Epsilon-VAE: Denoising as Visual Decoding

Long Zhao [1] **Sanghyun Woo** [1] **Ziyu Wan** [1 2 *] **Yandong Li** [1] **Han Zhang** [1] **Boqing Gong** [1] **Hartwig Adam** [1]
Xuhui Jia [1] **Ting Liu** [1]

Abstract

In generative modeling, tokenization simplifies complex data into compact, structured representations, creating a more efficient, learnable space. For high-dimensional visual data, it reduces redundancy and emphasizes key features for high-quality generation. Current visual tokenization methods rely on a traditional autoencoder framework, where the encoder compresses data into latent representations, and the decoder reconstructs the original input. In this work, we offer a new perspective by proposing *denoising as decoding*, shifting from single-step reconstruction to iterative refinement. Specifically, we replace the decoder with a diffusion process that iteratively refines noise to recover the original image, guided by the latents provided by the encoder. We evaluate our approach by assessing both reconstruction (rFID) and generation quality (FID), comparing it to state-of-the-art autoencoding approaches. By adopting iterative reconstruction through diffusion, our autoencoder, namely ϵ-VAE, achieves high reconstruction quality, which in turn enhances downstream generation quality by 22% at the same compression rates or provides 2.3$\times$ inference speedup through increasing compression rates. We hope this work offers new insights into integrating iterative generation and autoencoding for improved compression and generation.

1. Introduction

Two dominant paradigms in modern visual generative modeling are autoregression (Radford et al., 2018) and diffusion (Ho et al., 2020). Tokenization is essential for both: discrete tokens allow step-by-step conditional generation in autoregressive models, while continuous latents enable efficient learning in the denoising process of diffusion models. In either case, empirical results demonstrate that tokenization enhances generative performance. Here, we focus on *continuous* tokenization for latent diffusion models, which excel at generating high-dimensional visual data.

In this paper, we revisit the conventional autoencoding pipeline, which typically consists of an encoder that compresses the input into a latent representation and a decoder that reconstructs the original data in a single step. Instead of a deterministic decoder, we introduce a diffusion process (Ho et al., 2020; Song et al., 2021), where the encoder still compresses the input into a latent representation, but reconstruction is performed iteratively through denoising. This reframing turns the reconstruction phase into a progressive refinement process, where the diffusion model, guided by the latent representation, gradually restores the original data. While previous work (Preechakul et al., 2022) and concurrent work (Birodkar et al., 2024) have explored diffusion mechanisms in autoencoding, none have fully realized a practical diffusion-based autoencoder. By carefully co-designing architecture and objectives, we firstly show that our approach outperforms state-of-the-art autoencoding paradigms in reconstruction fidelity, sampling efficiency, and resolution generalization.

To effectively implement our approach, several key design factors must be carefully considered. First, the architectural design must ensure that the diffusion decoder is effectively conditioned on the encoder latent representations. Second, the training objectives should leverage synergies with traditional autoencoding losses, such as LPIPS (Zhang et al., 2018) and GAN (Esser et al., 2021). Finally, diffusion-specific design choices play a crucial role, including: (1) model parameterization, which defines the prediction target for the diffusion decoder; (2) noise scheduling, which shapes the optimization trajectory; and (3) the distribution of time steps during training and testing, which balances noise levels for effective learning and generation. Our study systematically examines these components through controlled experiments, demonstrating their impact on achieving a high-performing diffusion-based autoencoder. We show in the experiments that under the standard configuration (Rombach et al., 2022), our method obtains a 40% improvement in

*Work done as a student researcher at Google. [1]Google DeepMind [2]City University of Hong Kong. Correspondence to: Long Zhao <longzh@google.com>, Xuhui Jia <xhjia@google.com>, Ting Liu <liuti@google.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

terms of reconstruction quality, leading to 22% better image generation quality. More notably, we achieve 2.3× higher inference throughput by increasing compression rates, while keeping competitive generation quality.

In summary, our contributions are as follows: (1) introducing a novel approach that fully leverages the capabilities of diffusion decoders for more practical diffusion-based autoencoding, achieving strong rFID, high sampling efficiency (within 1 to 3 steps), and robust resolution generalization; (2) presenting key design choices in both architecture and objectives to optimize performance; and (3) conducting extensive controlled experiments that demonstrate our method achieves high-quality reconstruction and generation results, outperforming leading visual auto-encoding paradigms.

2. Background

We start by briefly reviewing the basic concepts required to understand the proposed method. A more detailed summary of related work is deferred to Appx. B.

Visual autoencoding. To achieve efficient and scalable high-resolution image synthesis, common generative models, including autoregressive models (Razavi et al., 2019; Esser et al., 2021; Chang et al., 2022) and diffusion models (Rombach et al., 2022), are typically trained in a low-resolution latent space by first downsampling the input image using a tokenizer. The tokenizer is generally implemented as a convolutional autoencoder consisting of an encoder, $\mathcal{E}$, and a decoder, $\mathcal{G}$. Specifically, the encoder, $\mathcal{E}$, compresses an input image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times 3}$ into a set of latent codes (*i.e.*, tokens), $\mathcal{E}(\boldsymbol{x}) = \boldsymbol{z} \in \mathbb{R}^{H/f\times W/f\times n_z}$, where f is the downsampling factor and n_z is the latent channel dimensions. The decoder, $\mathcal{G}$, then reconstructs the input from $\boldsymbol{z}$, such that $\mathcal{G}(\boldsymbol{z}) = \boldsymbol{x}$.

Training an autoencoder primarily involves several losses: reconstruction loss $\mathcal{L}_{\text{rec}}$, perceptual loss (LPIPS) $\mathcal{L}_{\text{LPIPS}}$, and adversarial loss $\mathcal{L}_{\text{adv}}$. The reconstruction loss minimizes pixel differences (*i.e.*, typically measured by the ℓ_1 or ℓ_2 distance) between $\boldsymbol{x}$ and $\mathcal{G}(\boldsymbol{z})$. The LPIPS loss (Zhang et al., 2018) enforces high-level structural similarities between inputs and reconstructions by minimizing differences in their intermediate features extracted from a pre-trained VGG network (Simonyan & Zisserman, 2015). The adversarial loss (Esser et al., 2021) introduces a discriminator, $\mathcal{D}$, which encourages more photorealistic outputs by distinguishing between real images, $\mathcal{D}(\boldsymbol{x})$, and reconstructions, $\mathcal{D}(\mathcal{G}(\boldsymbol{z}))$. The final training objective is a weighted combination of these losses:

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{LPIPS}} \cdot \mathcal{L}_{\text{LPIPS}} + \lambda_{\text{adv}} \cdot \mathcal{L}_{\text{adv}}, \tag{1}$$

where the λ values are weighting coefficients. In this paper, we consider the autoencoder optimized by Eq. 1 as our main competing baseline (Esser et al., 2021), as it has become a standard tokenizer training scheme widely adopted in state-of-the-art image and video generative models (Chang et al., 2022; Rombach et al., 2022; Yu et al., 2022; 2023; Kondratyuk et al., 2024; Esser et al., 2024).

Diffusion. Given a data distribution $p_{\boldsymbol{x}}$ and a noise distribution $p_{\boldsymbol{\epsilon}}$, a diffusion process progressively corrupts clean data $\boldsymbol{x}_0 \sim p_{\boldsymbol{x}}$ by adding noise $\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}}$ and then reverses this corruption to recover the original data (Song & Ermon, 2019; Ho et al., 2020), represented as:

$$\boldsymbol{x}_t = \alpha_t \cdot \boldsymbol{x}_0 + \sigma_t \cdot \boldsymbol{\epsilon}, \tag{2}$$

where $t \in [0, \mathrm{T}]$ and $\boldsymbol{\epsilon}$ is drawn from a standard Gaussian distribution, $p_{\boldsymbol{\epsilon}} = \mathcal{N}(0, I)$. The functions α_t and σ_t govern the trajectory between clean data and noise, affecting both training and sampling. The basic parameterization in Ho et al. (2020) defines $\sigma_t = \sqrt{1-\alpha_t^2}$ with $\alpha_t = \left(\prod_{s=0}^{t}(1-\beta_s)\right)^{\frac{1}{2}}$ for discrete timesteps. The diffusion coefficients β_t are linearly interpolated values between β_0 and β_{T-1} as $\beta_t = \beta_0 + \frac{t}{T-1}(\beta_{T-1} - \beta_0)$, with start and end values are set empirically.

The forward and reverse diffusion processes are described by the following factorizations:

$$\begin{aligned} q(\boldsymbol{x}_{\Delta t:\mathrm{T}}|\boldsymbol{x}_0) &= \prod_{i=1}^{\mathrm{T}} q(\boldsymbol{x}_{i\cdot\Delta t}|\boldsymbol{x}_{(i-1)\cdot\Delta t}) \\ \text{and}\quad p(\boldsymbol{x}_{0:\mathrm{T}}) &= p(\boldsymbol{x}_{\mathrm{T}}) \prod_{i=1}^{\mathrm{T}} p(\boldsymbol{x}_{(i-1)\cdot\Delta t}|\boldsymbol{x}_{i\cdot\Delta t}), \end{aligned} \tag{3}$$

where the forward process $q(\boldsymbol{x}_{\Delta t:\mathrm{T}}|\boldsymbol{x}_0)$ transitions clean data $\boldsymbol{x}_0$ to noise $\boldsymbol{x}_{\mathrm{T}} = \boldsymbol{\epsilon}$, while the reverse process $p(\boldsymbol{x}_{0:\mathrm{T}})$ recovers clean data from noise. Δt denotes the time step interval or step size.

During training, the model learns the score function $\nabla \log p_t(\boldsymbol{x}) \propto -\frac{\epsilon}{\sigma_t}$, which represents gradient pointing toward the data distribution along the noise-to-data trajectory. In practice, the model $s_\Theta(\boldsymbol{x}_t, t)$ is optimized by minimizing the score-matching objective:

$$\mathcal{L}_{\text{score}} = \min_{\Theta} \mathbb{E}_{t\sim\pi(t),\epsilon\sim\mathcal{N}(0,I)} \left[w_t \| \sigma_t s_\Theta(\boldsymbol{x}_t, t) + \boldsymbol{\epsilon} \|^2 \right], \tag{4}$$

where $\pi(t)$ defines the time-step sampling distribution and w_t is a time-dependent weight. These elements together influence which time steps or noise levels are prioritized during training. Conceptually, the diffusion model learns the tangent of the trajectory at each point along the path. During sampling, it progressively recovers clean data from noise based on its predictions.

