE without architectural complexity or auxiliary losses. Furthermore, we find that diffusion transformer training can be stable and efficient in these higher-dimensional latent spaces. With the right architectural adjustments, higher-dimensional representations are not a liability but an *advantage*, offering richer structure, faster convergence, and better generation quality. Notably, higher-dimensional latents introduce effectively *no extra compute or memory costs* since the token count is fixed (determined by the patch size) and the channels are projected to the DiT hidden dimension in the first layer.

We formalize this insight through *Representation Autoencoders* (RAEs), a new class of autoencoders that replace the VAE with a pretrained representation encoder (e.g., DINO) paired with a trained decoder. RAEs produce latent spaces that are semantically rich, structurally coherent, and *diffusion-friendly*, linking semantic and generative modeling through a shared latent representation.

While feasible in principle, adapting diffusion transformers to these high-dimensional semantic latents requires careful design. Original DiTs, designed for compact SD-VAEs, struggle with the increased dimensionality due to: (1) *Transformer design*: DiTs cannot fit even a single image unless their width exceeds the token dimension, implying width must scale with latent dimensionality; (2) *Noise scheduling*: resolution-based schedule shifts (Chen, 2023; Hoogeboom et al., 2023; Esser et al., 2024), derived from pixel- and VAE-based inputs, neglect token dimensionality, motivating a dimensionality-dependent shift; (3) *Decoder robustness*: unlike VAEs trained on continuous latent distributions (Kingma & Welling, 2014), RAE decoders learn from discretely supported latents but must reconstruct samples from a diffusion model that follow a continuous distribution, which we address by noise-augmented decoder training. Finally, we introduce a new DiT variant, DiT^{DH}, inspired by DDT (Wang et al., 2025c) but motivated by a different design perspective. It augments the standard DiT architecture with a lightweight, *shallow yet wide* head, enabling the diffusion model to scale in width without incurring quadratic computational costs. Empirically, this design further enhances diffusion transformer training in high-dimensional RAE spaces.

Empirically, RAEs demonstrate strong visual generation performance. On ImageNet, our RAE-based DiT^{DH} achieves FIDs of 1.51 at 256×256 without any guidance, and 1.13 at both 256×256 and 512×512 with AutoGuidance (Karras et al., 2025), showing the effectiveness of RAEs as an alternative to conventional VAEs in diffusion transformer training. More broadly, these results reframes autoencoding from a compression mechanism into a representation foundation, one that enables diffusion transformers to train more efficiently and generate more effectively.

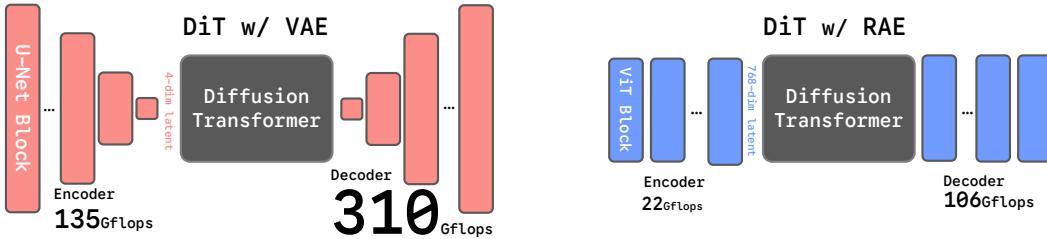


Figure 2: **Comparison of SD-VAE and RAE (DINOv2-B).** The VAE relies on convolutional backbones with aggressive down- and up-sampling, while the RAE uses a ViT architecture *without* compression. SD-VAE is also more computationally expensive, requiring about 6x and 3x more GFLOPs than RAE for the encoder and decoder, respectively. GFlops are evaluated on one 256×256 image.

2 RELATED WORKS

Here, we discuss previous work on the line of representation learning and reconstruction/generation. We present a more detailed related work discussion in Section A.

Representation for Reconstruction. Recent work explores enhancing VAEs with semantic representations: VA-VAE (Yao et al., 2025) aligns VAE latents with a pretrained representation encoder, while MAETok (Chen et al., 2025a), DC-AE 1.5 (Chen et al., 2025d), and l-DEtok (Yang et al., 2025) incorporate MAE- or DAE (Vincent et al., 2008)-inspired objectives into VAE training. Such alignment greatly improves reconstruction and generation performance of VAEs, yet its reliance on heavily compressed, low-dimensional latents still limits both reconstruction fidelity and representation quality. In contrast, we reconstruct directly from representation encoders features without compression. We show that, with a simple ViT decoder on top of frozen representation encoders features, it achieves reconstruction quality comparable to or better than SD-VAE (Rombach et al., 2022), while preserving substantially stronger representations.

Representation for Generation. Recent work also explores using semantic representations to improve generative modeling. REPA (Yu et al., 2025) accelerates DiT convergence by aligning its middle block with representation encoders features. DDT (Wang et al., 2025c) further improves convergence by decoupling DiT into an encoder-decoder and applying REPA loss to the encoder output. REG (Wu et al., 2025) introduces a learnable token into the DiT sequence and explicitly aligns it with a representation encoders representation. ReDi (Kouzelis et al., 2025b) generates both VAE latents and PCA components of DINOv2 features within a diffusion model. In contrast, we train diffusion models directly on representation encoders and achieve faster convergence. We also recommend (Dieleman, 2025) for a broader perspective on this topic.

3 HIGH FIDELITY RECONSTRUCTION FROM FROZEN ENCODERS

In this section, we challenge the common assumption that pretrained representation encoders, such as DINOv2 (Oquab et al., 2023) and SigLIP2 (Tschannen et al., 2025), are unsuitable for the reconstruction task because they “emphasize high-level semantics while downplaying low-level details” (Tang et al., 2025; Yu et al., 2024b). We show that, with a properly trained decoder, frozen representation encoders can in fact serve as strong encoders for the diffusion latent space. Our **Representation Autoencoders (RAE)** pair frozen, pretrained representation encoders with a ViT-based decoder, yielding reconstructions on par with or even better than SD-VAE. More importantly, RAEs alleviate the fundamental limitations of VAEs (Kingma & Welling, 2014), whose heavily compressed latent space (e.g. SD-VAE maps 256^2 images to $32^2 \times 4$ (Esser et al., 2021; Rombach et al., 2022) latents) restricts reconstruction fidelity and more importantly, representation quality.

Our training recipe for the RAE decoder is as follows. Given an input $\mathbf{x} \in \mathbb{R}^{3 \times H \times W}$ and the frozen representation encoder E with patch size p_e and hidden size d , we obtain $N = HW/p_e^2$ tokens with channel d . A ViT decoder D with patch size p_d maps them back to pixels with shape $3 \times H \frac{p_d}{p_e} \times W \frac{p_d}{p_e}$; By default we use $p_d = p_e$, so the reconstruction matches the input resolution. For all experiments on 256×256 images, the encoder produces 256 tokens, matching the token count of most prior DiT-based models trained with SD-VAE latents (Peebles & Xie, 2023; Yu et al., 2025; Ma et al., 2024). The decoder D is trained with a combination of L1, LPIPS (Zhang et al., 2018),

Model	rFID	Decoder	rFID	GFLOPs	Encoder	rFID	Model	Top-1 Acc.
DINOv2-B	0.49	ViT-B	0.58	22.2	DINOv2-S	0.52	DINOv2-B	84.5
SigLIP2-B	0.53	ViT-L	0.50	78.1	DINOv2-B	0.49	SigLIP2-B	79.1
MAE-B	0.16	ViT-XL	0.49	106.7	DINOv2-L	0.52	MAE-B	68.0
SD-VAE	0.62	SD-VAE	0.62	310.4			SD-VAE	8.0

(a) **Encoder choice.** All encoders outperform SD-VAE.
(b) **Larger decoders** improve rFID while remaining much more efficient than VAEs.
(c) **Encoder scaling.** rFID is stable across RAE sizes.
(d) **Representation quality.** RAEs have much higher linear probing accuracy than VAEs.

Table 1: RAEs consistently outperform SD-VAE in reconstruction (rFID) and representation quality (linear probing accuracy) on ImageNet-1K, while being more efficient. If not specified, we use ViT-XL as the decoder and DINOv2-B as the encoder for RAE. Default settings in this paper are in gray.

and adversarial losses (Goodfellow et al., 2014), following common practice in VAEs:

$$z = E(x), \hat{x} = D(z)$$

$$\mathcal{L}_{rec}(x) = \omega_L \text{LPIPS}(\hat{x}, x) + \text{L1}(\hat{x}, x) + \omega_G \lambda \text{GAN}(\hat{x}, x),$$

We provide implementation details about the decoder architecture, hyperparameters, coefficients, and GAN training details in Appendix C.

We select three representative encoders from different pretraining paradigms: DINOv2-B (Oquab et al., 2023) ($p_e=14, d=768$), a self-supervised self-distillation model; SigLIP2-B (Tschanne et al., 2025) ($p_e=16, d=768$), a language-supervised model; and MAE-B (He et al., 2021) ($p_e=16, d=768$), a masked autoencoder. For DINOv2, we also study different model sizes S,B,L ($d=384,768,1024$). Unless otherwise specified, we use an ViT-XL decoder for all RAEs. We use FID score (Heusel et al., 2017) computed on the reconstructed ImageNet (Russakovsky et al., 2015) validation set as our main metric for reconstruction quality, denoted as rFID.

Reconstruction, scaling, and representation. As shown in Table 1a, RAEs with frozen encoders achieve consistently better reconstruction quality (rFID) than SD-VAE. For instance, RAE with MAE-B/16 reaches an rFID of 0.16, clearly outperforming SD-VAE and challenging the assumption that representation encoders cannot recover pixel-level detail.

We next study the scaling behavior of both encoders and decoders. As shown in Table 1c, reconstruction quality remains stable across DINOv2-S, B, and L, indicating that even small representation encoders models preserve sufficient low-level detail for decoding. On the decoder side (Table 1b), increasing capacity consistently improves rFID: from 0.58 with ViT-B to 0.49 with ViT-XL. Importantly, ViT-B already outperforms SD-VAE while being 14× more efficient in GFLOPs, and ViT-XL further improves quality at only one-third of SD-VAE’s cost.

We also evaluate representation quality via linear probing on ImageNet-1K in Table 1d. Because RAEs use frozen pretrained encoders, they directly inherit the representation of the underlying representation encoders. Since RAEs use frozen pretrained encoders, they retain the strong representations of the underlying representation encoders. In contrast, SD-VAE achieves only ∼8% accuracy.

4 TAMING DIFFUSION TRANSFORMERS FOR RAE

With RAE demonstrating good reconstruction quality, we now proceed to investigate the *diffusability* (Skorokhodov et al., 2025) of its latent space; that is, how easily its latent distribution can be modeled by a diffusion model, and how good the generation performance can be.

Before turning to generation, we fix the encoder to study generation capabilities. Table 1a shows that MAE, SigLIP2, and DINOv2 all achieve lower reconstruction rFID than SD-VAE, with MAE the best among them. However, reconstruction alone does not determine generation quality. Empirically, DINOv2 produces the strongest generation results, so unless otherwise noted, we adopt DINOv2 as our default encoder and defer the full comparison to Section G.1.