Rectified flow provides a specific parametrization of α_t and σ_t such that the trajectory between data and noise follows a

"straight" path (Liu et al., 2023; Albergo & Vanden-Eijnden, 2023). This trajectory is represented as:

$$\boldsymbol{x}_t = (1-t)\cdot\boldsymbol{x}_0 + t\cdot\boldsymbol{\epsilon}, \tag{5}$$

where $t \in [0,1]$. In this formulation, the gradient along the trajectory, $\boldsymbol{\epsilon} - \boldsymbol{x}_0$, is deterministic, often referred to as the velocity. The model $v_\Theta(\boldsymbol{x}_t, t)$ is parameterized to predict velocity by minimizing:

$$\min_\Theta \mathbb{E}_{t\sim\pi(t),\boldsymbol{\epsilon}\sim\mathcal{N}(0,I)}\left[\|v_\Theta(\boldsymbol{x}_t,t) - (\boldsymbol{\epsilon}-\boldsymbol{x})\|^2\right]. \tag{6}$$

We note that this objective is equivalent to a score matching form (Eq. 4), with the weight $w_t = (\frac{1}{1-t})^2$. This equivalence highlights that alternative model parameterizations reduce to a standard denoising objective, where the primary difference lies in the time-dependent weighting functions and the corresponding optimization trajectory (Kingma & Gao, 2024).

During sampling, the model follows a simple probability flow ODE:

$$\mathrm{d}\boldsymbol{x}_t = v_\Theta(\boldsymbol{x}_t, t)\cdot\mathrm{d}t. \tag{7}$$

Although a perfect straight path could theoretically be solved in a single step, the independent coupling between data and noise often results in curved trajectories, necessitating multiple steps to generate high-quality samples (Liu et al., 2023; Lee et al., 2024). In practice, we iteratively apply the standard Euler solver (Euler, 1845) to sample data from noise.

3. Method

We introduce ϵ-VAE, with an overview provided in Fig. 1. The core idea is to replace single-step, deterministic decoding with an iterative, stochastic denoising process. By reframing autoencoding as a conditional denoising problem, we anticipate two key improvements: (1) more effective generation of latent representations, allowing the downstream latent diffusion model to learn more efficiently, and (2) enhanced decoding quality due to the iterative and stochastic nature of the diffusion process.

We systematically explore the design space of model architecture, objectives, and diffusion training configurations, including noise and time scheduling. While this work primarily focuses on generating continuous latents for latent diffusion models, the concept of iterative decoding could also be extended to discrete tokens, which we leave for future exploration.

3.1. Modeling

ϵ-VAE retains the encoder $\mathcal{E}$ while enhancing the decoder $\mathcal{G}$ by incorporating a diffusion model, transforming the standard decoding process into an iterative denoising task.

Conditional denoising. Specifically, the input $\boldsymbol{x} \sim p_{\boldsymbol{x}}$ is encoded by the encoder as $\boldsymbol{z} = \mathcal{E}(\boldsymbol{x})$, and this encoding serves as a condition to guide the subsequent denoising process. This reformulates the reverse process in Eq. 3 into a conditional form (Nichol & Dhariwal, 2021):

$$p(\boldsymbol{x}_{0:\mathrm{T}}|\boldsymbol{z}) = p(\boldsymbol{x}_\mathrm{T})\prod_{i=1}^{\mathrm{T}} p(\boldsymbol{x}_{(i-1)\cdot\Delta t}|\boldsymbol{x}_{i\cdot\Delta t}, \boldsymbol{z}), \tag{8}$$

where the denoising process from the noise $\boldsymbol{x}_\mathrm{T} = \boldsymbol{\epsilon}$ to the input $\boldsymbol{x}_0 = \boldsymbol{x}$, is additionally conditioned on $\boldsymbol{z}$ over time. Here, the decoder is no longer deterministic, as the process starts from random noise. For a more detailed discussion on this autoencoding formulation, we refer readers to Appx. A.

Architecture and conditioning. We adopt the standard U-Net architecture from Dhariwal & Nichol (2021) for our diffusion decoder $\mathcal{G}$, while also exploring Transformer-based models (Peebles & Xie, 2023). For conditional denoising, we concatenate the conditioning signal with the input channel-wise, following the approach of diffusion-based super-resolution models (Ho et al., 2022; Saharia et al., 2022b). Specifically, low-resolution latents are upsampled using nearest-neighbor interpolation to match the resolution of $\boldsymbol{x}_t$, then concatenated along the channel dimension. In Appx. D, although we experimented with conditioning via AdaGN (Nichol & Dhariwal, 2021), it did not yield significant improvement and introduced additional overhead, so we adopt channel concatenation.

3.2. Objectives

We adopt the standard autoencoding objective from Eq. 1 to train ϵ-VAE, with a key modification: replacing the reconstruction loss $\mathcal{L}_{\text{rec}}$ used for the standard decoder with the score-matching loss $\mathcal{L}_{\text{score}}$ for training the diffusion decoder. Additionally, we introduce a strategy to adjust the perceptual $\mathcal{L}_{\text{LPIPS}}$ and adversarial $\mathcal{L}_{\text{adv}}$ losses to better align with the diffusion decoder training.

Velocity prediction. We adopt the rectified flow parameterization, utilizing a linear optimization trajectory between data and noise, combined with velocity-matching objective (Eq. 6). We inject the encoder output $\boldsymbol{z}$ into the objective by replacing $v_\Theta(\boldsymbol{x}_t, t)$ with $\mathcal{G}(\boldsymbol{x}_t, t, \boldsymbol{z})$.

Perceptual matching. The LPIPS loss (Zhang et al., 2018) minimizes the perceptual distance between the reconstructions and real images using pre-trained models, typically VGG network (Esser et al., 2021; Yu et al., 2023; 2022). We apply this feature-matching objective to train ϵ-VAE. However, unlike traditional autoencoders, ϵ-VAE predicts velocity instead of directly reconstructing the image during training, making it infeasible to compute the LPIPS loss directly between the prediction and the target image. To address this, we leverage the simple reversing step from

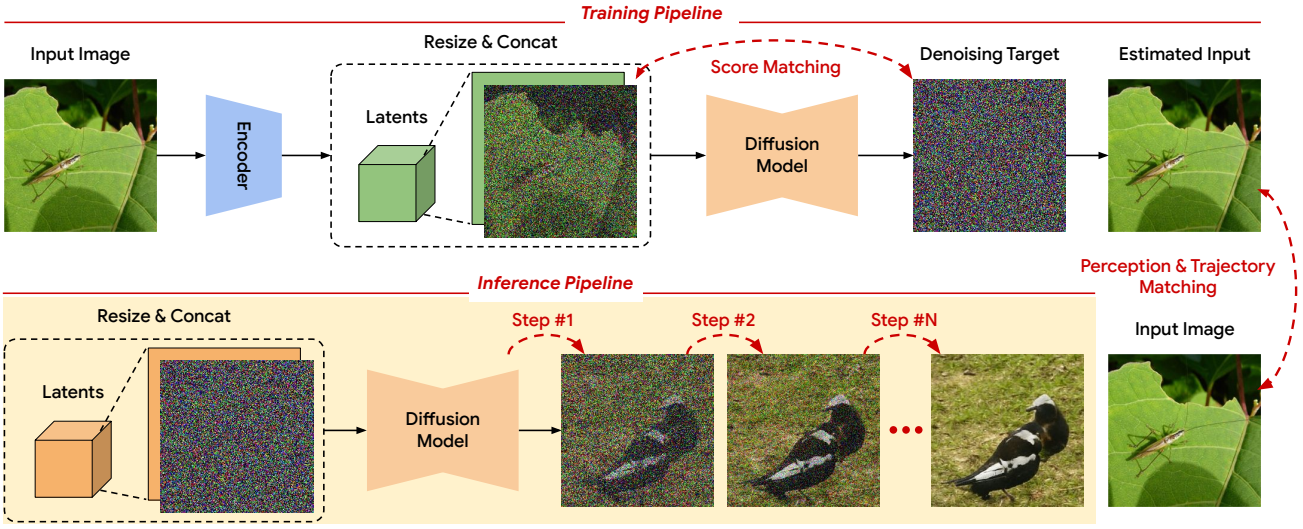


Figure 1. **An overview of ϵ-VAE**. We frame visual decoding as an iterative denoising problem by replacing the autoencoder decoder with a diffusion model, optimized using a combination of score, perception, and trajectory matching losses. During inference, images are reconstructed (or generated) from encoded (or sampled) latents through an iterative denoising process. The number of sampling steps N can be flexibly adjusted within small NFE regimes (from 1 to 3). We empirically confirm that ϵ-VAE significantly outperforms the standard VAE schema, even with just a few steps.

Eq. 6 to estimate $\boldsymbol{x}_0$ from the prediction and $\boldsymbol{x}_t$ as follows:

$$\hat{\boldsymbol{x}}_0^t = \boldsymbol{x}_t - t \cdot \mathcal{G}(\boldsymbol{x}_t, t, \boldsymbol{z}), \tag{9}$$

where $\hat{\boldsymbol{x}}_0^t$ represents the reconstructed image estimated by the model at time t. We then compute the LPIPS loss between $\hat{\boldsymbol{x}}_0^t$ and the target real image $\boldsymbol{x}$.

Denoising trajectory matching. The adversarial loss encourages photorealistic outputs by comparing the reconstructions to real images. We modify this to better align with a diffusion decoder. Specifically, our approach adapts the standard adversarial loss to enforce trajectory consistency rather than solely on realism. In practice, we achieve this by minimizing the following divergence, $\mathcal{D}_{\text{adv}}$:

$$\min_{\Theta} \mathbb{E}_{t\sim p_t} \left[\mathcal{D}_{\text{adv}} \left(q(\boldsymbol{x}_0|\boldsymbol{x}_t) || p_\Theta(\hat{\boldsymbol{x}}_0^t|\boldsymbol{x}_t) \right) \right], \tag{10}$$

where $\mathcal{D}_{\text{adv}}$ is a probability distance metric (Goodfellow et al., 2014; Arjovsky et al., 2017), and we adopt the basic non-saturating GAN (Goodfellow et al., 2014).

For adversarial training, we design a time-dependent discriminator that takes time as input using AdaGN approach (Dhariwal & Nichol, 2021). To simulate the trajectory, we concatenate $\boldsymbol{x}_0$ and $\boldsymbol{x}_t$ along the channel dimension. The generator parameterized by Θ, and the discriminator, parameterized by Φ, are then optimized through a minimax game as:

$$\begin{aligned} \min_{\Theta} \max_{\Phi} \mathcal{L}_{\text{adv}} = \mathbb{E}_{q(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[\log \mathcal{D}_\Phi(\boldsymbol{x}_0, \boldsymbol{x}_t, t) \right] \\ + \mathbb{E}_{p_\Theta(\hat{\boldsymbol{x}}_0^t|\boldsymbol{x}_t)} \left[\log \left(1 - \mathcal{D}_\Phi(\hat{\boldsymbol{x}}_0^t, \boldsymbol{x}_t, t) \right) \right], \end{aligned} \tag{11}$$

where fake trajectories $p_\Theta(\hat{\boldsymbol{x}}_0^t|\boldsymbol{x}_t)$ are contrasted with real trajectories $q(\boldsymbol{x}_0|\boldsymbol{x}_t)$. To further stabilize training, we apply the R_1 gradient penalty to the discriminator parameters (Mescheder et al., 2018). In Appx. D, we explore alternative matching approaches, including the standard adversarial method of comparing individual reconstructions $\hat{\boldsymbol{x}}_0^t$ with real images $\boldsymbol{x}_0$, matching the trajectory steps $\boldsymbol{x}_t \to \boldsymbol{x}_{t-\Delta t}$ (Xiao et al., 2022; Wang et al., 2024a), and our start-to-end trajectory matching $\boldsymbol{x}_t \to \boldsymbol{x}_0$, with the latter showing the best performance.

Final training objective combines $\mathcal{L}_{\text{score}}$, $\mathcal{L}_{\text{LPIPS}}$, and $\mathcal{L}_{\text{adv}}$, with empirically adjusted weights (see Appx. C.2).

Note that applying LPIPS and adversarial losses on the estimated one-step sample could lead to potential objective bias. However, we would like to emphasize that ϵ-VAE differs significantly from traditional diffusion models in that its diffusion decoder is conditioned on encoded latents $\boldsymbol{z}$. This conditioning provides a strong prior about the input image to reconstruct, resulting in a more accurate estimated one-step sample than in typical diffusion scenarios. Therefore, we believe the potential for objective bias is considerably reduced in ϵ-VAE. Fine-tuning the diffusion decoder with frozen $\boldsymbol{z}$ like Sargent et al. (2025) could be a promising avenue for further improvement, which we will explore in our future work.

3.3. Noise and time scheduling

Noise scheduling. In diffusion models, noise scheduling involves progressively adding noise to the data over time

by defining specific functions for α_t and σ_t in Eq. 2. This process is crucial as it determines the signal-to-noise ratio, $\lambda_t = \frac{\alpha_t^2}{\sigma_t^2}$, which directly influences training dynamics. Noise scheduling can also be adjusted by scaling the intermediate states $\boldsymbol{x}_t$ with a constant factor $\gamma \in (0, 1]$, which shifts the signal-to-noise ratio downward. This makes training more challenging over time while preserving the shape of the trajectory (Chen, 2023).

In this work, we define α_t and σ_t according to rectified flow formulation, while also scaling $\boldsymbol{x}_t$ by γ, with the value chosen empirically. However, when $\gamma \neq 1$, the variance of $\boldsymbol{x}_t$ changes, which can degrade performance (Karras et al., 2022). To address this, we normalize the denoising input $\boldsymbol{x}_t$ by its variance after scaling, ensuring it preserves unit variance over time (Chen, 2023).

Time scheduling. Another important aspect in diffusion models is time scheduling for both training and sampling, controlled by $\pi(t)$ during training and Δt during sampling, as outlined in Eq. 3 and Eq. 4. A common choice for $\pi(t)$ is the uniform distribution $\mathcal{U}(0, T)$, which applies equal weight to each time step during training. Similarly, uniform time steps $\Delta t = \frac{1}{T}$ are typically used for sampling. However, to improve model performance on more challenging time steps and focus on noisy regimes during sampling, the time scheduling strategy should be adjusted accordingly.

In this work, we sample t from a logit-normal distribution (Atchison & Shen, 1980), which emphasizes intermediate timesteps (Esser et al., 2024). During sampling, we apply a reversed logarithm mapping $\rho_{\log}$ defined as:

$$\rho_{\log}(t; m, n) = \frac{\log(m) - \log\left(t \cdot (m - n) + n\right)}{\log(m) - \log(n)}, \quad (12)$$

where we set $m = 1$ and $n = 100$, resulting in denser sampling steps early in the inference process.

4. Experiments

We evaluate the effectiveness of ϵ-VAE on image reconstruction and generation tasks using the ImageNet (Deng et al., 2009). The VAE formulation by Esser et al. (2021) serves as a strong baseline due to its widespread use in modern image generative models (Rombach et al., 2022; Peebles & Xie, 2023; Esser et al., 2024). We perform controlled experiments to compare reconstruction and generation quality by varying model scale, latent dimension, downsampling rates, and input resolution.

Model configurations. We use the encoder and discriminator architectures from VQGAN (Esser et al., 2021) and keep consistent across all models. The decoder design follows BigGAN (Brock et al., 2019) for VAE and from ADM (Dhariwal & Nichol, 2021) for ϵ-VAE. Additionally, we experiment with the DiT architecture (Peebles & Xie, 2023) for ϵ-VAE. To evaluate model scaling, we test five decoder variants: base (B), medium (M), large (L), extra-large (XL), and huge (H), by adjusting width and depth accordingly. Further model specifications are provided in Appx. C.1.

We experiment with the following two encoder configurations. **ϵ-VAE-lite**: a light-weight version with 6M parameters, a downsampling rate of 16, and 8 latent channels; **ϵ-VAE-SD**: a standard version based on Stable Diffusion with 34M parameters, a downsampling rate of 8, and 4 latent channels. ϵ-VAE-lite is intentionally designed as a more challenging setup and serves as the primary focus of analysis in the paper. For this configuration, we further explore downsampling rates of 4, 8, and 32, as well as latent dimensions of 4, 16, and 32 channels. Both VAE and ϵ-VAE are trained to reconstruct 128×128 images under these controlled conditions. Additionally, we validate our method in the standard setup of ϵ-VAE-SD, where we compare it against state-of-the-art VAEs.

Evaluation. We evaluate the autoencoder on both reconstruction and generation quality using Fréchet Inception Distance (FID) (Heusel et al., 2017) as the primary metric, and we also report PSNR and SSIM as secondary metrics. For reconstruction quality (rFID), FID is computed at both training and higher resolutions to assess generalization across resolutions. For generation quality (FID), we generate latents from the trained autoencoders and use them to train the DiT-XL/2 latent generative model (Peebles & Xie, 2023). This latent model remains fixed across all generation experiments, meaning improved autoencoder latents directly enhance generation quality.

4.1. Reconstruction quality

Decoder architecture. We explore two major architectural designs: the UNet-based architecture from ADM (Dhariwal & Nichol, 2021) and the Transformer-based DiT (Peebles & Xie, 2023). We compare various model sizes–ADM:{B, M, L, XL, H} and DiT:{S, B, L, XL} with patch sizes of {4, 8}. The results are summarized in Fig. 2 (left). ADM consistently outperforms DiT across the board. While we observe rFID improvements in DiT when increasing the number of tokens by reducing patch size, this comes with significant computational overhead. The overall result aligns with the original design intentions: ADM for pixel-level generation and DiT for latent-level generation. For the following experiments, we use the ADM architecture for our diffusion decoder.

Compression rate. Compression can be achieved by adjusting either the channel dimensions of the latents or the downsampling factor of the encoder. In Fig. 2 (middle and right), we compare VAE and ϵ-VAE across these two aspects. The results show that ϵ-VAE consistently outperforms VAE

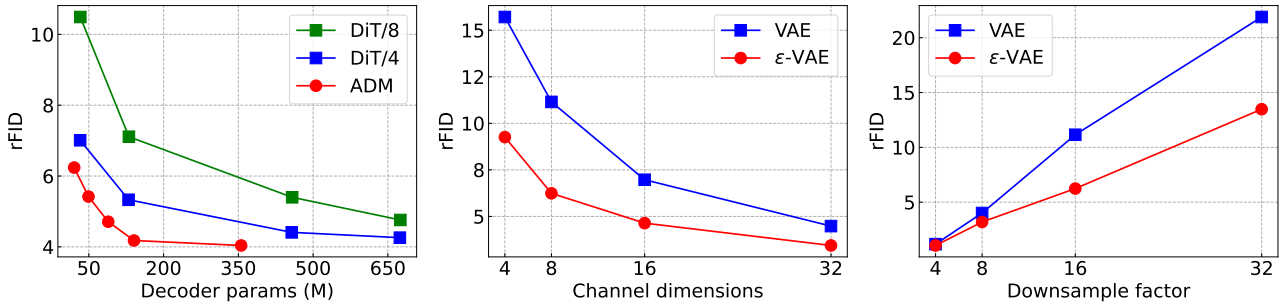


Figure 2. **Architecture and compression analysis.** The ϵ-VAE decoder uses either a UNet-based ADM or Transformer-based DiT (left). ϵ-VAE and VAE are evaluated under different compression rates by varying latent channel dimensions (middle) or encoder downsampling factors (right). *We follow the ϵ-VAE-lite configuration in these experiments.*

Table 1. **ImageNet reconstruction results (rFID) at different resolutions using VAEs trained at** 128×128**.** † denotes training at 128×128 followed by fine-tuning at a higher resolution.