Following standard practice, we adopt the flow matching objective (Lipman et al., 2023; Liu et al., 2023) with linear interpolation $\mathbf{x}_t = (1 - t)\mathbf{x} + t\boldsymbol{\varepsilon}$, where $\mathbf{x} \sim p(\mathbf{x})$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I})$, and train the model to predict the velocity $v(\mathbf{x}_t, t)$ (see Section J). We use LightningDiT (Yao et al., 2025), a variant of DiT (Peebles & Xie, 2023), as our model backbone. We adopt a patch size of 1, which results in a sequence length of 256 for all RAEs on 256×256 images, matching the token

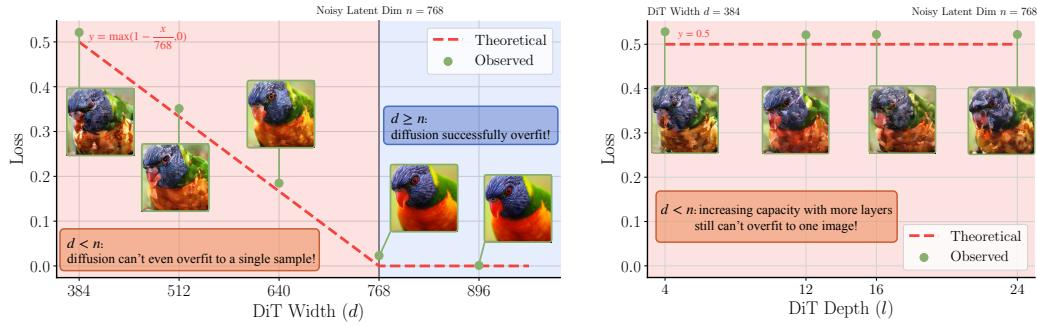


Figure 3: **Overfitting to a single sample.** Left: increasing model width lead to lower loss and better sample quality; Right: changing model depth has marginal effect on overfitting results.

length used by VAE-based DiTs (Peebles & Xie, 2023; Yu et al., 2025; Yao et al., 2025). Since the computational cost of DiT depends primarily on the sequence length, **using DiT on RAE therefore effectively incurs no additional overhead compared to its VAE-based counterparts**. We evaluate our models using FID computed on 50K samples generated with 50 steps with the Euler sampler (denoted as gFID), and all quantitative results are trained for 80 training epochs on ImageNet at 256×256 unless otherwise specified. More training details are included in Section D.

DiT does not work out of the box. To our surprise, the standard diffusion recipe fails with RAE (see Table 2). Training directly on RAE latents causes a small backbone such as DiT-S to completely fail, while a larger backbone like DiT-XL significantly underperforms its counterpart with the SD-VAE latents.

To investigate this observation, we raise several hypotheses detailed below, which we will discuss in the following sections:

	RAE	SD-VAE
DiT-S	215.76	51.74
DiT-XL	23.08	7.13

Table 2: **Standard DiT struggles to model RAE’s latent distribution.**

- **Suboptimal design for diffusion transformers.** When modeling high-dimensional RAE tokens, the optimal design choices for diffusion transformers can diverge from those of the standard DiT, which was originally tailored for low-dimensional VAE tokens.
- **Suboptimal noise scheduling.** Prior noise scheduling and loss re-weighting tricks are derived for image-based or VAE-based input, and it remains unclear if they transfer well to high-dimension semantic tokens.
- **Diffusion generates noisy latents.** VAE decoders are trained to reconstruct images from noisy latents, making them more tolerant to small noises in diffusion outputs. In contrast, RAE decoders are trained on only clean latents and may therefore struggle to generalize.

4.1 SCALING DiT WIDTH TO MATCH TOKEN DIMENSIONALITY

To better understand the training dynamics of diffusion transformers with RAE latents, we first construct a simplified experiment. Rather than training on the entire ImageNet, we randomly select **a single image**, encode it by RAE, and test whether the diffusion model can *reconstruct* it.

Table 2 shows that although RAE underperforms SD-VAE, DiT performance improves with increased capacity. To dissect this effect, we vary model width while fixing depth. Using a DiT-S, we increase the hidden dimension from 384 to 784. As shown in Figure 3, sample quality is poor when the model width $d < \text{token dimension } n = 768$, but improves sharply and reproduces the input almost perfectly once $d \geq n$. Training losses exhibit the same trend, converging only when $d \geq n$.

One might suspect that this improvement still arises from the larger model capacity. To disentangle this effect, we fix the width at $d = 384$ and vary the depth of the SiT-S model. As shown in Figure 3, even when doubling the depth from 12 to 24, the generated images remain artifact-heavy, and the training losses shown in Figure 3 fail to converge to similar level of $d = 768$.

Together, the results indicate that for generation in RAE’s latent space to succeed, **the diffusion model’s width must match or exceed the RAE’s token dimension**. At first glance, this appears to contradict the common belief that data manifolds have low intrinsic dimensionality (Pope et al., 2021), allowing generative models such as GANs to operate effectively within that manifold without scaling to the full data dimension (Sauer et al., 2022). We argue that this contradiction arises from the formulation of diffusion models (Section J): injecting Gaussian noise directly to the data (e.g., in the construction of x_t) throughout training effectively extends the data distribution’s support to the entire space, thereby “diffusing” the data manifold into a full-rank one, requiring model capacity that scales with the full data dimensionality.

In the following, we provide a theoretical justification for this conjecture:

Theorem 1. *Assuming $\mathbf{x} \sim p(\mathbf{x}) \in \mathbb{R}^n$, $\varepsilon \sim \mathcal{N}(0, \mathbf{I}_n)$, $t \in [0, 1]$. Let $\mathbf{x}_t = (1 - t)\mathbf{x} + t\varepsilon$, consider the function family*

$$\mathcal{G}_d = \{g(\mathbf{x}_t, t) = \mathbf{B}f(\mathbf{A}\mathbf{x}_t, t) : \mathbf{A} \in \mathbb{R}^{d \times n}, \mathbf{B} \in \mathbb{R}^{n \times d}, f : [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d\} \quad (1)$$

where $d < n$, f refers to a stack of standard DiT blocks whose width is smaller than the token dimension from the representation encoder, and \mathbf{A}, \mathbf{B} denote the input and output linear projections, respectively. Then for any $g \in \mathcal{G}_d$,

$$\mathcal{L}(g, \theta) = \int_0^1 \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}), \varepsilon \sim \mathcal{N}(0, \mathbf{I}_n)} [\|g(\mathbf{x}_t, t) - (\varepsilon - \mathbf{x})\|^2] dt \geq \sum_{i=d+1}^n \lambda_i \quad (2)$$

where λ_i are the eigenvalues of the covariance matrix of the random variable $W = \varepsilon - \mathbf{x}$.

Notably, when $d \geq n$, \mathcal{G}_d contains the unique minimizer to $\mathcal{L}(g, \theta)$.

Proof. See Section B.1. □

In our toy setting where $p(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_0)$, we have $W \sim \mathcal{N}(-\mathbf{x}, \mathbf{I}_n)$ and $\lambda_i = 1$ for all i . Thus by Theorem 1, the lower bound of the average loss becomes $\tilde{\mathcal{L}}(g, \theta) \geq \frac{1}{n} \sum_{i=d+1}^n 1 = \frac{n-d}{n}$. As shown in Figure 3, this theoretical bound is consistent with our empirical results.

We further extend our investigation to a more practical setting by examining three models of varying width, {DiT-S, DiT-B, DiT-L}. Each model is overfit on a single image encoded by {DINOv2-S, DINOv2-B, DINOv2-L}, respectively, corresponding to different token dimensions. As shown in Table 3, convergence occurs only when the model width is at least as large as the token dimension (e.g., DiT-B with DINOv2-B), while the loss fails to converge otherwise (e.g., DiT-S with DINOv2-B).

	DiT-S	DiT-B	DiT-L
DINOv2-S	3.6e-2 ✓	1.0e-3 ✓	9.7e-4 ✓
DINOv2-B	5.2e-1 ✗	2.4e-2 ✓	1.3e-3 ✓
DINOv2-L	6.5e-1 ✗	2.7e-1 ✗	2.2e-2 ✓

Table 3: **Overfitting losses.** Compared between different combinations of model width and token dimension.

- **Suboptimal design for diffusion transformers.** We now fix the width of DiT to be at least as large as the RAE token dimension. For RAE with the DINOv2-B encoder, we pair it with DiT-XL in our following experiments.

4.2 DIMENSION-DEPENDENT NOISE SCHEDULE SHIFT

Many prior works (Teng et al., 2023; Chen, 2023; Hoogeboom et al., 2023; Esser et al., 2024) have observed that, for inputs $\mathbf{z} \in \mathbb{R}^{C \times H \times W}$, increasing the spatial resolution ($H \times W$) reduces information corruption at the same noise level, impairing diffusion training. These findings, however, are based mainly on pixel- or VAE-encoded inputs with few channels (e.g., $C \leq 16$). In practice, the Gaussian noise is applied to both spatial and channel dimensions; as the number of channels increases, the effective “resolution” per token also grows, reducing information corruption further. We therefore argue that proposed resolution-dependent strategies in these prior works should be generalized to the *effective data dimension*, defined as the number of tokens times their dimensionality.

We adopt the shifting strategy of [Esser et al. \(2024\)](#): for a schedule $t_n \in [0, 1]$ and input dimensions n, m , the shifted timestep is defined as $t_m = \frac{\alpha t_n}{1 + (\alpha - 1)t_n}$ where $\alpha = \sqrt{m/n}$ is a dimension-dependent scaling factor. We follow ([Esser et al., 2024](#)) in using $n = 4096$ as the base dimension and set m to the effective data dimension of RAE. As shown in Table 4, this yields significant performance gains, showing its importance for training diffusion models in the high-dimensional RAE latent space.

	gFID
w/o shift	23.08
w/ shift	4.81

Table 4: **Impact of schedule shift.**

- **Suboptimal noise scheduling.** *We now default the noise schedule to be dependent on the effective data dimension for all our following experiments.*

4.3 NOISE-AUGMENTED DECODING

Unlike VAEs, where latent tokens are encoded as a continuous distribution $\mathcal{N}(\mu, \sigma^2 \mathbf{I})$ ([Kingma & Welling, 2014](#)), the RAE decoder D is trained to reconstruct images from the discrete distribution $p(\mathbf{z}) = \sum_i \delta(\mathbf{x} - \mathbf{z}_i)$, where $\{\mathbf{z}_i\}$ denotes the training set processed by the RAE encoder E . At inference time, however, the diffusion model may generate latents that are noisy or deviate slightly from the training distribution due to imperfect training and sampling [Abuduweili et al. \(2024\)](#). This could introduce a notable out-of-distribution challenge for D , which degrades sampling quality.

To mitigate this issue, inspired by prior works on Normalizing Flows ([Dinh et al., 2017](#); [Ho et al., 2019](#); [Zhai et al., 2025](#)), we augment the RAE decoder training with an additive noise $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. Concretely, rather than decoding directly from the clean latent distribution $p(\mathbf{z})$, we train D on a smoothed distribution $p_{\mathbf{n}}(\mathbf{z}) = \int p(\mathbf{z} - \mathbf{n}) \mathcal{N}(0, \sigma^2 \mathbf{I})(\mathbf{n}) d\mathbf{n}$ to enhance the decoder’s generalization to the denser output space of diffusion models. We further introduce stochasticity into σ by sampling it from $[\mathcal{N}(0, \tau^2)]$, which helps regularize training and improve robustness.

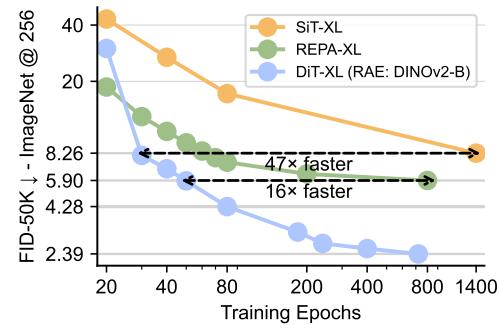
	gFID	rFID
$\mathbf{z} \sim p(\mathbf{z})$	4.81	0.49
$\mathbf{z} \sim p_{\mathbf{n}}(\mathbf{z})$	4.28	0.57

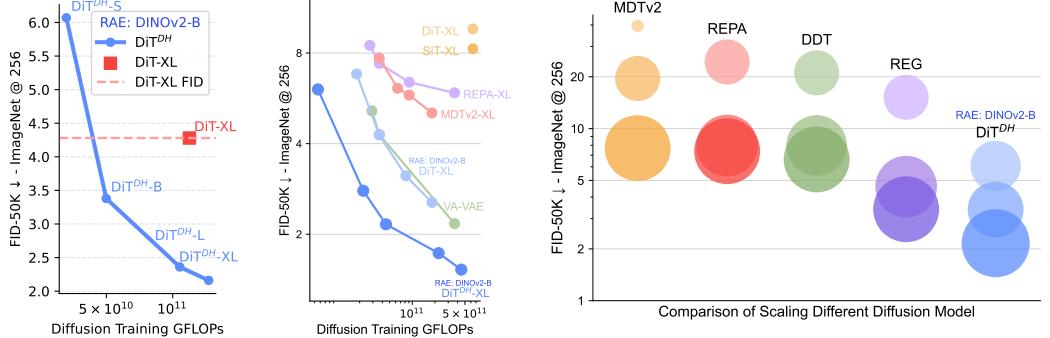
Table 5: **Impact of $p_{\mathbf{n}}(\mathbf{z})$.**

We analyze how $p_{\mathbf{n}}(\mathbf{z})$ affects reconstruction and generation. As shown in Table 5, it improves gFID but slightly worsens rFID. This trade-off is expected: adding noise smooths the latent distribution and, therefore, helps reduce OOD issues for the decoder, but also removes fine details, lowering reconstruction quality. We conduct more ablation experiments on τ and different encoders in Section G.2.

- **Diffusion generates noisy latents.** *We now adopt the noise-augmented decoding for all our following experiments.*

We combine all of the above techniques to train a DiT-XL model on RAE latents. Our improved diffusion recipe achieves a gFID of **4.28** (Figure 4) after only 80 epochs and **2.39** after 720 epochs in RAE’s latent space. With same model size, this not only surpasses prior diffusion baselines (e.g., SiT-XL ([Ma et al., 2024](#))) trained on VAE latents (achieving a $47\times$ training speedup), but also outperforms the convergence speed of recent methods based on representation alignment (e.g., REPA-XL ([Yu et al., 2025](#))), achieving a $16\times$ training speedup. In the following sections, we investigate ways to make RAE generation more efficient and effective, pushing it toward state-of-the-art performance.

Figure 4: **DiT w/ RAE reaches much faster convergence and better FID than SiT or REPA.**



(a) DiT^{DH} scales much better than DiT with RAE latents.
(b) DiT^{DH} with RAE converges faster than VAE-based methods.
(c) DiT^{DH} with RAE reaches better FID than VAE-based methods at all model scales. Bubble area indicates the flops of the model.

Figure 6: **Scalability of DiT^{DH} .** With RAE latents, DiT^{DH} scales more efficiently in both training compute and model size than RAE-based DiT and VAE-based methods.

5 IMPROVING THE MODEL SCALABILITY WITH WIDE DIFFUSION HEAD

As discussed in Section 4, within the standard DiT framework, handling higher-dimensional RAE latents requires scaling up the width of the entire backbone, which quickly becomes computationally expensive. To overcome this limitation, we draw inspiration from DDT (Wang et al., 2025c) and introduce the DDT head—a *shallow yet wide* transformer module dedicated to denoising. By attaching this head to a standard DiT, we effectively increase model width without incurring quadratic growth in FLOPs. We refer to this augmented architecture as DiT^{DH} throughout the remainder of the paper. We also conduct experiment of the design choice of DDT head in Section G.3

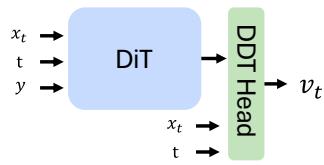


Figure 5: **The Wide DDT Head.**

Wide DDT head. Formally, a DiT^{DH} model consists of a base DiT M and an additional wide, shallow transformer head H . Given a noisy input x_t , timestep t , and an optional class label y , the combined model predicts the velocity v_t as

$$z_t = M(x_t | t, y), \\ v_t = H(x_t | z_t, t),$$

DiT^{DH} converges faster than DiT. We train a series of DiT^{DH} models with varying backbone sizes ($\text{DiT}^{\text{DH}}\text{-S}$, B , L , and XL) on RAE latents. We use a 2-layer, 2048-dim DDT head for all DiT^{DH} models. Performance is compared against the standard DiT-XL baseline. As shown in Figure 6a, DiT^{DH} is substantially more FLOP-efficient than DiT. For example, $\text{DiT}^{\text{DH}}\text{-B}$ requires only ~40% of the training FLOPs yet outperforms DiT-XL by a large margin; when scaled to $\text{DiT}^{\text{DH}}\text{-XL}$ under a comparable training budget, DiT^{DH} achieves an FID of 2.16—nearly half that of DiT-XL.

DiT^{DH} maintains its advantage across RAE scales. We compare $\text{DiT}^{\text{DH}}\text{-XL}$ and DiT-XL on three RAE encoders—DINOv2-S, DINOv2-B, and DINOv2-L. As shown in Table 6, DiT^{DH} consistently outperforms DiT, and the advantage grows with encoder size. For example, with DINOv2-L, DiT^{DH} improves FID from 6.09 to 2.73. We attribute this robustness to the DDT head. Larger encoders produce higher-dimensional latents, which amplify the width bottleneck of DiT. DiT^{DH} addresses this by satisfying the width requirement discussed in Section 4 while keeping features compact. It also filters out noisy information that becomes more prevalent in high-dimensional RAE latents.

Table 6: **DiT^{DH} outperforms DiT across RAE encoder sizes.**

Method	Epochs	#Params	Generation@256 w/o guidance				Generation@256 w/ guidance			
			gFID↓	IS↑	Prec.↑	Rec.↑	gFID↓	IS↑	Prec.↑	Rec.↑
Autoregressive										
VAR (Tian et al., 2024)	350	2.0B	1.92	323.1	0.82	0.59	1.73	350.2	0.82	0.60
MAR (Li et al., 2024b)	800	943M	2.35	227.8	0.79	0.62	1.55	303.7	0.81	0.62
xAR (Ren et al., 2025)	800	1.1B	-	-	-	-	1.24	301.6	0.83	0.64
Pixel Diffusion										
ADM (Dhariwal & Nichol, 2021)	400	554M	10.94	101.0	0.69	0.63	3.94	215.8	0.83	0.53
RIN (Jabri et al., 2023)	480	410M	3.42	182.0	-	-	-	-	-	-
PixelFlow (Chen et al., 2025e)	320	677M	-	-	-	-	1.98	282.1	0.81	0.60
PixNerd (Wang et al., 2025b)	160	700M	-	-	-	-	2.15	297.0	0.79	0.59
SiD2 (Hoogeboom et al., 2025)	1280	-	-	-	-	-	1.38	-	-	-
Latent Diffusion with VAE										
DiT (Peebles & Xie, 2023)	1400	675M	9.62	121.5	0.67	0.67	2.27	278.2	0.83	0.57
MaskDiT (Zheng et al.)	1600	675M	5.69	177.9	0.74	0.60	2.28	276.6	0.80	0.61
SiT (Ma et al., 2024)	1400	675M	8.61	131.7	0.68	0.67	2.06	270.3	0.82	0.59
MDTv2 (Gao et al., 2023)	1080	675M	-	-	-	-	1.58	314.7	0.79	0.65
VA-VAE (Yao et al., 2025)	80	675M	4.29	-	-	-	-	-	-	-
	800	675M	2.17	205.6	0.77	0.65	1.35	295.3	0.79	0.65
REPA (Yu et al., 2025)	80	675M	7.90	122.6	0.70	0.65	-	-	-	-
	800	675M	5.78	158.3	0.70	0.68	1.29	306.3	0.79	0.64
DDT (Wang et al., 2025c)	80	675M	6.62	135.2	0.69	0.67	1.52	263.7	0.78	0.63
	400	675M	6.27	154.7	0.68	0.69	1.26	310.6	0.79	0.65
REPA-E (Leng et al., 2025)	80	675M	3.46	159.8	0.77	0.63	1.67	266.3	0.80	0.63
	800	675M	1.70	217.3	0.77	0.66	1.15	304.0	0.79	0.66
Latent Diffusion with RAE (Ours)										
DiT-XL (DINOv2-S)	800	676M	1.87	209.7	0.80	0.63	1.41	309.4	0.80	0.63
	20		3.71	198.7	0.86	0.50	-	-	-	-
DiT ^{DH} -XL (DINOv2-B)	80	839M	2.16	214.8	0.82	0.59	-	-	-	-
	800		1.51	242.9	0.79	0.63	1.13	262.6	0.78	0.67

Table 8: **Class-conditional performance on ImageNet 256×256.** RAE reaches an FID of 1.51 without guidance, outperforming all prior methods by a large margin. It also achieves an FID of 1.13 with AutoGuidance (Karras et al., 2025). We identified an inconsistency in the FID evaluation protocol in prior literature and re-ran the sampling process for several baselines. This resulted in higher baseline numbers than those originally reported. Further details are discussed in Section 5.1.

5.1 STATE-OF-THE-ART DIFFUSION TRANSFORMERS

Convergence. We compare the convergence behavior of DiT^{DH}-XL with previous state-of-the-art diffusion models (Peebles & Xie, 2023; Ma et al., 2024; Yu et al., 2025; Gao et al., 2023; Yao et al., 2025) in terms of FID without guidance. In Figure 6b, we show the convergence curve of DiT^{DH}-XL with training epochs/GFLOPs, while baseline models are plotted at their reported final performance. DiT^{DH}-XL already surpasses REPA-XL, MDTV2-XL, and SiT-XL around 5×10^{10} GFLOPs, and by 5×10^{11} GFLOPs it achieves the best FID overall, requiring over $40 \times$ less compute.

Scaling. We compare DiT^{DH} with recent methods of different model at different scales. As shown in Figure 6c, increasing the size of DiT^{DH} consistently improves the FID scores. The smallest model, DiT^{DH}-S, reaches a competitive FID of 6.07, already outperforming the much larger REPA-XL. When scaling from DiT^{DH}-S to DiT^{DH}-B, the FID improves significantly from 6.07 to 3.38, surpassing all prior works of similar or even

Method	Generation@512			
	gFID↓	IS↑	Prec.↑	Rec.↑
BigGAN-deep (Brock et al., 2019)	8.43	177.9	0.88	0.29
StyleGAN-XL (Sauer et al., 2022)	2.41	267.8	0.77	0.52
VAR (Tian et al., 2024)	2.63	303.2	-	-
MAGVIT-v2 (Yu et al., 2024a)	1.91	324.3	-	-
XAR (Ren et al., 2025)	1.70	281.5	-	-
ADM	3.85	221.7	0.84	0.53
SiD2	1.50	-	-	-
DiT	3.04	240.8	0.84	0.54
SiT	2.62	252.2	0.84	0.57
DiffiT (Hatamizadeh et al., 2024)	2.67	252.1	0.83	0.55
REPA	2.08	274.6	0.83	0.58
DDT	1.28	305.1	0.80	0.63
EDM2 (Karras et al., 2024)	1.25	-	-	-
DiT ^{DH} -XL (DINOv2-B)	1.13	259.6	0.80	0.63

Table 7: **Class-conditional performance on ImageNet 512×512 with guidance.** DiT^{DH} with 400-epoch training achieves an strong FID score of 1.13.



Figure 7: **Qualitative samples** from our model trained at 512×512 resolution with AutoGuidance. The RAE-based DiT demonstrates strong diversity, fine-grained detail, and high visual quality.

larger scale. The performance continues to improve with $\text{DiT}^{\text{DH}}\text{-XL}$, setting a new state-of-the-art result of 2.16 at 80 training epochs.

Performance. Finally, we provide a quantitative comparison between $\text{DiT}^{\text{DH}}\text{-XL}$, our most performant model, with recent state-of-the-art diffusion models on ImageNet 256×256 and 512×512 in Table 8 and Table 7. Our method outperforms all prior diffusion models by a large margin, setting new state-of-the-art FID scores of **1.51** without guidance and **1.13** with guidance at 256×256 . On 512×512 , with 400-epoch training, $\text{DiT}^{\text{DH}}\text{-XL}$ further achieves an FID of **1.13** with guidance, surpassing the previous best performance achieved by EDM-2 (1.25).