Resolution (ImageNet)	**128**	**256**	**512**	**256**†
SD-VAE (Rombach et al., 2022)	4.54	1.21	0.91	0.86
LiteVAE (Sadat et al., 2024)	4.40	0.97	-	0.73
ϵ-VAE-SD (B)	1.94	0.65	0.61	0.52
ϵ-VAE-SD (M)	1.58	0.55	0.53	0.47
ϵ-VAE-SD (L)	1.47	0.52	0.41	0.45
ϵ-VAE-SD (XL)	1.34	0.49	0.39	0.43
ϵ-VAE-SD (H)	1.00	0.44	0.35	0.38

in terms of rFID, particularly as the compression ratio increases. Specifically, as shown on the middle graph, ϵ-VAE achieves lower rFIDs than VAE across all channel dimensions, with a notable gap at lower dimensions (4 and 8). On the right graph, ϵ-VAE maintains lower rFIDs than VAE even as the downsampling factor increases, with the gap widening significantly at larger factors (16 and 32). Furthermore, ϵ-VAE delivers comparable or superior rFIDs even when the compression ratio is doubled, demonstrating its robustness and effectiveness in high-compression scenarios.

Resolution generalization. A notable feature of conventional autoencoders is their capacity to generalize and reconstruct images at higher resolutions during inference (Rombach et al., 2022). To assess this, we conduct inference on images with resolutions of 256×256 and 512×512, using ϵ-VAE and VAE models trained at 128×128. As shown in Tab. 1, ϵ-VAE effectively generalizes to higher resolutions, consistently preserving its performance advantage over other VAEs. Furthermore, we find that fine-tuning models at the target (higher) resolution leads to improvement at it, which is consistent with the observation made by Sadat et al. (2024). We hence utilize this multi-stage training strategy in the following experiments when the target resolution is larger than 128×128.

Comparisons to state-of-the-art VAEs. We provide image reconstruction results under the same configuration as VAEs in Stable Diffusion (SD-VAE): an encoder with 34M parameters and a channel dimension of 4 for 256×256 image reconstruction. We evaluate rFID, PSNR and SSIM on the full validation sets of ImageNet and COCO-2017 (Lin et al., 2014), with the results summarized in Tab. 2. Our finds reveal that ϵ-VAE outperforms state-of-the-art VAEs when the decoder sizes are comparable, and its performance can be further improved by scaling up the decoder. This demonstrates the strong model scalability of our framework.

One-step ϵ-VAE. Note that the denoising process of ϵ-VAE demonstrates promising results even with a single iteration. To show this, we provide a direct comparison between SD-VAE and our one-step ϵ-VAE models in Tab. 3. This table presents image reconstruction quality on ImageNet 256×256 with the 8×8 downsampling factor. As shown, both ϵ-VAE (B) and ϵ-VAE (M) outperform SD-VAE across all metrics. These results confirm the effectiveness and efficiency of our one-step models compared to SD-VAE. Consequently, this allows ϵ-VAE to be adapted for scenarios with latency-sensitive requirements, *e.g.*, real-time visualization during image generation, by reducing the decoding step to a single pass.

4.2. Class-conditional image generation

We now evaluate the generative performance of ϵ-VAE when combined with latent diffusion models (Rombach et al., 2022). We perform standard class-conditional image generation tasks using the DiT-XL/2 model as our latent generative model (Peebles & Xie, 2023). Further details on the training setup are provided in Appx. C.3. Tab. 4 presents the image generation results of ϵ-VAE and other competing VAEs at resolutions of 256×256. The results show that ϵ-VAE consistently outperforms other VAEs across different dowmsampling factors. In addition, we emphasize that ϵ-VAE achieves favorable generation quality while using only 25% of the token length typically required by SD-VAE.

Table 2. **Comparisons with state-of-the-art image autoencoders.** All results are computed on 256×256 ImageNet 50K validation set and COCO-2017 5K validation set. ϵ-VAE-SD (M) achieves better reconstruction quality while having similar parameters (49M) in the decoder with other VAEs. Further improvements are obtained after we scale up to ϵ-VAE-SD (H) which has 355M decoder parameters.

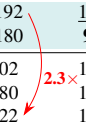
Downsample factor	Method	Discrete latent	Latent dim.	ImageNet rFID↓	ImageNet PSNR↑	ImageNet SSIM↑	COCO rFID↓	COCO PSNR↑	COCO SSIM↑
16×16	VQGAN (Esser et al., 2021)	✓	256	4.99	20.00	0.629	12.29	19.57	0.630
	MaskGIT (Chang et al., 2022)	✓	256	2.28	-	-	-	-	-
	LlamaGen (Sun et al., 2024)	✓	8	2.19	20.79	0.675	8.11	20.42	0.678
	SD-VAE (Rombach et al., 2022)	✗	4	2.93	20.57	0.662	8.89	19.95	0.670
	ϵ-VAE-SD (M)	✗	4	1.91	21.27	0.693	6.12	22.38	0.718
	ϵ-VAE-SD (H)	✗	4	**1.35**	**22.60**	**0.711**	**4.18**	**24.26**	**0.830**
8×8	VQGAN (Esser et al., 2021)	✓	4	1.19	23.38	0.762	5.89	23.08	0.771
	ViT-VQGAN (Yu et al., 2022)	✓	32	1.28	-	-	-	-	-
	LlamaGen (Sun et al., 2024)	✓	8	0.59	24.45	0.813	4.19	24.20	0.822
	SD-VAE (Rombach et al., 2022)	✗	4	0.74	25.68	0.820	4.45	25.41	0.831
	SDXL-VAE (Podell et al., 2024)	✗	4	0.68	26.04	0.834	4.07	25.76	0.845
	LiteVAE (Sadat et al., 2024)	✗	4	0.87	26.02	0.740	-	-	-
	ϵ-VAE-SD (M)	✗	4	0.47	27.65	0.841	3.98	25.88	0.850
	ϵ-VAE-SD (H)	✗	4	**0.38**	**29.49**	**0.851**	**3.65**	**26.01**	**0.856**

Table 3. **Image reconstruction results of one-step ϵ-VAE and SD-VAE on ImageNet** 256×256**.** A downsampling factor of 8×8 is used for comparison. We include two variants of our model in the results: ϵ-VAE-SD (B), which has a similar inference speed to SD-VAE, and ϵ-VAE-SD (M), which matches SD-VAE in the number of parameters.

Method	rFID↓	PSNR↑	SSIM↑
SD-VAE (Rombach et al., 2022)	0.74	25.68	0.820
ϵ-VAE-SD (B)	0.57	25.91	0.826
ϵ-VAE-SD (M)	**0.51**	**26.45**	**0.830**

Table 4. **Benchmarking class-conditional image generation on ImageNet** 256×256**.** We use the DiT-XL/2 architecture (Esser et al., 2024) for latent diffusion models, and we do not apply classifier-free guidance (Ho & Salimans, 2022).

Downsample factor	Method	Throughput (image/ks)	FID↓
32×32	SD-VAE (Rombach et al., 2022)	3991	21.31
	ϵ-VAE-SD (M)	3865	15.98
	ϵ-VAE-SD (H)	3870	**14.26**
16×16	SD-VAE (Rombach et al., 2022)	1220	14.59
	ϵ-VAE-SD (M)	1192	10.68
	ϵ-VAE-SD (H)	1180	**9.72**
8×8	Asym-VAE (Zhu et al., 2023)	502	10.85
	Omni-VAE (Wang et al., 2024b)	480	12.25
	SD-VAE (Rombach et al., 2022)	522	11.63
	ϵ-VAE-SD (M)	491	9.39
	ϵ-VAE-SD (H)	477	**8.85**



This token length reduction significantly accelerates latent diffusion model generation, leading to 2.3× higher inference throughput while maintaining competitive generation quality. These results confirm that the performance gains from the reconstruction task successfully transfer to the generation task, further validating the effectiveness of ϵ-VAE.

More importantly, ϵ-VAE-SD achieves around 25% improvement in generation quality over SD-VAE at the 32×32 downsampling factor, alongside a 3.2× inference speedup than SD-VAE at the 16×16 downsampling factor with comparable FID. We observed similar training speedups for latent diffusion models utilizing ϵ-VAE at this higher downsampling rate. These gains are more pronounced than those observed when increasing the downsampling factor from 8×8 to 16×16. These findings strongly suggest that the benefits of ϵ-VAE and latent diffusion pipeline could be amplified with higher downsampling factors.

An additional advantage of scaling the autoencoder over the latent model lies in computational efficiency. Recent trends show latent diffusion models increasingly adopt Transformer architectures (Peebles & Xie, 2023), where self-attention scales quadratically with input resolution. In contrast, our convolution-based UNet decoder offers more favorable linear scaling. As models grow, shifting complexity to the autoencoder helps reduce the burden on the latent model, leading to a more efficient overall system.

4.3. Ablation studies

We conduct a component-wise analysis to validate our key design choices, focusing on three critical aspects: architecture, objectives, and noise & time scheduling. We evaluate the reconstruction quality (rFID) and sampling efficiency (NFE). The results are summarized in Tab. 5.

Baseline. Our evaluation begins with a baseline model: an autoencoder with a diffusion decoder, trained solely using the score matching objective. This baseline follows the vanilla diffusion setup from Ho et al. (2020), including their

Table 5. **Ablation study on key design choices for the ϵ-VAE diffusion decoder.** A systematic evaluation of the proposed architecture ($\star$), objectives ($\dagger$), and noise & time scheduling ($\S$). Each row progressively modifies or builds upon the baseline decoder, showing improvements in performance. *The results are computed under the ϵ-VAE-lite configuration.*

Ablation	NFE↓	rFID↓
Baseline: DDPM-based diffusion decoder	1,000	28.22
† *(a)* Diffusion $\rightarrow$ Rectified flow parameterization	100	24.11
§ *(b)* Uniform $\rightarrow$ Logit-normal time step sampling during training	50	23.44
* *(c)* DDPM UNet $\rightarrow$ ADM UNet	50	22.04
† *(d)* Perceptual matching on $\hat{\boldsymbol{x}}_0^t$ and $\boldsymbol{x}_0$	10	11.76
† *(e)* Adversarial denoising trajectory matching on $(\hat{\boldsymbol{x}}_0^t, \boldsymbol{x}_t)$ and $(\boldsymbol{x}_0, \boldsymbol{x}_t)$	5	8.24
§ *(f)* Scale $\boldsymbol{x}_t$ by $\gamma = 0.6$	5	7.08
§ *(g)* Uniform $\rightarrow$ Reversed logarithm time spacing during inference	3	6.24

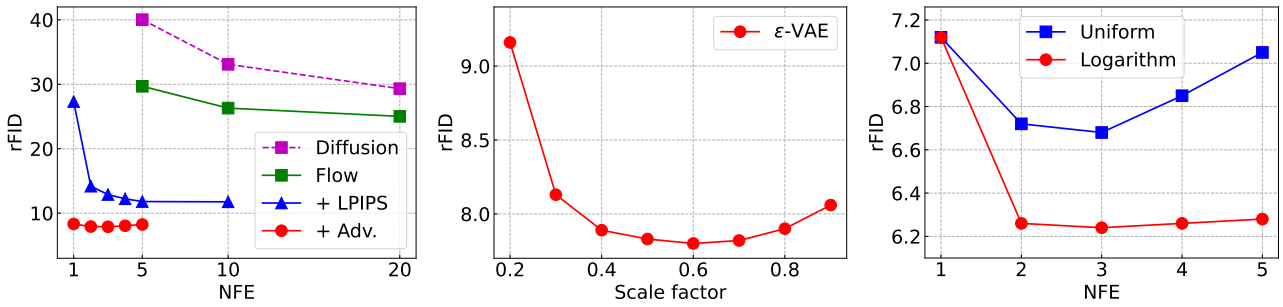


Figure 3. **Impact of our major diffusion decoder designs.** Improved training objectives, particularly perceptual matching loss and adversarial denoising trajectory matching loss, significantly contribute to better rFID scores and NFE (left). Effective noise scheduling by modulating the scaling factor γ further enhances rFID, with an optimum value of 0.6 in our experiments (middle). Lastly, adjusting time step spacing during inference ensures stable sampling in low NFE regimes (right). *We report results under the ϵ-VAE-lite configuration.*

UNet architecture, parameterization, and training configurations, while extending to a conditional form as described in Eq. 8. Building on this baseline, we progressively introduce updates and evaluate the impact of our proposed method.

Impact of proposals. In **(a)**, transitioning from standard diffusion to rectified flow (Liu et al., 2023) straightens the optimization path, resulting in significant gains in rFID and NFE. In **(b)**, adopting a logit-normal time step distribution optimizes rectified flow training (Esser et al., 2024), further improving both rFID and NFE. In **(c)**, updates to the UNet architecture (Nichol & Dhariwal, 2021) contribute to enhanced rFID scores. In **(d)**, LPIPS loss is applied to match reconstructions $\hat{\boldsymbol{x}}_0^t$ with real images $\boldsymbol{x}_0$. In **(e)**, adversarial trajectory matching loss aligns $(\hat{\boldsymbol{x}}_0^t, \boldsymbol{x}_t)$ with $(\boldsymbol{x}_0, \boldsymbol{x}_t)$, the target transition in rectified flow. Both objectives improve model understanding of the underlying optimization trajectory, significantly enhancing rFID scores and NFE.

Up to this point, with the full implementation of Eq. 1, we can compare our proposal with the VAE (B) model, which achieves an rFID score of 11.15. Our model, with a score of 8.24, already surpasses this baseline. We further improve performance by optimizing noise and time scheduling within our framework, as described next.

In **(f)**, scaling $\boldsymbol{x}_t$ reduces the signal-to-noise ratio (Chen, 2023), presenting challenges for more effective learning during training. Fig. 3 (middle) demonstrates that a scaling factor of 0.6 produces the best results. Finally, in **(g)**, reversed logarithmic time step spacing during inference allows for denser evaluations in noisier regions. Fig. 3 (right) demonstrates that this method provides more stable sampling in the lower NFE regime compared to the original uniform spacing.

In Fig. 3 (right), reconstruction quality degrades when the number of denoising steps exceeds three. To enable large step sizes for the reverse process during inference, we introduce the denoising trajectory matching loss to implicitly model the conditional distribution $p(\boldsymbol{x}_0|\boldsymbol{x}_t)$, shifting the denoising distributions from traditional Gaussian to non-Gaussian multimodal forms (Xiao et al., 2022). However, the assumptions underlying this approach are most effective when the total number of denoising steps is small. This reveals an optimal range of one to three inference steps. The degradation beyond this range also suggests that uniform step spacing may no longer be ideal. Accordingly, we empirically explored alternative sampling strategies and found that a reversed logarithmic schedule yields improved performance, as shown in the figure.

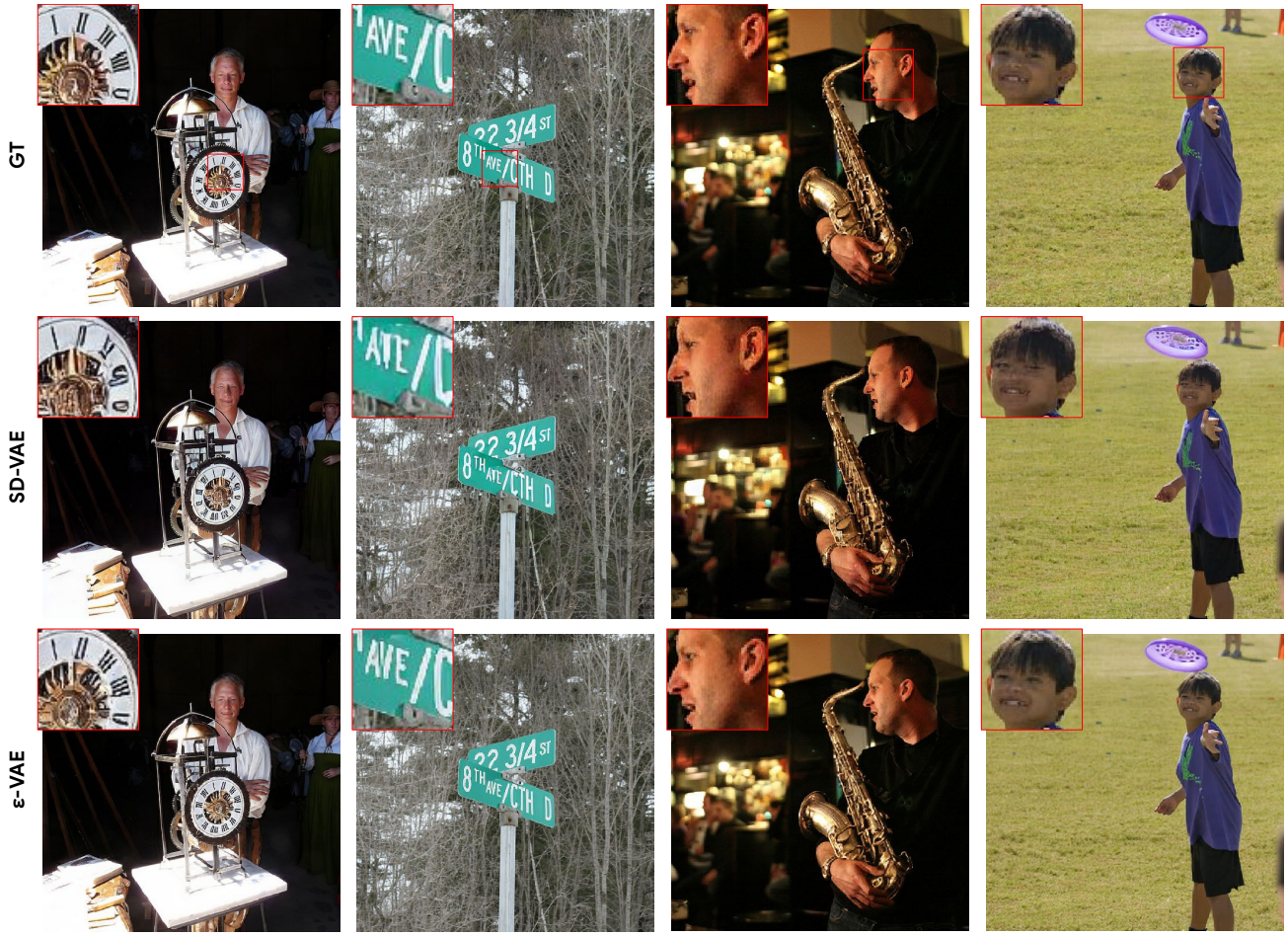


Figure 4. **Image reconstruction results under the SD-VAE configuration (Rombach et al., 2022) at the resolution of** 512×512**.** We find that ϵ-VAE produces more accurate visual details than SD-VAE in the highlighted regions with text or human face. *Best viewed when zoomed-in and in color.*

4.4. Visualization

In addition to the quantitative results, Fig. 4 shows high-resolution image reconstruction samples produced by SD-VAE (Rombach et al., 2022) and ϵ-VAE at the resolution of 512×512. We observe that reconstructed images generated by ϵ-VAE demonstrate a better visual quality than ones of SD-VAE. In particular, ϵ-VAE maintains a good visual quality for small text and human face. We provide more visual comparisons in Appx. E and throughout discussions on the major properties and advantages of ϵ-VAE compared to traditional VAEs in Appx. A.

5. Conclusion

We present ϵ-VAE, an effective visual tokenizer that introduces a diffusion decoder into standard autoencoders, turning single-step decoding into a multi-step probabilistic process. By exploring key design choices in modeling, objectives, and diffusion training, we demonstrate significant performance improvements. Our approach outperforms traditional autoencoders in both reconstruction and generation quality, particularly in high-compression scenarios. We hope our concept of iterative generation during decoding inspires further advancements in visual autoencoding.

Acknowledgements

We would like to thank Xingyi Zhou, Weijun Wang, and Caroline Pantofaru for reviewing the paper and providing feedback. We thank Rui Qian, Xuan Yang, and Mingda Zhang for helpful discussion. We also thank the Google Kauldron team for technical assistance.

Impact statement

Our work could lead to improved autoencoding techniques which have the potential to benefit generative modeling

across various perspectives, including reducing training time and memory requirements, improving visual qualities, *etc*. Although our work does not uniquely raise any new ethical challenges, visual generative modeling is a field with several ethical concerns worth acknowledging. For example, there are known issues around bias and fairness, either in the representation of generated images (Menon et al., 2020) or the implicit encoding of stereotypes (Steed & Caliskan, 2021), as well as potential risks in privacy. To ensure that the benefits of this technology are harnessed responsibly, we encourage continued open discussions in the community around the development of these new technologies.

References

Albergo, M. S. and Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. In *ICLR*, 2023.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *ICML*, pp. 214–223, 2017.

Atchison, J. and Shen, S. M. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272, 1980.

Baldridge, J., Bauer, J., Bhutani, M., Brichtova, N., Bunner, A., Chan, K., Chen, Y., Dieleman, S., Du, Y., Eaton-Rosen, Z., et al. Imagen 3. *arXiv preprint arXiv:2408.07009*, 2024.