Following the quantitative results, we present qualitative samples from our best models in Figure 7. These visualizations exhibit both high semantic diversity and fine-grained details comparable to ground-truth ImageNet samples, consistent with the achieved state-of-the-art FID. We provide additional visualization samples in Section M and unconditional generation results in Section L.

Remarks on FID evaluation. To construct the 50,000 samples used for conditional FID evaluation, we note that previous works such as DDT (Wang et al., 2025c), VAR (Tian et al., 2024), MAR (Li et al., 2024b), and xAR (Ren et al., 2025) typically evaluate using exactly 50 images per class, while others employ uniform random sampling across the 1,000 class labels. Although uniform random sampling asymptotically approaches the balanced sampling case, we surprisingly observed that class-balanced sampling consistently achieves around 0.1 lower FID scores. To ensure fair comparison, we re-evaluate several recent methods with accessible checkpoints, such as SiT (Peebles & Xie, 2023), REPA (Yu et al., 2025), REPA-E (Leng et al., 2025) using class-balanced sampling and update their reported scores accordingly. A more detailed comparison is provided in Section E.

6 DISCUSSIONS

6.1 HOW CAN RAE EXTEND TO HIGH-RESOLUTION SYNTHESIS EFFICIENTLY?

A central challenge in generating high-resolution images is that resolution scales with the number of tokens: doubling image size in each dimension requires roughly four times as many tokens. To address this, we let the decoder handle resolution scaling by allowing its patch size patch size p_d

to differ from the encoder patch size p_e . When $p_d = p_e$, the output matches the input resolution; setting $p_d = 2p_e$ produces a $2 \times$ upsampled image, reconstructing a 512×512 image from the same tokens used at 256×256 .

Since the decoder is decoupled from both the encoder and the diffusion process, we can reuse diffusion models trained at 256×256 resolution, simply swapping in an upsampling decoder to produce 512×512 outputs without retraining. As shown in Table 9, this approach slightly increases rFID but achieves competitive gFID, while being $4 \times$ more efficient than quadrupling the number of tokens.

6.2 DOES DiT^{DH} WORK WITHOUT RAE?

In this work, we propose and study RAE and DiT^{DH}. In Section 4, we showed that RAE with DiT already brings substantial benefits, even in the absence of DiT^{DH}. Here, we turn the question around: can DiT^{DH} still provide improvements, without the latent space of RAE?

	VAE	DINOv2-B
DiT-XL	7.13	4.28
DiT ^{DH} -XL	11.70	2.16

Table 10: **Performance on VAE.** DiT^{DH} yields worse FID than DiT, despite using extra compute for the wide DDT head.

Method	#Tokens	gFID \downarrow	rFID \downarrow
Direct	1024	1.13	0.53
Upsample	256	1.61	0.97

Table 9: **Comparison on ImageNet 512×512 .** Decoder upsampling achieves competitive FID compared to direct 512×512 training. Both models are trained for 400 epochs.

To investigate, we train both DiT-XL and DiT^{DH}-XL on SD-VAE latents with a patch size of 2, alongside DINOv2-B for comparison, for 80 epochs, and report unguided FID. As shown in Table 10, DiT^{DH}-XL performs even worse than DiT-XL on SD-VAE, despite the additional computation introduced by the diffusion head. This indicates that the DDT head provides little benefit in low-dimensional latent spaces, and its primary strength arises in high-dimensional diffusion tasks introduced by RAE.

6.3 THE ROLE OF STRUCTURED REPRESENTATION IN HIGH-DIMENSIONAL DIFFUSION?

DiT^{DH} achieves strong performance when paired with the high-dimensional latent space of RAE. This raises a key question: is the structured representation of RAE essential, or would DiT^{DH} work equally well on unstructured high-dimensional inputs such as *raw pixels*?

To evaluate this, we train DiT-XL and DiT^{DH}-XL directly on raw pixels. For 256×256 images with a patch size of 16, the resulting DiT input token dimensionality is $16 \times 16 \times 3 = 768$, matching that of the DINOv2-B latents. We report unguided FID after 80 epochs. As shown in Table 11, DiT^{DH} outperforms DiT on pixels, but both models perform far worse than their counterparts trained on RAE latents. These results demonstrate that high dimensionality alone is not sufficient: the structured representation provided by RAE is crucial for achieving strong performance gains.

	Pixel	DINOv2-B
DiT-XL	51.09	4.28
DiT ^{DH} -XL	30.56	2.16

Table 11: **Comparison on pixel diffusion.** Pixel Diffusion has much worse FID than diffusion on DINOv2-B.

7 CONCLUSION

In this work, we challenge the belief that pretrained representation encoders are too high-dimensional and too semantic for reconstruction or generation. We show that a frozen representation encoder, paired with a lightweight trained decoder, forms an effective Representation Autoencoder (RAE). On this latent space, we train Diffusion Transformers in a stable and efficient way with three added components: (1) match DiT width to the encoder token dimensionality; (2) apply a dimension-dependent shift to the noise schedule; and (3) add decoder noise augmentation so the decoder handles diffusion outputs. We also introduce DiT^{DH}, a shallow-but-wide diffusion transformer head that increases width without quadratic compute. Empirically, RAEs enable strong visual generation: on ImageNet, our RAE-based DiT^{DH}-XL achieves 1.51 FID at 256×256 (without guidance) and 1.13 at both 256×256 and 512×512 (with guidance). We believe RAE latents serve as a strong candidate for training diffusion transformers in the future.

8 ACKNOWLEDGMENTS

The authors would like to thank Sihyun Yu, Jaskirat Singh, Shusheng Yang, Xichen Pan, Chenyu Li, Bingda Tang, Fred Lu, David Fan, Amir Bar, Shangyuan Tong for insightful discussions and feedback on the manuscript. This work was mainly supported by the Google TPU Research Cloud (TRC) program and the Open Path AI Foundation. ST is supported by Meta AI Mentorship Program. SX also acknowledges support from the MSIT IITP grant (RS-2024-00457882) and the NSF award IIS-2443404.

REFERENCES

- Abulikemu Abuduweili, Chenyang Yuan, Changliu Liu, and Frank Permenter. Enhancing sample generation of diffusion models using noise level correction. *TMLR*, 2024. 7
- Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. 20, 27, 28
- 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 *CVPR*, 2023. 2
- Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 9
- Shiyue Cao, Yueqin Yin, Lianghua Huang, Yu Liu, Xin Zhao, Deli Zhao, and Kaigi Huang. Efficient-vqgan: Towards high-resolution image generation with efficient vision transformers. In *ICCV*, 2023. 19
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 22
- 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 *ICML*, 2025a. 3
- Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, Le Xue, Caiming Xiong, and Ran Xu. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset, 2025b. 19
- 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 *ICLR*, 2025c. 19
- Junyu Chen, Dongyun Zou, Wenkun He, Junsong Chen, Enze Xie, Song Han, and Han Cai. Dc-ae 1.5: Accelerating diffusion model convergence with structured latent space, 2025d. 3
- Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020a. 19
- Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025e. 9
- Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023. 2, 6
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020b. 19
- Zhe Chen, Jiannan Wu, Wenhui Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024. 19
- Timothée Dariset, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2025. 22
- Jeffrey De Fauw, Sander Dieleman, and Karen Simonyan. Hierarchical autoregressive image models with auxiliary decoders. *arXiv preprint arXiv:1903.04933*, 2019. 19
- Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021. 9, 24, 27, 28
- Sander Dieleman. Generative modelling in latent space, 2025. URL <https://sander.ai/2025/04/15/latents.html>. 3

- Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *ICLR*, 2017. 7
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 19, 22
- Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 3, 19, 23
- Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 2, 6, 7, 27
- David Fan, Shengbang Tong, Jiachen Zhu, Koustuv Sinha, Zhuang Liu, Xinlei Chen, Michael Rabat, Nicolas Ballas, Yann LeCun, Amir Bar, et al. Scaling language-free visual representation learning. In *ICCV*, 2025a. 19
- Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. In *ICLR*, 2025b. 19
- Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023. 9
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 4, 22
- Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 2020. 19
- Philippe Hansen-Estruch, David Yan, Ching-Yao Chung, Orr Zohar, Jialiang Wang, Tingbo Hou, Tao Xu, Sriram Vishwanath, Peter Vajda, and Xinlei Chen. Learnings from scaling visual tokenizers for reconstruction and generation. In *ICML*, 2025. 19
- Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. DiffiT: Diffusion vision transformers for image generation. In *ECCV*, 2024. 9
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arxiv e-prints, art.* In *CVPR*, 2019. 19
- Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2021. 2, 4, 19, 22
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 4, 28
- Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786), 2006. 19
- Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 27
- Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *ICML*, 2019. 7
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 19, 27
- Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, 2023. 2, 6

- Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion. In *CVPR*, 2025. 9
- Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *ICML*, 2023. 9
- Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 27
- Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *CVPR*, 2024. 9
- Tero Karras, Miika Aittala, Tuomas Kynkänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *NeurIPS*, 2025. 2, 9, 27
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2, 3, 7, 19
- Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Eq-vae: Equivariance regularized latent space for improved generative image modeling. In *ICML*, 2025a. 19
- Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. In *NeurIPS*, 2025b. 3
- Tuomas Kynkänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *NeurIPS*, 2019. 28
- Tuomas Kynkänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *NeurIPS*, 2024. 27
- Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *CVPR*, 2022. 19
- 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 *ICCV*, 2025. 2, 9, 10, 25
- Tianhong Li, Dina Katahi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. In *NeurIPS*, 2024a. 28
- Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *NeurIPS*, 2024b. 9, 10, 19, 24, 25
- Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 4, 27
- Bingchen Liu, Ehsan Akhgari, Alexander Visheratin, Aleks Kamko, Linmiao Xu, Shivam Shrirao, Chase Lambert, Joao Souza, Suhail Doshi, and Daiqing Li. Playground v3: Improving text-to-image alignment with deep-fusion large language models, 2024. 2
- Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 4, 27
- 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 *ECCV*, 2024. 1, 3, 7, 9, 19, 24, 25, 27, 28
- Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *ICLR*, 2024. 19
- Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021. 19

- Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017. 19
- 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. *TMLR*, 2023. 2, 3, 4
- Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Juhai Chen, Kunpeng Li, Felix Juefei-Xu, Ji Hou, and Saining Xie. Transfer between modalities with metaqueries, 2025. 19
- Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018. 19
- William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 1, 3, 4, 5, 9, 10, 19, 24
- Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. In *ICLR*, 2021. 6
- Kai Qiu, Xiang Li, Hao Chen, Jason Kuen, Xiaohao Xu, Jiuxiang Gu, Yinyi Luo, Bhiksha Raj, Zhe Lin, and Marios Savvides. Image tokenizer needs post-training, 2025. URL <https://arxiv.org/abs/2509.12474>. 19
- 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 *ICML*, 2021. 2, 19
- Vivek Ramanujan, Kushal Tirumala, Armen Aghajanyan, Luke Zettlemoyer, and Ali Farhadi. When worse is better: Navigating the compression-generation tradeoff in visual tokenization. In *NeurIPS*, 2025. 19
- Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022. 19
- Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *NeurIPS*, 2019. 19
- Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. *arXiv preprint arXiv:2502.20388*, 2025. 9, 10, 24, 25
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 3, 19
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 4, 23, 24
- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *NeurIPS*, 2016. 28
- Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *SIGGRAPH*, 2022. 6, 9
- Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. In *ICML*, 2023. 22, 23
- Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. In *ICML*, 2025. 2, 4
- Kiwhan Song, Jaeyeon Kim, Sitan Chen, Yilun Du, Sham Kakade, and Vincent Sitzmann. Selective underfitting in diffusion models. *arXiv preprint arXiv:2510.01378*, 2025. 2

- Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. [23](#)
- Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *CVPR*, 2024. [19](#)
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *CVPR*, 2016. [28](#)
- Hao Tang, Chenwei Xie, Xiaoyi Bao, Tingyu Weng, Pandeng Li, Yun Zheng, and Liwei Wang. Unilip: Adapting clip for unified multimodal understanding, generation and editing, 2025. [3, 19](#)
- Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. In *ICLR*, 2023. [6](#)
- Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *NeurIPS*, 2024. [9, 10, 24, 25](#)
- Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. In *ICCV*, 2025. [19](#)
- Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. [2, 3, 4, 19](#)
- Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *NeurIPS*, 2021. [1](#)
- Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008. [3, 19](#)
- Hanyu Wang, Saksham Suri, Yixuan Ren, Hao Chen, and Abhinav Shrivastava. Larp: Tokenizing videos with a learned autoregressive generative prior. In *ICLR*, 2025a. [19](#)
- Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. *arXiv preprint arXiv:2507.23268*, 2025b. [9](#)
- Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer, 2025c. [2, 3, 8, 9, 10, 24, 25](#)
- Xiao Wang, Ibrahim Alabdulmohsin, Daniel Salz, Zhe Li, Keran Rong, and Xiaohua Zhai. Scaling pre-training to one hundred billion data for vision language models. *arXiv preprint arXiv:2502.07617*, 2025d. [19](#)
- Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, Ming-Ming Cheng, and Xiang Li. Representation entanglement for generation:training diffusion transformers is much easier than you think, 2025. [3](#)
- Jiawei Yang, Tianhong Li, Lijie Fan, Yonglong Tian, and Yue Wang. Latent denoising makes good visual tokenizers, 2025. [3, 19](#)
- Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *CVPR*, 2025. [2, 3, 4, 5, 9, 23, 24](#)
- Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *TMLR*, 2022. [19](#)

- Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. In *ICLR*, 2024a. 9
- 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 *NeurIPS*, 2024b. 2, 3, 19
- 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 *ICLR*, 2025. 2, 3, 5, 7, 9, 10, 19, 24, 25
- Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models. In *ICML*, 2025. 7
- Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 19
- Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 3, 22
- Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. In *NeurIPS*, 2020. 23
- Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and video tokenization with binary spherical quantization. In *ICLR*, 2025. 19
- Anlin Zheng, Xin Wen, Xuanyang Zhang, Chuofan Ma, Tiancai Wang, Gang Yu, Xiangyu Zhang, and Xiaojuan Qi. Vision foundation models as effective visual tokenizers for autoregressive image generation, 2025. 19
- Chuanxia Zheng, Tung-Long Vuong, Jianfei Cai, and Dinh Phung. Movq: Modulating quantized vectors for high-fidelity image generation. In *NeurIPS*, 2022. 19
- Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *TMLR*. 9
- Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 19
- Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xinguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhui Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. 19
- Yongxin Zhu, Bocheng Li, Hang Zhang, Xin Li, Linli Xu, and Lidong Bing. Stabilize the latent space for image autoregressive modeling: A unified perspective. In *NeurIPS*, 2024. 19

A EXTENDED RELATED WORK

Representation encoder as autoencoder. Recent studies have investigated leveraging semantic representations for reconstruction, particularly in MLLMs where diffusion decoders are conditioned on semantic tokens (Sun et al., 2024; Chen et al., 2025b; Pan et al., 2025; Tong et al., 2025). While this improves visual quality, the reliance on large pretrained diffusion decoders makes reconstructions less faithful to the input, limiting their effectiveness as true autoencoders. Very recently, UniLIP (Tang et al., 2025) employs a one-step convolutional decoder on top of InternViT (Zhu et al., 2025), achieving reconstruction quality surpassing SD-VAE. However, UniLIP relies on additional large-scale fine-tuning of pretrained ViTs, arguing that frozen pretrained representation encoders lacks sufficient visual detail. In contrast, we show this is not the case: frozen representation encoders achieves comparable reconstruction performance while enabling much faster convergence in diffusion training.

Another line of related work also try to utilize representation encoders directly as tokenizers. VFM-Tok (Zheng et al., 2025) and DiGIT (Zhu et al., 2024) applies vector-quantization directly to pretrained representation encoders like Dino or SigLIP. These approaches transform representation encoders into an effective tokenizer for AR models, but still suffer from the information capacity bottleneck brought by quantization.

Compressed image tokenizers. Autoencoders have long been used to compress images into low-dimensional representations for reconstruction (Hinton & Salakhutdinov, 2006; Vincent et al., 2008). VAEs (Kingma & Welling, 2014) extend this paradigm by mapping inputs to Gaussian distributions, while VQ-VAEs (Oord et al., 2017; Razavi et al., 2019) introduce discrete latent codes. VQGAN (Esser et al., 2021) adds adversarial objectives, and ViT-VQGAN (Esser et al., 2021; Cao et al., 2023) modernizes the architecture with Vision Transformers (ViTs) (Dosovitskiy et al., 2021). Other advances include multi-stage quantization (Lee et al., 2022; Zheng et al., 2022), lookup-free schemes (Mentzer et al., 2024; Zhao et al., 2025), token-efficient designs such as TiTok and DCAE (Yu et al., 2024b; Chen et al., 2025c), and structure-preserving approaches like EQ-VAE (Kouzelis et al., 2025a). (Hansen-Estruch et al., 2025) further explores the scaling behavior of VAEs. LARP, CRT, REPA-E (Wang et al., 2025a; Ramanujan et al., 2025; Yu et al., 2025) tried to improve VAE with generative priors via back-propagation. In contrast, we dispense with aggressive compression and instead adopt a pretrained representation encoders as encoder. This avoids the encoder collapsing into shallow features optimized only for reconstruction loss, while providing strong pretrained representations that serve as a robust latent space.

Generative models. Modern image generation is dominated by two paradigms: autoregressive (AR) models and diffusion models. AR models (De Fauw et al., 2019; Ramesh et al., 2022; Yu et al., 2022; Parmar et al., 2018; Chen et al., 2020a) generate images sequentially, token by token, and benefit from powerful language-model architectures but often suffer from slow sampling. Diffusion models (Ho et al., 2020; Nichol & Dhariwal, 2021; Rombach et al., 2022; Ma et al., 2024; Peebles & Xie, 2023) instead learn to iteratively denoise noisy signals, offering superior sample quality and scalability, though at the cost of many sampling steps. In this work, we build on diffusion models but adapt them to high-dimensional latent spaces provided by pretrained representation encoders. We find representation encoders provide faster convergence and improved scaling behavior.

Robust decoders for generation. Recent works suggest that incorporating masking or latent denoising losses can improve tokenizer training. *l*-DeTok (Yang et al., 2025) shows that combining both losses yields strong VAEs for second-stage MAR (Li et al., 2024b; Fan et al., 2025b) generation, while RobusTok (Qiu et al., 2025) demonstrates that training with perturbed tokens makes decoders more robust. Notably, Yang et al. (2025) report that denoising loss only increase performance when encoder and decoder are jointly trained, whereas we observe substantial gains in generation quality with frozen encoders.

Visual representation learning. Visual representations primarily fall into two broad families: self-supervised encoders that learn invariances from augmented views or masked prediction (He et al., 2019; Chen et al., 2020b; Grill et al., 2020; Zhou et al., 2021; He et al., 2021), and language-supervised encoders trained from image–text pairs (Radford et al., 2021; Zhai et al., 2023; Chen et al., 2024; Tschannen et al., 2025). Recent work (Fan et al., 2025a; Tschannen et al., 2025; Wang

et al., 2025d) scales both lines on billion-scale corpora, yielding robust, semantically rich features. In this work, we leverage these pretrained encoders directly: we freeze the encoder, and train a lightweight decoder to form a Representation Autoencoder (RAE). We show that such RAEs are effective for both reconstruction and generation.

B PROOFS

B.1 PROOF OF LOWER BOUND FOR TRAINING LOSS

Theorem 1. Assuming $\mathbf{x} \sim p(\mathbf{x}) \in \mathbb{R}^n$, $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}_n)$, $t \in [0, 1]$. Let $\mathbf{x}_t = (1 - t)\mathbf{x} + t\boldsymbol{\varepsilon}$, consider the function family

$$\mathcal{G}_d = \{g(\mathbf{x}_t, t) = \mathbf{B}f(\mathbf{A}\mathbf{x}_t, t) : \mathbf{A} \in \mathbb{R}^{d \times n}, \mathbf{B} \in \mathbb{R}^{n \times d}, f : [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d\} \quad (1)$$

where $d < n$, f refers to a stack of standard DiT blocks whose width is smaller than the token dimension from the representation encoder, and \mathbf{A}, \mathbf{B} denote the input and output linear projections, respectively. Then for any $g \in \mathcal{G}_d$,

$$\mathcal{L}(g, \theta) = \int_0^1 \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}), \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}_n)} [\|g(\mathbf{x}_t, t) - (\boldsymbol{\varepsilon} - \mathbf{x})\|^2] dt \geq \sum_{i=d+1}^n \lambda_i \quad (2)$$

where λ_i are the eigenvalues of the covariance matrix of the random variable $W = \boldsymbol{\varepsilon} - \mathbf{x}$.

Notably, when $d \geq n$, \mathcal{G}_d contains the unique minimizer to $\mathcal{L}(g, \theta)$.

Proof. By Albergo et al. (2023), the distribution ρ_t of \mathbf{x}_t satisfies $\rho_0 = p(\mathbf{x})$, $\rho_1 = \mathcal{N}(0, \mathbf{I}_n)$, and $\partial_t \rho + \nabla \cdot (v\rho) = 0$ where v is the optimal velocity predictor defined as $v(\mathbf{x}_t, t) = \mathbb{E}[\boldsymbol{\varepsilon} - \mathbf{x} | \mathbf{x}_t]$. Also, by Theorem 2.7 in Albergo et al. (2023), there exists $f^* \in C^0((C^1(\mathbb{R}^n))^n; [0, 1])$ ¹ that uniquely minimizes the $\mathcal{L}(f, \theta)$ and perfectly approximates v .

By our training setting, it's reasonable to assume that $\mathbf{x} \sim p(\mathbf{x})$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}_n)$ are independent. Then the distribution of the objective $\mathbf{y} = \boldsymbol{\varepsilon} - \mathbf{x} \sim p_{\mathbf{y}}(\mathbf{y})$ satisfies $p_{\mathbf{y}}(\mathbf{y}) = \int_{\mathbb{R}^n} \mathcal{N}(0, \mathbf{I}_n)(\mathbf{y} + \mathbf{x})p(\mathbf{x})d\mathbf{x}$. Clearly, $p_{\mathbf{y}}$ has full support on \mathbb{R}^n and is strictly positive, indicating \mathbf{y} has a non-zero probability anywhere in \mathbb{R}^n . Similarly, for $\mathbf{x}_t = (1 - t)\mathbf{x} + t\boldsymbol{\varepsilon}$, given any t , $p_{\mathbf{x}_t}(\mathbf{w}) = \int_{\mathbb{R}^n} \mathcal{N}(0, t^2\mathbf{I}_n)(\mathbf{w} - \mathbf{x}) \frac{1}{(1-t)} p\left(\frac{\mathbf{x}}{1-t}\right)d\mathbf{x}$ also has full support on \mathbb{R}^n and is strictly positive, indicating \mathbf{x}_t has a non-zero probability anywhere in \mathbb{R}^n as well.

Recall that for any function $f : \mathcal{X} \rightarrow \mathcal{Y}$, $\text{Im}(f) = \{f(x) : x \in \mathcal{X}\}$. Then for linear transformation $f(\mathbf{x}) = \mathbf{M}\mathbf{x}$ with $\mathbf{M} \in \mathbb{R}^{d \times n}$, $\text{Im}(f) = \{\mathbf{M}\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\}$; we denote this as $\text{Im}(\mathbf{M})$. Now, for any $g \in \mathcal{G}_d$, $\text{Im}(g) = \{\mathbf{B}f(\mathbf{A}\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\} \subseteq \{\mathbf{B}\mathbf{y} : \mathbf{y} \in \mathbb{R}^d\} = \text{Im}(\mathbf{B})$. Since $\text{rank}(\mathbf{B}) \leq d$, $\dim \text{Im}(g) \leq \text{rank}(\mathbf{B}) \leq d < n$, therefore $\text{Im}(g) \subseteq \text{Im}(\mathbf{B}) \subset \mathbb{R}^n$.