Birodkar, V., Barcik, G., Lyon, J., Ioffe, S., Minnen, D., and Dillon, J. V. Sample what you cant compress. *arXiv preprint arXiv:2409.02529*, 2024.

Blau, Y. and Michaeli, T. The perception-distortion tradeoff. In *CVPR*, pp. 6228–6237, 2018.

Blau, Y. and Michaeli, T. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *ICML*, pp. 675–685, 2019.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

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

Chang, H., Zhang, H., Jiang, L., Liu, C., and Freeman, W. T. MaskGIT: Masked generative image transformer. In *CVPR*, pp. 11315–11325, 2022.

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

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *ICML*, pp. 1691–1703, 2020.

Chen, T. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

Chen, T., Li, L., Saxena, S., Hinton, G., and Fleet, D. J. A generalist framework for panoptic segmentation of images and videos. In *ICCV*, pp. 909–919, 2023.

Chen, Y., Girdhar, R., Wang, X., Rambhatla, S. S., and Misra, I. Diffusion autoencoders are scalable image tokenizers. *arXiv preprint arXiv:2501.18593*, 2025.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

Dhariwal, P. and Nichol, A. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021.

Ding, Z., Zhang, M., Wu, J., and Tu, Z. Patched denoising diffusion models for high-resolution image synthesis. In *ICLR*, 2024.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *CVPR*, pp. 12873–12883, 2021.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Lacey, K., Goodwin, A., Marek, Y., and Rombach, R. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Euler, L. *Institutionum calculi integralis*, volume 4. impensis Academiae imperialis scientiarum, 1845.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *NeurIPS*, 2014.

Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Fei-Fei, L., Essa, I., Jiang, L., and Lezama, J. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*, 2023.

Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. Flax: A neural network library and ecosystem for JAX, 2024. URL http://github.com/google/flax.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*, 2017.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science*, 313 (5786):504–507, 2006.

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

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

Hoogeboom, E., Agustsson, E., Mentzer, F., Versari, L., Toderici, G., and Theis, L. High-fidelity image compression with score-based generative models. *arXiv preprint arXiv:2305.18231*, 2023a.

Hoogeboom, E., Heek, J., and Salimans, T. simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, pp. 13213–13232, 2023b.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *CVPR*, pp. 4401–4410, 2019.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

Kingma, D. and Gao, R. Understanding diffusion objectives as the elbo with simple data augmentation. In *NeurIPS*, 2024.

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

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Kondratyuk, D., Yu, L., Gu, X., Lezama, J., Huang, J., Hornung, R., Adam, H., Akbari, H., Alon, Y., Birodkar, V., et al. VideoPoet: A large language model for zero-shot video generation. In *ICML*, 2024.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. Improved precision and recall metric for assessing generative models. In *NeurIPS*, 2019.

Lee, S., Kim, B., and Ye, J. C. Minimizing trajectory curvature of ODE-based generative models. In *ICML*, pp. 18957–18973, 2023.

Lee, S., Lin, Z., and Fanti, G. Improving the training of rectified flows. *arXiv preprint arXiv:2405.20320*, 2024.

Li, T., Tian, Y., Li, H., Deng, M., and He, K. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *ECCV*, pp. 740–755, 2014.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Luo, S., Tan, Y., Huang, L., Li, J., and Zhao, H. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Ma, N., Goldstein, M., Albergo, M. S., Boffi, N. M., Vanden-Eijnden, E., and Xie, S. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024.

Menon, S., Damian, A., Hu, S., Ravi, N., and Rudin, C. PULSE: Self-supervised photo upsampling via latent space exploration of generative models. In *CVPR*, pp. 2437–2445, 2020.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for GANs do actually converge? In *ICML*, pp. 3481–3490, 2018.

Nguyen, T. H. and Tran, A. SwiftBrush: One-step text-to-image diffusion model with variational score distillation. In *CVPR*, pp. 7807–7816, 2024.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *ICML*, pp. 8162–8171, 2021.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *ICCV*, pp. 4195–4205, 2023.

Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. FiLM: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

Pernias, P., Rampas, D., Richter, M. L., Pal, C. J., and Aubreville, M. Würstchen: An efficient architecture for large-scale text-to-image diffusion models. In *ICLR*, 2024.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024.

Preechakul, K., Chatthee, N., Wizadwongsa, S., and Suwajanakorn, S. Diffusion autoencoders: Toward a meaningful and decodable representation. In *CVPR*, pp. 10619–10629, 2022.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018.

Razavi, A., Van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, 2019.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10684–10695, 2022.

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

Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., and Norouzi, M. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH*, pp. 1–10, 2022a.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. Image super-resolution via iterative refinement. *IEEE TPAMI*, 45(4):4713–4726, 2022b.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training GANs. In *NeurIPS*, 2016.

Sargent, K., Hsu, K., Johnson, J., Fei-Fei, L., and Wu, J. Flow to the mode: Mode-seeking diffusion autoencoders for state-of-the-art image tokenization. *arXiv preprint arXiv:2503.11056*, 2025.

Sauer, A., Lorenz, D., Blattmann, A., and Rombach, R. Adversarial diffusion distillation. In *ECCV*, pp. 87–103, 2024.

Shannon, C. E. et al. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec*, 4(142-163): 1, 1959.

Shi, J., Wu, C., Liang, J., Liu, X., and Duan, N. DiVAE: Photorealistic images synthesis with denoising diffusion decoder. *arXiv preprint arXiv:2206.00386*, 2022.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Steed, R. and Caliskan, A. Image representations learned with unsupervised pre-training contain human-like biases. In *ACM conference on fairness, accountability, and transparency*, pp. 701–713, 2021.

Sun, P., Jiang, Y., Chen, S., Zhang, S., Peng, B., Luo, P., and Yuan, Z. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. Conditional image generation with pixelcnn decoders. In *NeurIPS*, 2016.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. In *NeurIPS*, 2017.

Wang, F.-Y., Huang, Z., Bergman, A. W., Shen, D., Gao, P., Lingelbach, M., Sun, K., Bian, W., Song, G., Liu, Y., et al. Phased consistency model. *arXiv preprint arXiv:2405.18407*, 2024a.

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

Wang, Z., Jiang, Y., Zheng, H., Wang, P., He, P., Wang, Z. A., Chen, W., and Zhou, M. Patch diffusion: Faster and more data-efficient training of diffusion models. In *NeurIPS*, 2024c.

Wu, Y. and He, K. Group normalization. In *ECCV*, pp. 3–19, 2018.

Xiao, Z., Kreis, K., and Vahdat, A. Tackling the generative learning trilemma with denoising diffusion GANs. In *ICLR*, 2022.

Yang, R. and Mandt, S. Lossy image compression with conditional diffusion models. In *NeurIPS*, 2024.

Yu, J., Li, X., Koh, J. Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., and Wu, Y. Vector-quantized image modeling with improved VQGAN. In *ICLR*, 2022.

Yu, L., Cheng, Y., Sohn, K., Lezama, J., Zhang, H., Chang, H., Hauptmann, A. G., Yang, M.-H., Hao, Y., Essa, I., et al. MAGVIT: Masked generative video transformer. In *CVPR*, pp. 10459–10469, 2023.

Yu, L., Lezama, J., Gundavarapu, N. B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Gupta, A., Gu, X., Hauptmann, A. G., et al. Language model beats diffusion–tokenizer is key to visual generation. In *ICLR*, 2024a.

Yu, Q., Weber, M., Deng, X., Shen, X., Cremers, D., and Chen, L.-C. An image is worth 32 tokens for reconstruction and generation. *arXiv preprint arXiv:2406.07550*, 2024b.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pp. 586–595, 2018.

Zhao, Y., Xiong, Y., and Krähenbühl, P. Image and video tokenization with binary spherical quantization. *arXiv preprint arXiv:2406.07548*, 2024.

Zhu, Z., Feng, X., Chen, D., Bao, J., Wang, L., Chen, Y., Yuan, L., and Hua, G. Designing a better asymmetric VQGAN for StableDiffusion. *arXiv preprint arXiv:2306.04632*, 2023.

A. Discussion

Distribution-aware compression. Traditional image compression methods optimize the rate-distortion trade-off (Shannon et al., 1959), prioritizing compactness over input fidelity. Building on this, we also aim to capture the broader input distribution during compression, generating compact representations suitable for latent generative models. This approach introduces an additional dimension to the trade-off, perception or distribution fidelity (Blau & Michaeli, 2018), which aligns more closely with the rate-distortion-perception framework (Blau & Michaeli, 2019).

Iterative and stochastic decoding. A key question within the rate-distortion-perception trade-off is whether the iterative, stochastic nature of diffusion decoding offers advantages over traditional single-step, deterministic methods (Kingma, 2013). The strengths of diffusion (Ho et al., 2020) lie in its iterative process, which progressively refines the latent space for more accurate reconstructions, while stochasticity allows for capturing complex variations within the distribution. Although iterative methods may appear less efficient, our formulation is optimized to achieve optimal results in just three steps and also supports single-step decoding, ensuring decoding efficiency remains practical (see Fig. 3 (left)). While stochasticity might suggest the risk of "hallucination" in reconstructions, the outputs remain faithful to the underlying distribution by design, producing perceptually plausible results. This advantage is particularly evident under extreme compression scenarios (see Fig. 5), with the degree of stochasticity adapting based on compression levels (see Fig. 6).

Multi-step *vs.* single-step decoding. While replacing single-step decoding with an iterative process may seem counter-intuitive due to increased computational cost, the diffusion-based decoder addresses this concern in three key ways. First, it offers scalable inference, where even a single-step variant already outperforms a plain VAE decoder, and additional steps further enhance quality (see Tab. 3). Second, it provides controllable trade-offs between computation and visual fidelity, allowing the number of steps to be adjusted at inference time based on application needs. Third, as shown in Tab. 4, it enables training under higher compression ratios, which helps offset the added cost of iterative decoding by reducing the size of latent representations.

Scalability. As discussed in Sec. 4.1, our diffusion-based decoding method maintains the resolution generalizability typically found in standard autoencoders. This feature is highly practical: the autoencoder is trained on lower-resolution images, while the subsequent latent generative model is trained on latents derived from higher-resolution inputs. However, we acknowledge that memory overhead and throughput become concerns with our UNet-based diffusion decoder, especially for high-resolution inputs. This challenge becomes more pronounced as models, datasets, or resolutions scale up. A promising future direction is patch-based diffusion (Ding et al., 2024; Wang et al., 2024c), which partitions the input into smaller, independently processed patches. This approach has the potential to reduce memory usage and enable faster parallel decoding.

B. Related work

Image tokenization. Image tokenization is crucial for effective generative modeling, transforming images into compact, structured representations. A common approach employs an autoencoder framework (Hinton & Salakhutdinov, 2006), where the encoder compresses images into low-dimensional latent representations, and the decoder reconstructs the original input. These latent representations can be either discrete commonly used in autoregressive models (Van den Oord et al., 2016; Van Den Oord et al., 2017; Chen et al., 2020; Chang et al., 2022; Yu et al., 2023; Kondratyuk et al., 2024), or continuous, as found in diffusion models (Ho et al., 2020; Dhariwal & Nichol, 2021; Rombach et al., 2022; Peebles & Xie, 2023; Gupta et al., 2023; Brooks et al., 2024). The foundational form of visual autoencoding today originates from Van Den Oord et al. (2017). While advancements have been made in modeling (Yu et al., 2022; 2024b; Chen et al., 2024), objectives (Zhang et al., 2018; Karras et al., 2019; Esser et al., 2021), and quantization methods (Yu et al., 2024a; Zhao et al., 2024), the core encoding-and-decoding scheme remains largely the same.

In this work, we propose a new perspective by replacing the traditional decoder with a diffusion process. Specifically, our new formulation retains the encoder but introduces a conditional diffusion decoder. Within this framework, we systematically study various design choices, resulting in a significantly enhanced autoencoding setup.

Additionally, we refer to the recent work MAR (Li et al., 2024), which leverages diffusion to model per-token distribution in autoregressive frameworks. In contrast, our approach models the overall input distribution in autoencoders using diffusion. This difference leads to distinct applications of diffusion during generation. For instance, MAR generates samples autoregressively, decoding each token iteratively using diffusion, token by token. In our method, we first sample all tokens from the downstream generative model and then decode them iteratively using diffusion as a whole.

Image compression. Our work shares similarities with recent image compression approaches that leverage diffusion models. For example, Hoogeboom et al. (2023a); Birodkar et al. (2024) use diffusion to refine autoencoder residuals, enhancing high-frequency details. Yang & Mandt (2024) employs a diffusion decoder conditioned on quantized dis-

Figure 5. **Reconstruction results with varying downsampling ratios.** ϵ-VAE maintains both high fidelity and perceptual quality, even under extreme downsampling conditions, whereas VAE fails to preserve semantic integrity. *Best viewed when zoomed-in and in color.*

crete codes and omits the GAN loss. However, these methods primarily focus on the traditional rate-distortion tradeoff, balancing rate (compactness) and distortion (input fidelity) (Shannon et al., 1959), with the goal of storing and transmitting data efficiently without significant loss of information.

In this work, we emphasize perception (distribution fidelity) alongside the rate-distortion tradeoff, ensuring that reconstructions more closely align with the overall data distribution (Heusel et al., 2017; Zhang et al., 2018; Blau & Michaeli, 2019), thereby enhancing the decoded results from the sampled latents of downstream generative models. We achieve this by directly integrating the diffusion process into the decoder, unlike Hoogeboom et al. (2023a); Birodkar et al. (2024). Moreover, unlike Yang & Mandt (2024), we do not impose strict rate-distortion regularization in the latent space and allow the GAN loss to synergize with our approach.

Diffusion decoder. Several studies (Preechakul et al., 2022; Shi et al., 2022; Pernias et al., 2024; Nguyen & Tran, 2024; Sauer et al., 2024; Luo et al., 2023) have explored diffusion decoders conditioned on compressed latents of the input, which are relevant to our work. We outline the key differences between these works and ϵ-VAE: First, prior works have not fully leveraged the synergy between diffusion decoders and standard VAE training objectives. In this work, we enhance state-of-the-art VAE objectives by replacing the reconstruction loss with a score matching loss and adapting LPIPS and GAN losses to ensure compatibility with diffusion decoders. These changes yield significant improvements in autoencoding performance, as evidenced by lower rFID scores and faster inference. Second, we are the first to investigate various parameterizations (*e.g.*, epsilon and velocity) and demonstrate that modern velocity parameterization, coupled with optimized train and test-time

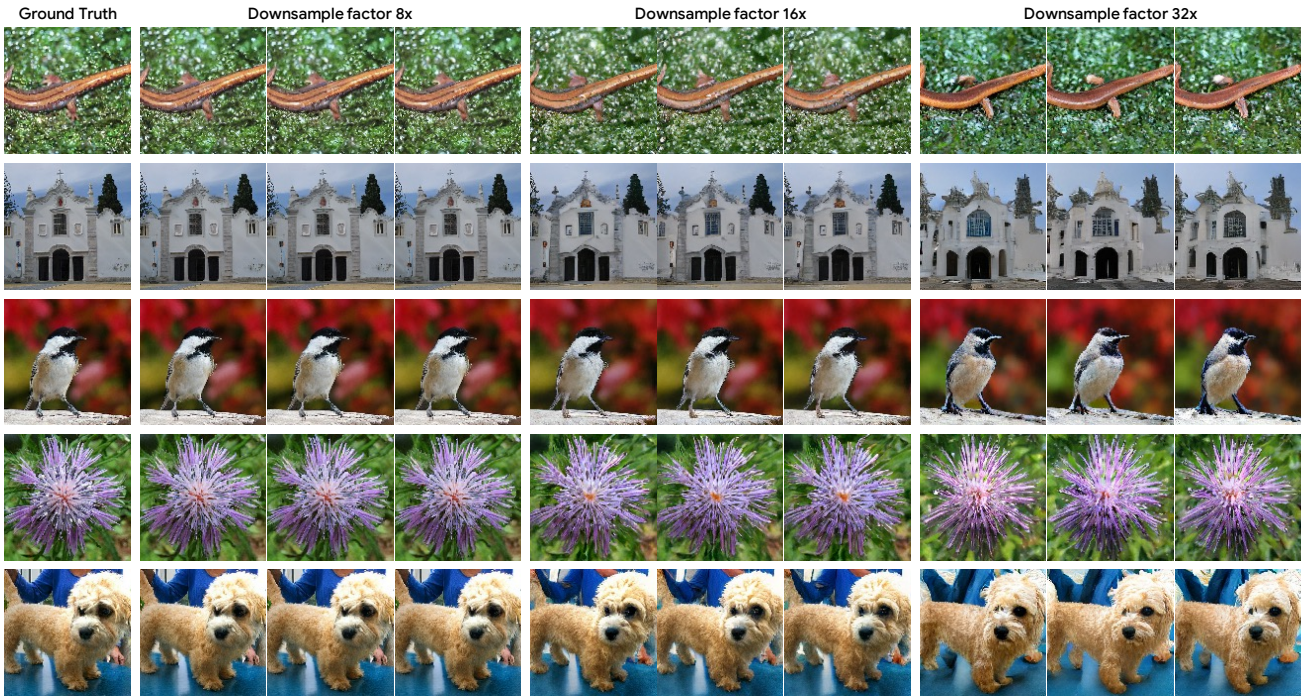


Figure 6. **ϵ-VAE reconstruction results with varying random seeds and downsampling ratios.** At lower compression levels, the reconstruction behaves more deterministically, whereas higher compression introduces stochasticity, enabling more flexible reconstruction of plausible inputs. *Best viewed when zoomed-in and in color*.

noise scheduling, provides substantial benefits. These enhancements improve both reconstruction performance and sampling efficiency. Third, previous diffusion-based decoders (Preechakul et al., 2022; Shi et al., 2022; Pernias et al., 2024), which often rely on ad-hoc techniques like distillation or consistency regularization to speed up inference (Nguyen & Tran, 2024; Sauer et al., 2024; Luo et al., 2023), our approach achieves fast decoding (1 to 3 steps) without such techniques. This is made possible by integrating our proposed objectives and parameterizations. Last but not least, ϵ-VAE exhibits strong resolution generalization capabilities, a key property of standard VAEs. In contrast, models like DiffusionAE (Preechakul et al., 2022) and DiVAE (Shi et al., 2022) either lack this ability or are inherently limited. For example, DiVAE's bottleneck add/concat design restricts its capacity to generalize across resolutions.

SWYCC (Birodkar et al., 2024) also explores joint learning of continuous encoders and decoders using a diffusion model. However, SWYCC differs fundamentally from our approach: it replaces the GAN loss with a diffusion-based loss, while we focus on identifying optimal synergies between traditional autoencoding losses (including GAN loss) and diffusion-based decoding. Our goal is to identify an optimal strategy for combining these elements, rather than simply substituting one for another.

Another closely related work, DiTo (Chen et al., 2025), also presents a diffusion-based tokenizer which learns compact visual representations for image generation. Its main insight is that a single diffusion learning objective is capable of training scalable image tokenizers. More than that, our method demonstrates that traditional autoencoding losses such as LPIPS and GAN losses are complimentary to the diffusion target, leading to better reconstruction quality. This design substantially differ our work from DiTo.

While following a different motivation, Lee et al. (2023) essentially also proposes a VAE with a denoising decoder but uses the encoding as the "initial noise" instead of as conditioning for a standard diffusion model starting from a standard Gaussian distribution. This idea could be potentially used for speeding up the proposed approach, which we will explore in the future.