Now, given $g \in \mathcal{G}_d$ and the deterministic pair $(\mathbf{x}_t, \mathbf{y}, t) \in (\mathbb{R}^n, \mathbb{R}^n, [0, 1])$, by Projection Theorem,

$$\|g(\mathbf{x}_t, t) - \mathbf{y}\|^2 \geq \|\mathbf{u}_g - \mathbf{y}\|^2 \quad (3)$$

where $\mathbf{u}_g \in \text{Im}(g)$ is the unique minimizer and $\mathbf{u}_g - \mathbf{y}$ is orthogonal to $\text{Im}(g)$. Since $\|\cdot\|^2 \geq 0$, we can take expectation on both sides

$$\begin{aligned} \inf_{g \in \mathcal{G}_d} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}), \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}_n)} [\|g(\mathbf{x}_t, t) - \mathbf{y}\|^2] &\geq \inf_{g \in \mathcal{G}_d} \mathbb{E}[\|\mathbf{u}_g - \mathbf{y}\|^2] \\ &\geq \inf_{\mathbf{u} \in S; \dim S \leq d} \mathbb{E}[\|\mathbf{u} - \mathbf{y}\|^2] \\ &\geq \inf_{S; \dim S \leq d} \mathbb{E}[\|\mathbf{y}\|^2 - \|\mathbf{P}_S \mathbf{y}\|^2] \end{aligned} \quad (4)$$

¹family of functions $f : [0, 1] \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ that is continuous in t for all $(\mathbf{x}, t) \in [0, 1] \times \mathbb{R}^n$, and $f(\cdot, t)$ is a continuously differentiable function from \mathbb{R}^n to \mathbb{R}^n .

where \mathbf{P}_S denote the projection matrix from \mathbb{R}^n onto S . Without loss of generality, we assume $\mathbb{E}[\mathbf{x}] = \mathbf{0}$ ², then Eq 4 can be expanded as

$$\begin{aligned} \inf_{\mathcal{G}_d} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}), \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}_n)} [\|g(\mathbf{x}_t, t) - \mathbf{y}\|^2] &\geq \sum_{i=1}^n \mathbb{E}[y_i^2] - \sup_{S; \dim S \leq d} \sum_{i=1}^n \mathbb{E}[(\mathbf{P}_S \mathbf{y})_i^2] \\ &= \text{Tr}(\text{Cov}(\mathbf{y})) - \sup_{S; \dim S \leq d} \text{Tr}(\text{Cov}(\mathbf{P}_S \mathbf{y})) \\ &\geq \sum_{i=1}^n \lambda_i - \sum_{i=1}^d \lambda_i = \sum_{i=d+1}^n \lambda_i \end{aligned} \quad (5)$$

where Eq 5 is obtained via Ky-Fan Maximum Principle.

When $d \geq n$, $\sup_S \mathbb{E}[\|\mathbf{P}_S \mathbf{y}\|^2] = \mathbb{E}[\|\mathbf{y}\|^2]$, leading to a trivial lower bound in Eq 5, and $\mathcal{G}_d = C^0((C^1(\mathbb{R}^n))^n; [0, 1])$. \square

B.2 PROOF OF LOWER BOUND FOR INFERENCE LOSS

Theorem 2. Consider the same setup as Theorem 1. Let \mathbf{x}_1 be the initial random variables in the sampling process, and

$$\begin{aligned} \mathbf{x}_0 &= ODE(g, \mathbf{x}_1, 1 \rightarrow 0) \\ \mathbf{x}_0^* &= ODE(f^*, \mathbf{x}_1, 1 \rightarrow 0) \end{aligned}$$

where $ODE(f, \mathbf{x}, t \rightarrow s)$ refers to any ODE solver that integrates f from time t to s using \mathbf{x} as the initial condition. We further assume for any $(\mathbf{x}, \mathbf{y}, t) \in (\mathbb{R}^n, \mathbb{R}^n, [0, 1])$, there exists constant $L > 0$ such that

$$\|f^*(\mathbf{x}, t) - f^*(\mathbf{y}, t)\| \leq L \|\mathbf{x} - \mathbf{y}\|$$

then

$$\|\mathbf{x}_0^* - \mathbf{x}_0\| \geq \frac{1 - e^{-L}}{L} \sum_{i=d+1}^n \lambda_i \quad (6)$$

Proof. We first define a forward ODE that integrates from $0 \rightarrow 1$ $\mathbf{x}_{\leftarrow_t} := \mathbf{x}_{-t}$, and

$$\begin{aligned} d\mathbf{x}_{\leftarrow_t} &= g(\mathbf{x}_{\leftarrow_t}, t) dt \\ d\mathbf{x}_{\leftarrow_t}^* &= f^*(\mathbf{x}_{\leftarrow_t}^*, t) dt \end{aligned}$$

Then

$$\begin{aligned} \frac{d}{dt} \|\mathbf{x}_{\leftarrow_t}^* - \mathbf{x}_{\leftarrow_t}\| &= \|f^*(\mathbf{x}_{\leftarrow_t}^*, t) - g(\mathbf{x}_{\leftarrow_t}, t)\| \\ &\geq \|f^*(\mathbf{x}_{\leftarrow_t}^*, t) - g(\mathbf{x}_{\leftarrow_t}, t)\| - \|f^*(\mathbf{x}_{\leftarrow_t}^*, t) - f^*(\mathbf{x}_{\leftarrow_t}, t)\| \\ &\geq \|\Delta\| - L \|\mathbf{x}_{\leftarrow_t}^* - \mathbf{x}_{\leftarrow_t}\| \end{aligned} \quad (7)$$

where Δ denotes the approximation error to f^* for $g \in \mathcal{G}_d$. Applying Gronwall's Lemma, we have

$$\begin{aligned} e^{L\leftarrow_t} \|\mathbf{x}_{\leftarrow_t}^* - \mathbf{x}_{\leftarrow_t}\| \Big|_0^1 &\geq \int_0^t e^{Ls} \|\Delta\| ds \\ \implies \|\mathbf{x}_0^* - \mathbf{x}_0\| &= \|\mathbf{x}_{\leftarrow_1}^* - \mathbf{x}_{\leftarrow_1}\| \geq \frac{(1 - e^{-L})}{L} \sum_{i=d+1}^n \lambda_i \end{aligned} \quad (8)$$

where by Theorem 1, $\|\Delta\| \geq \sum_{i=d+1}^n \lambda_i$. \square

²Non-centered \mathbf{x} with $\mathbb{E}[\mathbf{x}] = \mu$ will additionally introduce $\|\mu\|^2 - \|\mathbf{P}_S \mu\|^2$ to the lower bound; we ignore this term since most data processing pipelines will center the data.

C RAE IMPLEMENTATION

C.1 ENCODER NORMALIZATION

For any given frozen representation encoders, we discard any [CLS] or [REG] token produced by the encoder, and keep the all of patch tokens. We then apply a layer normalization to each token independently, to ensure each token has zero mean and unit variance across channels. We note that all representation encoders we use adopt the standard ViT architecture ([Dosovitskiy et al., 2021](#)), which have already applied layer normalization after the last transformer block. Therefore, we only need to cancel the affine parameters of the layer normalization in representation encoders. This does not affect the representation quality of representation encoders, as it is a linear transformation.

Practical Notes. Specifically, we use DINOv2 with Registers ([Darct et al., 2025](#)). Since DINOv2 only provides variants with $p_e = 14$, we interpolate the input images to 224×224 but set $p_d = 16$, ensuring the model still produces 256 tokens while reconstructing 256×256 images.

C.2 DECODER TRAINING DETAILS

Datasets. We primarily use ImageNet-1K for training all decoders. Most experiments are conducted at a resolution of 256×256 . For 512-resolution synthesis without decoder upsampling, we train decoders directly on 512×512 images.

Decoder Architecture. The decoder takes the token embeddings produced by the frozen encoder takes the token embedding reconstructs them back into the pixel space using the same patch size as the encoder. As a result, it can generate images with the same spatial spatial resolution as the encoder’s inputs input. Following [He et al. \(2021\)](#), we prepend a learnable [CLS] token decoder’s input sequence and discard it after decoding.

Discriminator Architecture. We include the majority of our decoder training details in Table 12. We follow most of the design choices in StyleGAN-T ([Sauer et al., 2023](#)), except for using a frozen Dino-S/8 ([Caron et al., 2021](#)) instead of Dino-S/16 as the discriminator. We found using Dino-S/8 stabilizes training and avoid the decoder to generate adversarial patches. We also remove the virtual batch norm in [Sauer et al. \(2023\)](#) and use the standard batch norm instead. All input is interpolated to 224×224 resolution before feeding into the discriminator.

Table 12: Training configuration for decoder and discriminator.

Component	Decoder	Discriminator
optimizer	Adam	Adam
max learning rate	2×10^{-4}	2×10^{-4}
min learning rate	2×10^{-5}	2×10^{-5}
learning rate schedule	cosine decay	cosine decay
optimizer betas	(0.5, 0.9)	(0.5, 0.9)
weight decay	0.0	0.0
batch size	512	512
warmup	1 epoch	1 epoch
loss	$\ell_1 + \text{LPIPS} + \text{GAN}$	adv.
Model	ViT-(B, L, XL)	Dino-S/8 (frozen)
LPIPS start epoch	0	–
disc. start epoch	–	6
adv. loss start epoch	8	–
Training epochs	16	10

Losses. For training the decoder, we use a mixture of L1, LPIPS ([Zhang et al., 2018](#)) and adversarial loss ([Goodfellow et al., 2014](#)):

$$z = E(x), \hat{x} = D(z) \\ \mathcal{L}_{rec}(x) = \omega_L \text{LPIPS}(\hat{x}, x) + \text{L1}(\hat{x}, x) + \omega_G \lambda \text{GAN}(\hat{x}, x),$$



Figure 8: **Reconstruction examples.** From left to right: input image, RAE (DINOv2-B), RAE (SigLIP2-B), RAE (MAE-B), SD-VAE. Zoom in for details.

where E, D are the encoder and decoder of RAE. We set $\omega_L = 1$. and $\omega_G = 0.75$. We use the same losses as in StyleGAN-T (Sauer et al., 2023) for discriminator, and a GAN loss as in Esser et al. (2021). We also adopt the adaptive weight λ for GAN loss proposed in Esser et al. (2021) to balance the scales of reconstruction and adversarial losses. λ is defined as:

$$\lambda = \frac{\|\nabla_{\hat{x}} \mathcal{L}_{rec}\|}{\|\nabla_{\hat{x}} \text{GAN}(\hat{x}, x)\| + \epsilon},$$

Augmentations. For data augmentation, we first resize the input image to 384×384 and then randomly crop to 256×256 . We also apply differentiable augmentations with default hyperparameters in Zhao et al. (2020) before discriminator.

C.3 VISUALIZATIONS

We present visualizations of reconstructions from different RAEs. As shown in Figure 8, all RAEs achieve satisfactory reconstruction fidelity.

D DIFFUSION MODEL IMPLEMENTATION

Datasets. We primarily use ImageNet-1K (Russakovsky et al., 2015) for diffusion training. Most experiments are conducted at a resolution of 256×256 . For 512-resolution synthesis without decoder upsampling, we train diffusion models directly on 512×512 images.

Models. By default, we use LightningDiT (Yao et al., 2025) as the backbone of our diffusion model. We use a continuous time formulation of flow matching and restrict the timestep input to real values in $[0, 1]$. Following prior work (Song et al., 2021), we replace the timestep embedding with a Gaussian Fourier embedding layer. We also add Absolute Positional Embeddings (APE) to the input tokens in addition to RoPE, though we do not observe significant performance difference with or without APE.

Model Config	Dim	Num-Heads	Depth
S	384	6	12
B	768	12	12
L	1024	16	24
XL	1152	16	28
XXL	1280	16	32
H	1536	16	32
G	2048	16	40
T	2688	21	40

Table 13: Model configurations for different sizes.

For DiT^{DH}, we generally follow the same architecture as DiT, and does not reapply APE for the DDT head input. We use a linear layer to map the DiT^{DH} encoder output to the DiT^{DH} decoder

dimension when the dimension of DiT and DDT head mismatches. We provide a detailed model configuration in Table 13.

Compute. For all models based on RAE, we use a patch size of 1. For baseline experiments on VAE and pixel inputs, we use patch sizes of 2 and 16, respectively. Across all 256×256 experiments, the diffusion model processes a token sequence of length 256. Consequently, the computational cost of the DiT backbone remains identical across RAE, VAE, and pixel settings, with differences arising only from the patchification step. Since patchification accounts for less than 1% of the total GFLOPs, training DiT on different autoencoders introduces only a negligible compute overhead.

Optimization. For DiT, we strictly follow the optimization strategy in LightningDiT (Yao et al., 2025), using AdamW with a constant learning rate of 2.0×10^{-4} , a batch size of 1024 and an EMA weight of 0.9999. We do not observe instability or abnormal training dynamics with this recipe on DiT. For DiT^{DH} , we find using the recipe in (Yao et al., 2025) leads to loss spikes at later epochs and slow EMA model convergence at early epochs. We instead use a linear decay from 2.0×10^{-4} to 2.0×10^{-5} with a constant warmup of 40 epochs. To encourage the convergence of EMA model, we change the EMA weight from 0.9999 to 0.9995. Additionally, we use gradient clipping of 1.0 for DiT^{DH} . Other optimization hyperparameters are the same as DiT. All models are trained for 80 epochs unless otherwise specified. We only report EMA model performance.

Sampling. We use standard ODE sampling with Euler sampler and 50 steps by default. We find the performance generally converges above 50 steps. We use the same sampling hyperparameters for both DiT and DiT^{DH} .

Computation. We use PyTorch/XLA on TPU for all training and inference on RAE. For evaluation, we use a single v6e-8 to generate 50K samples. For the 800 epoch $\text{DiT}^{\text{DH}}\text{-XL}$ (DINOv2-B) result, we conduct the generation on a machine with 4 A100s and evaluate FID on CPU due to a lack of TensorFlow GPU support. We use an internal JAX codebase for training baseline models on VAEs.

E SAMPLING DETAILS FOR FID EVALUATION

For the FID-50K evaluation on RAE, we followed the protocol used in (Tian et al., 2024; Li et al., 2024b; Ren et al., 2025; Wang et al., 2025c), sampling 50 images from each class for a total of 50,000 images. The reference statistics were taken from the ADM pre-computed statistics (Dhariwal & Nichol, 2021) over the full ImageNet dataset. Another commonly adopted sampling strategy is to uniformly sample class labels 50K times and generate images accordingly (Peebles & Xie, 2023; Ma et al., 2024; Yu et al., 2025). It is worth noting that this strategy is not equivalent to the per-class sampling scheme—although random sampling asymptotically converges to the per-class version, the resulting FID scores differ slightly in practice.

To ensure a fair comparison, we re-evaluate previous state-of-the-art methods that did not use class-balanced sampling with class-balanced sampling. We also evaluate our method with random sampling. As shown in Table 14, all methods’ FID improves consistently with class-balanced sampling.

We also note that the original ImageNet training set is inherently unbalanced, with class sizes ranging from approximately 732 to 1,300 samples (Russakovsky et al., 2015). Therefore, the common assumption behind both balanced and random sampling—namely, that the label distribution is uniform—is not entirely accurate. However, since 895 classes contain exactly 1,300 samples, the dataset exhibits a high degree of near-equivalence among most classes. This partial balance may partly explain why balanced sampling consistently yields better results, as it more closely approximates the true label distribution of the training set. As the FID values approach the lower ranges, these subtle details begin to have a greater impact. We therefore want to raise awareness within the community. Finally, while the FID metric remains a useful indicator of generative quality, its absolute value becomes less meaningful as generation fidelity continues to improve.

F THEORY EXPERIMENT SETUP

In this section we list the setup of experiments in Section 4 for overfitting images.

Method	Epochs	Generation@256 w/o guidance				Generation@256 w/ guidance			
		Random		Balanced		Random		Balanced	
		gFID↓	IS↑	gFID	IS	gFID↓	IS↑	gFID	IS
Autoregressive									
VAR (Tian et al., 2024)	350	-	-	1.92	323.1	-	-	1.73	350.2
MAR (Li et al., 2024b)	800	-	-	2.35	227.8	-	-	1.55	303.7
xAR-H (Ren et al., 2025)	800	-	-	-	-	-	-	1.24	301.6
Latent Diffusion with VAE									
SiT (Ma et al., 2024)	1400	8.61	131.7	8.54	132.0	2.06	270.3	1.95	259.5
REPA (Yu et al., 2025)	800	5.90	157.8	5.78	158.3	1.42	305.7	1.29	306.3
DDT (Wang et al., 2025c)	400	-	-	6.27	154.7	1.40	303.6	1.26	310.6
REPA-E (Leng et al., 2025)	800	1.83	217.3	1.70	217.3	1.26	314.9	1.15	304.0
Latent Diffusion with RAE (Ours)									
DiT ^{DH} -XL (DINOv2-B)	800	1.60	242.7	1.51	242.9	1.28	262.9	1.13	262.6

Table 14: **Performance of different methods using different sampling strategies.** The officially reported numbers are marked in gray.

Models. By default, we use a DiT with depth 12, width 768 and a attention head of 4. The depth varies in {384, 512, 640, 768, 896} and width varies in {4, 12, 16, 24} in Figure 3. Other configurations are the same as Section D.

Targets. We use three images for overfitting experiments, and all numbers reported are the average on three independent run on each images. We resize all targets to 256×256 and not use any data augmentation .

Optimizations & Sampling. For a single target image, the batch size only influences the timestep. We therefore use a relatively small batch size of 32 and a constant learning rate of 2×10^{-4} , optimized with AdamW ($\beta = (0.9, 0.95)$). The model is trained for 1200 steps without EMA. For sampling, We use standard ODE sampling with Euler sampler and 25 steps by default.

G ADDITIONAL ABLATION STUDIES

G.1 GENERATION PERFORMANCE ACROSS ENCODERS

As shown in Table 15a, DINOv2-B achieves the best overall performance. MAE performs substantially worse in generation, despite yielding much lower rFID. This shows that a low rFID does not necessarily imply a good image tokenizer. Therefore, we use DINOv2-B as the default encoder for our image generation experiments.

(a) gFID and rFID of different encoders w/ and w/o noisy-robust decoding.	(b) gFID and rFID of different DINOv2 sizes w/ and w/o noisy-robust decoding.	(c) Scaling τ for DINOv2-B.																																							
<table border="1"> <thead> <tr> <th>Model</th> <th>gFID</th> <th>rFID</th> </tr> </thead> <tbody> <tr> <td>DINOv2-B</td> <td>4.81 / 4.28</td> <td>0.49 / 0.57</td> </tr> <tr> <td>SigLIP2-B</td> <td>6.69 / 4.93</td> <td>0.53 / 0.82</td> </tr> <tr> <td>MAE-B</td> <td>16.14 / 8.38</td> <td>0.16 / 0.28</td> </tr> </tbody> </table>	Model	gFID	rFID	DINOv2-B	4.81 / 4.28	0.49 / 0.57	SigLIP2-B	6.69 / 4.93	0.53 / 0.82	MAE-B	16.14 / 8.38	0.16 / 0.28	<table border="1"> <thead> <tr> <th>Model</th> <th>gFID</th> <th>rFID</th> </tr> </thead> <tbody> <tr> <td>S</td> <td>3.83 / 3.50</td> <td>0.52 / 0.64</td> </tr> <tr> <td>B</td> <td>4.81 / 4.28</td> <td>0.49 / 0.57</td> </tr> <tr> <td>L</td> <td>6.77 / 6.09</td> <td>0.52 / 0.59</td> </tr> </tbody> </table>	Model	gFID	rFID	S	3.83 / 3.50	0.52 / 0.64	B	4.81 / 4.28	0.49 / 0.57	L	6.77 / 6.09	0.52 / 0.59	<table border="1"> <thead> <tr> <th>τ</th> <th>gFID</th> <th>rFID</th> </tr> </thead> <tbody> <tr> <td>0.0</td> <td>4.81</td> <td>0.49</td> </tr> <tr> <td>0.5</td> <td>4.39</td> <td>0.54</td> </tr> <tr> <td>0.8</td> <td>4.28</td> <td>0.57</td> </tr> <tr> <td>1.0</td> <td>4.20</td> <td>0.60</td> </tr> </tbody> </table>	τ	gFID	rFID	0.0	4.81	0.49	0.5	4.39	0.54	0.8	4.28	0.57	1.0	4.20	0.60
Model	gFID	rFID																																							
DINOv2-B	4.81 / 4.28	0.49 / 0.57																																							
SigLIP2-B	6.69 / 4.93	0.53 / 0.82																																							
MAE-B	16.14 / 8.38	0.16 / 0.28																																							
Model	gFID	rFID																																							
S	3.83 / 3.50	0.52 / 0.64																																							
B	4.81 / 4.28	0.49 / 0.57																																							
L	6.77 / 6.09	0.52 / 0.59																																							
τ	gFID	rFID																																							
0.0	4.81	0.49																																							
0.5	4.39	0.54																																							
0.8	4.28	0.57																																							
1.0	4.20	0.60																																							

Table 15: **Ablations on noise-augmented decoder training.** Despite minor drop in rFID, the noise-augmented training strategy can greatly improve the gFID across different encoders and model sizes. Default setups are marked in gray.

Depth	Width	GFLops	FID ↓
6	1152 (XL)	25.65	2.36
4	2048 (G)	53.14	2.31
2	2048 (G)	26.78	2.16

Table 16: **DDT head should be wide and shallow.** A wide, shallow head yields lower FID than deeper (4-layer G) or narrower (6-layer XL) ones .

	2-768	2-1536	2-2048	2-2688
Dino-S	2.66	2.47	2.42	2.43
Dino-B	2.49	2.24	2.16	2.22
Dino-L	N/A	2.95	2.73	2.64

Table 17: **Unguided gFID of different RAE and DDT head.** Larger RAE benefits more from wider DDT head. $d-w$: a DDT head with d layers and width w . Default setups are marked in gray.

G.2 DESIGN CHOICES FOR NOISE-AUGMENTED DECODING

We first analyze how noise-robust decoding affects reconstruction and generation. Table 15c shows that larger τ improves generative FID (gFID) consistently, but slightly worsens reconstruction FID (rFID). This supports our intuition: noise encourages the decoder to learn smoother mappings that generalize better to imperfect latents, improving generation quality, but reducing exact reconstruction accuracy.

To test the robustness of this trade-off, we evaluate different encoders (Table 15a) with $\sigma = 0.8$. Across all encoders, noisy training improves gFID while mildly harming rFID. The effect is strongest for weaker encoders such as MAE-B, where gFID improves from 16.14 to 8.38. Finally, Table 15b shows that the benefit holds across encoder sizes, suggesting that robust decoder training is broadly applicable.

Together, these results highlight a general principle: decoders should not only reconstruct clean latents, but also handle their noisy neighborhoods. This simple change enables RAEs to serve as stronger backbones for diffusion models.

G.3 DESIGN CHOICES FOR THE DDT HEAD

We now investigate design variants of the DDT head to identify those that serve its role more effectively. Two factors turn out to be crucial: (a) the head needs to be wide and shallow, and (b) its benefit depends on the size of the underlying RAE encoder.

Width and Depth. We first vary the architecture of the DDT head, sweeping both width and depth while keeping the total parameter count approximately fixed. As shown in Table 16, a 2-layer, 2048-dim (G) head outperforms a 6-layer, 1152-dim (XL) head by a large margin, despite having similar GFlops. Moreover, a 4-layer, 2048-dim head does not improve over the 2-layer version, even though it has double the GFlops. This suggests that a wide and shallow head is more effective for denoising.

Dependence on Encoder Size. Next, we analyze how the effect of the DDT head scales with the size of the RAE encoder. We fix the DiT backbone as DiT-XL and vary the DDT head width from 768 (B) to 1536 (H), 2048 (G), and 2688 (T). We train DiT^{DH} models on top of three RAEs: DINOv2-S, DINOv2-B, and DINOv2-L. As shown in Table 17, the optimal DDT head width increases as the encoder scales. When using DINOv2-S and DINOv2-B, the performance converges at a DDT head width of 2048 (G), while 2688 (T) head still brings performance gains on DINOv2-L. This suggests that the larger RAE encoders benefit more from a wider DDT head.

By default, we use a 2-layer, 2048-dim DDT head for all DiT^{DH} models.

H ADDITIONAL SCALING RESULTS

We examine how model scale affects training loss in Figure 9. Increasing the model’s computational capacity leads DiT^{DH} to converge faster and reach a lower final loss.

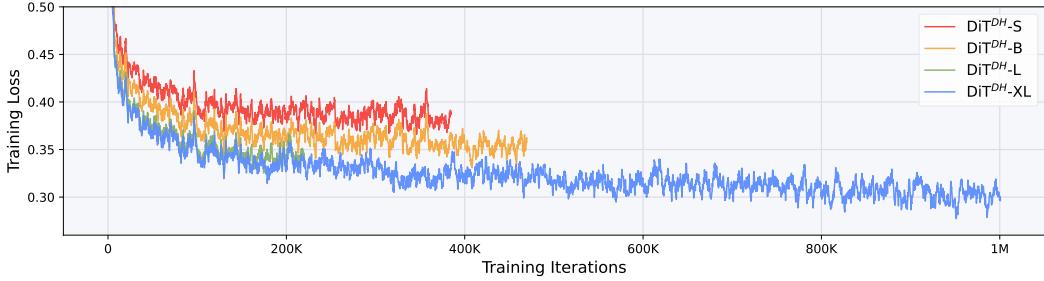


Figure 9: **Training loss of DiT^{DH} on DINOv2-B.** We use an EMA weight of 0.9 to smooth the loss.

I GUIDANCE

We primarily adopt AutoGuidance (Karras et al., 2025) as our guidance method, as it is easier to tune than CFG with interval (Ho & Salimans, 2022; Kynkänniemi et al., 2024) and consistently delivers better performance. CFG with interval is used only for the DiT-XL + DINOv2-S with guidance result reported in Table 8.

AutoGuidance. We adopt AutoGuidance (Karras et al., 2025) as our primary guidance method. The idea is to use a weaker diffusion model to guide a stronger one, analogous to the principle of Classifier-Free Guidance (CFG) (Ho & Salimans, 2022). Empirically, we observe that weaker base models and earlier training checkpoints consistently yield stronger guidance effects. In practice, we use the smallest variant, DiT^{DH}-S, as the guiding model, typically adopting an early checkpoint. Unless otherwise specified, AutoGuidance results are obtained using a guidance scale of 1.5 and a 20-epoch checkpoint of DiT^{DH}-S. For the best result of 1.13 FID (DiT^{DH}-XL on DINOv2-B, 256×256), we sweep the guidance scale and use a guidance scale of 1.42 with a 14-epoch DiT^{DH}-S checkpoint. Notably, training this base model requires only about 0.05% of the compute used to train the guided model (DiT^{DH}-XL for 800 epochs).

Classifier-Free Guidance. We also evaluate CFG (Ho & Salimans, 2022) on RAE. Interestingly, CFG without interval does not improve FID; in fact, applying it from the first diffusion step increases FID. With Guidance Interval (Kynkänniemi et al., 2024), CFG can achieve competitive FID after careful grid search over scale and interval. However, on our final model (DiT^{DH}-XL with DINOv2-B), the best CFG result remains inferior to AutoGuidance. Considering both performance and tuning overhead, we adopt AutoGuidance as our default guidance method.

J DESCRIPTIONS FOR FLOW-BASED MODELS

Diffusion Models (Ho et al., 2020; Dhariwal & Nichol, 2021; Karras et al., 2022) and more generally flow-based models (Albergo et al., 2023; Lipman et al., 2023; Liu et al., 2023) are a family of generative models that learn to reverse a reference “noising” process. One of the most commonly used “noising” process is the linear interpolation between i.i.d Gaussian noise and clean data (Esser et al., 2024; Ma et al., 2024):

$$\mathbf{x}_t = (1-t)\mathbf{x} + t\boldsymbol{\varepsilon}$$

where $\mathbf{x} \sim p(\mathbf{x})$, $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}_n)$, $t \in [0, 1]$, and we denote \mathbf{x}_t ’s distribution as $\rho_t(\mathbf{x})$ with $\rho_0 = p(\mathbf{x})$ and $\rho_1 = \mathcal{N}(0, \mathbf{I})$. Generation then starts at $t = 1$ with pure noise, and simulates some differential equation to progressively denoise the sample to a clean one. Specifically for flow-based models, the differential equations (an ordinary differential equation (ODE) or a stochastic differential equation (SDE)) are formulated through an underlying velocity $v(\mathbf{x}_t, t)$ and a score function $s(\mathbf{x}_t, t)$

$$\begin{aligned} \text{ODE} \quad & d\mathbf{x}_t = v(\mathbf{x}_t, t)dt \\ \text{SDE} \quad & d\mathbf{x}_t = v(\mathbf{x}_t, t)dt - \frac{1}{2}w_t s(\mathbf{x}_t, t)dt + \sqrt{w_t} d\bar{\mathbf{w}}_t \end{aligned}$$

where w_t is any scalar-valued continuous function (Ma et al., 2024), and \bar{w}_t is the reverse-time Wiener process. The velocity $v(\mathbf{x}_t, t)$ is represented as a conditional expectation

$$v(\mathbf{x}_t, t) = \mathbb{E}[\dot{\mathbf{x}}_t | \mathbf{x}_t] = \mathbb{E}[\boldsymbol{\varepsilon} - \mathbf{x} | \mathbf{x}_t]$$

and can be approximated with model v_θ by minimizing the following training objective

$$\mathcal{L}_{\text{velocity}}(\theta) = \int_0^1 \mathbb{E}_{\mathbf{x}, \boldsymbol{\varepsilon}} \left[\|v_\theta(\mathbf{x}_t, t) - (\boldsymbol{\varepsilon} - \mathbf{x})\|^2 \right] dt$$

The score function $s(\mathbf{x}_t, t)$ is also represented as a conditional expectation

$$s(\mathbf{x}_t, t) = -\frac{1}{t} \mathbb{E}[\boldsymbol{\varepsilon} | \mathbf{x}_t]$$

Notably, s is equivalent to v up to constant factor (Albergo et al., 2023), so it's enough to estimate only one of the two vectors.

K EVALUATION DETAILS

K.1 EVALUATION

We strictly follow the setup and use the same reference batches of ADM (Dhariwal & Nichol, 2021) for evaluation, following their official implementation.³ We use TPUs for generating 50k samples and use one single NVIDIA A100 80GB GPU for evaluation.

In what follows, we explain the main concept of metrics that we used for the evaluation.

- **FID** (Heusel et al., 2017) evaluates the distance between the feature distributions of real and generated images. It relies on the Inception-v3 network (Szegedy et al., 2016) and assumes both distributions follow multivariate Gaussians.
- **IS** (Salimans et al., 2016) also uses Inception-v3, but evaluates logits directly. It measures the KL divergence between the marginal label distribution and the conditional label distribution after softmax normalization.
- **Precision and recall** (Kynkänniemi et al., 2019) follow their standard definitions: precision reflects the fraction of generated images that appear realistic, while recall reflects the portion of the training data manifold covered by generated samples.

L UNCONDITIONAL GENERATION

We are also interested in how RAEs perform in unconditional generation. To evaluate this, we train DiT^{DH}-XL on RAE latents without labels. Following RCG (Li et al., 2024a), we set labels to null during training and use the same null label at generation time. While classifier-free guidance (CFG) does not apply in this setting, AutoGuidance remains applicable. We therefore train DiT^{DH}-XL for 200 epochs with AG (detailed in Section I).

As shown in Table 18, our model achieves substantially better performance than DiT-XL trained on VAE latents. Compared to RCG, a method specifically designed for unconditional generation, our approach attains competitive performance while being much simpler and more straightforward, without the need for two-stage generation.

Method	gFID ↓	IS ↑
DiT-XL + VAE	30.68	32.73
DiT ^{DH} -XL + DINOv2-B (w/ AG)	4.96	123.12
RCG + DiT-XL	4.89	143.2

Table 18: Comparison of unconditional generation on ImageNet 256×256 .

M VISUAL RESULTS

We show uncurated 512×512 samples sampled from our most performant model: DiT^{DH}-XL on DINOv2-B with autoguidance scale = 1.5.

³<https://github.com/openai/guided-diffusion/tree/main/evaluations>



Figure 10: Uncurated 512×512 $\text{DiT}^{\text{DH}}\text{-XL}$ samples.

AutoGudance Scale = 1.5

Class label = "golden retriever" (207)



Figure 11: Uncurated 512×512 $\text{DiT}^{\text{DH}}\text{-XL}$ samples.

AutoGudance Scale = 1.5

Class label = "husky" (250)



Figure 12: **Uncurated** 512×512 **DiT^{DH}-XL** samples.

AutoGudance Scale = 1.5
Class label = "cliff" (972)



Figure 13: **Uncurated** 512×512 **DiT^{DH}-XL** samples.

AutoGudance Scale = 1.5
Class label = "macaw" (88)

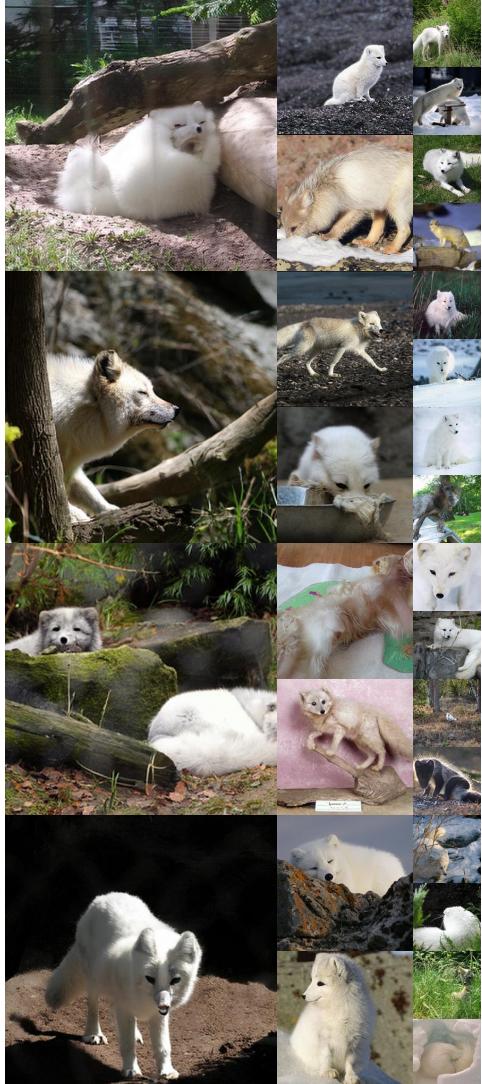


Figure 14: **Uncurated** 512×512 **DiT^{DH}-XL** samples.

AutoGuidance Scale = 1.5

Class label = "arctic fox" (279)



Figure 15: **Uncurated** 512×512 **DiT^{DH}-XL** samples.

AutoGuidance Scale = 1.5

Class label = "balloon" (417)