Image generation. Recent advances in image generation span a wide range of approaches, including VAEs (Kingma, 2013), GANs (Goodfellow et al., 2014), autoregressive models (Chen et al., 2020) and diffusion models (Song et al., 2021; Ho et al., 2020). Among these, diffusion models have emerged as the leading approach for generating high-dimensional data such as images (Saharia et al., 2022a; Baldridge et al., 2024; Esser et al., 2024) and videos (Brooks et al., 2024; Gupta et al., 2023), where the gradual refinement of global structure is crucial. The current focus in diffusion-based generative models lies in advancing archi-

Table 6. **Hyper-parameters for decoder variants.**

Model	Channel dim.	Depth multipliers	# of blocks
Base (B)	64	{1, 1, 2, 2, 4}	2
Medium (M)	96	{1, 1, 2, 2, 4}	2
Large (L)	128	{1, 1, 2, 2, 4}	2
Extra-large (XL)	128	{1, 1, 2, 2, 4}	4
Huge (H)	256	{1, 1, 2, 2, 4}	2

Table 7. **Image reconstruction results on ImageNet** 128×128**.**

Configuration	NFE↓	rFID↓
Baseline (c) in Tab. 5:		
Inject conditioning by channel-wise concatenation	50	22.04
Inject conditioning by AdaGN	50	22.01
Baseline (e) in Tab. 5:		
Matching the distribution of $\hat{\boldsymbol{x}}_0^t$ and $\boldsymbol{x}_0$	-	N/A
Matching the trajectory of $\boldsymbol{x}_t \to \boldsymbol{x}_0$	5	8.24
Matching the trajectory of $\boldsymbol{x}_t \to \boldsymbol{x}_{t-\Delta t}$	5	10.53

tectures (Rombach et al., 2022; Peebles & Xie, 2023; Hoogeboom et al., 2023b), parameterizations (Karras et al., 2022; Kingma & Gao, 2024; Ma et al., 2024; Esser et al., 2024), or better training dynamics (Nichol & Dhariwal, 2021; Chen, 2023; Chen et al., 2023). However, tokenization, an essential component in modern diffusion models, often receives less attention.

In this work, we focus on providing compact continuous latents without applying quantization during autoencoder training (Rombach et al., 2022), as they have been shown to be effective in state-of-the-art latent diffusion models (Rombach et al., 2022; Saharia et al., 2022a; Peebles & Xie, 2023; Esser et al., 2024; Baldridge et al., 2024). We compare our autoencoding performance against the baseline approach (Esser et al., 2021) using the DiT framework (Peebles & Xie, 2023) as the downstream generative model.

C. Experiment setups

In this section, we provide additional details on our experiment configurations for reproducibility.

C.1. Model specifications

Tab. 6 summarizes the primary architecture details for each decoder variant. The channel dimension is the number of channels of the first U-Net layer, while the depth multipliers are the multipliers for subsequent resolutions. The number of residual blocks denotes the number of residual stacks contained in each resolution.

C.2. Implementation details

During the training of discriminators, Esser et al. (2021) introduced an adaptive weighting strategy for λ_{adv}. However, we notice that this adaptive weighting does not introduce any benefit which is consistent with the observation made by Sadat et al. (2024). Thus, we set $\lambda_{\text{adv}} = 0.5$ in the experiments for more stable model training across different configurations.

The autoencoder loss follows Eq. 1, with weights set to $\lambda_{\text{LPIPS}} = 0.5$ and $\lambda_{\text{adv}} = 0.5$. We use the Adam optimizer (Kingma & Ba, 2015) with $\beta_1 = 0$ and $\beta_2 = 0.999$, applying a linear learning rate warmup over the first 5,000 steps, followed by a constant rate of 0.0001 for a total of one million steps. The batch size is 256, with data augmentations including random cropping and horizontal flipping. An exponential moving average of model weights is maintained with a decay rate of 0.999. All models are implemented in JAX/Flax (Bradbury et al., 2018; Heek et al., 2024) and trained on TPU-v5lite pods.

C.3. Latent diffusion models

We follow the setting in Peebles & Xie (2023) to train the latent diffusion models for unconditional image generation on the ImageNet dataset. The DiT-XL/2 architecture is used for all experiments. The diffusion hyperparameters from ADM (Dhariwal & Nichol, 2021) are kept. To be specific, we use a $t_{\text{max}} = 1000$ linear variance schedule ranging from 0.0001 to 0.02, and results are generated using 250 DDPM sampling steps. For simplicity and training stability, we remove the variational lower bound loss term during training, which leads to a slight drop in generation qualities.

All models are trained with Adam (Kingma & Ba, 2015) with no weight decay. We use a constant learning rate of 0.0001 and a batch size of 256. Horizontal flipping and random cropping are used for data augmentation. We maintain an exponential moving average of DiT weights over training with a decay of 0.9999. We use identical training hyperparameters across all experiments and train models for one million steps in total. No classifier-free guidance (Ho & Salimans, 2022) is employed in all the experiments. Inference throughputs are computed on a Tesla H100 GPU.

D. Additional experimental results

We note that all experiments conducted in this section are under the ϵ-VAE-lite configuration.

Conditioning. In addition to injecting conditioning via channel-wise concatenation, we explore providing conditioning to the diffusion model by adaptive group normalization (AdaGN) (Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021). To achieve this, we resize the conditioning (*i.e.*, encoded latents) via bilinear sampling to the desired

Table 8. **Model scaling and resolution generalization analysis.** Five model variants are trained and evaluated. Δ_{rFID} represents the absolute differences (or relative ratio) in rFID between the corresponding model size variants of VAE and ϵ-VAE. † denotes resolution generalization experiments. To fairly evaluate the impact of ϵ-VAE under controlled model parameters, we highlight three groups of model variants with comparable parameters, using different colors.

Model	$\mathcal{G}$ params (M)	ImageNet 128×128		ImageNet 256×256 †		ImageNet 512×512 †	
		rFID↓	Δ_{rFID}	rFID↓	Δ_{rFID}	rFID↓	Δ_{rFID}
VAE (B)	10.14	11.15	-	5.74	-	3.69	-
VAE (M)	22.79	9.26	-	4.63	-	2.69	-
VAE (L)	40.48	8.49	-	4.78	-	2.78	-
VAE (XL)	65.27	7.58	-	4.42	-	2.41	-
VAE (H)	161.81	7.12	-	4.29	-	2.37	-
ϵ-VAE (B)	20.63	6.24	4.91 (44.0%)	3.90	1.84 (32.0%)	2.06	1.63 (44.2%)
ϵ-VAE (M)	49.33	5.42	3.84 (41.5%)	2.79	1.84 (39.7%)	2.02	0.67 (24.9%)
ϵ-VAE (L)	88.98	4.71	3.78 (44.5%)	2.60	2.03 (43.8%)	1.92	0.86 (30.9%)
ϵ-VAE (XL)	140.63	4.18	3.40 (44.9%)	2.38	2.04 (46.2%)	1.82	0.59 (24.5%)
ϵ-VAE (H)	355.62	4.04	3.08 (43.3%)	2.31	1.98 (46.2%)	1.78	0.59 (24.9%)

Table 9. **Unconditional image generation quality.** The DiT-XL/2 is trained on latents provided by the trained autoencoders, VAE and ϵ-VAE, with varying model sizes using ImageNet. We evaluate the generation quality at resolutions of 128×128 and 256×256 using four standard metrics. Additionally, we report rFID to determine if the improvement trend observed in reconstruction task extends to the generation task. We highlight three groups of model variants with comparable parameters.

Model	ImageNet 128×128					ImageNet 256×256				
	rFID↓	FID↓	IS↑	Prec.↑	Rec.↑	rFID↓	FID↓	IS↑	Prec.↑	Rec.↑
VAE (B)	11.15	36.8	17.9	0.48	0.53	5.74	46.6	23.4	0.45	0.56
VAE (M)	9.26	34.6	18.2	0.49	0.55	4.63	44.7	23.8	0.47	0.58
VAE (L)	8.49	33.9	18.4	0.50	0.56	4.78	44.3	24.7	0.47	0.59
VAE (XL)	7.58	31.7	19.3	0.51	0.57	4.42	43.1	24.9	0.47	0.59
VAE (H)	7.12	30.9	19.8	0.52	0.57	4.29	41.6	25.9	0.48	0.59
ϵ-VAE (B)	6.24	29.5	20.7	0.53	0.59	3.90	39.5	25.2	0.46	0.61
ϵ-VAE (M)	5.42	27.6	21.2	0.55	0.59	2.79	35.4	26.2	0.51	0.62
ϵ-VAE (L)	4.71	27.3	22.1	0.55	0.59	2.60	34.8	26.5	0.51	0.63
ϵ-VAE (XL)	4.18	25.3	22.7	0.55	0.59	2.38	34.0	27.4	0.53	0.63
ϵ-VAE (H)	4.04	24.9	23.0	0.56	0.60	2.31	33.2	27.5	0.54	0.64

resolution of each stage in the U-Net model, and incorporates it into each residual block after a group normalization operation (Wu & He, 2018). This is similar to adaptive instance norm (Karras et al., 2019) and FiLM (Perez et al., 2018). We report the results in Tab. 7 (top), where we find that channel-wise concatenation and AdaGN obtain similar reconstruction quality in terms of rFID. Because of the additional computational cost required by AdaGN, we thus apply channel-wise concatenation in our model by default.

Trajectory matching. The proposed denoising trajectory matching objective matches the start-to-end trajectory $\boldsymbol{x}_t \rightarrow \boldsymbol{x}_0$ by default. One alternative choice is to directly matching the distribution of $\hat{\boldsymbol{x}}_0^t$ and $\boldsymbol{x}_0$ without coupling on $\boldsymbol{x}_t$. However, we find this formulation leads to unstable training and could not produce reasonable results. Here, we present the results when matching the trajectory of $\boldsymbol{x}_t \rightarrow \boldsymbol{x}_{t-\Delta t}$, which is commonly used in previous work (Xiao et al., 2022; Wang et al., 2024a). Specifically, for each timestep t during training, we randomly sample a step Δt from $(0, t)$. Then, we construct the real trajectory by computing $\boldsymbol{x}_{t-\Delta t}$ via Eq. 5 and concatenating it with $\boldsymbol{x}_t$, while the fake trajectory is obtained in a similar way but using Eq. 9 instead. Tab. 7 (bottom) shows the comparison. We observe that matching trajectory $\boldsymbol{x}_t \rightarrow \boldsymbol{x}_0$ yields better performance than matching trajectory $\boldsymbol{x}_t \rightarrow \boldsymbol{x}_{t-\Delta t}$, confirming the effectiveness of the proposed objective which is designed for the rectified flow formulation.

Comparisons with plain diffusion ADM. Under the same training setup of Tab. 5, we directly trained a plain diffusion model (ADM) for comparison, which resulted in rFID score of 38.26. Its conditional form is already provided as a baseline in Tab. 5, achieving 28.22. This demonstrates that our conditional form $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{z})$ offers a better approximation of the true posterior $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ compared to the standard form $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$. By further combining LPIPS and GAN loss, we achieve rFID of 8.24, outperforming its VAE counterpart, which achieves 11.15. With better training configurations, our final rFID improves to 6.24. This pro-

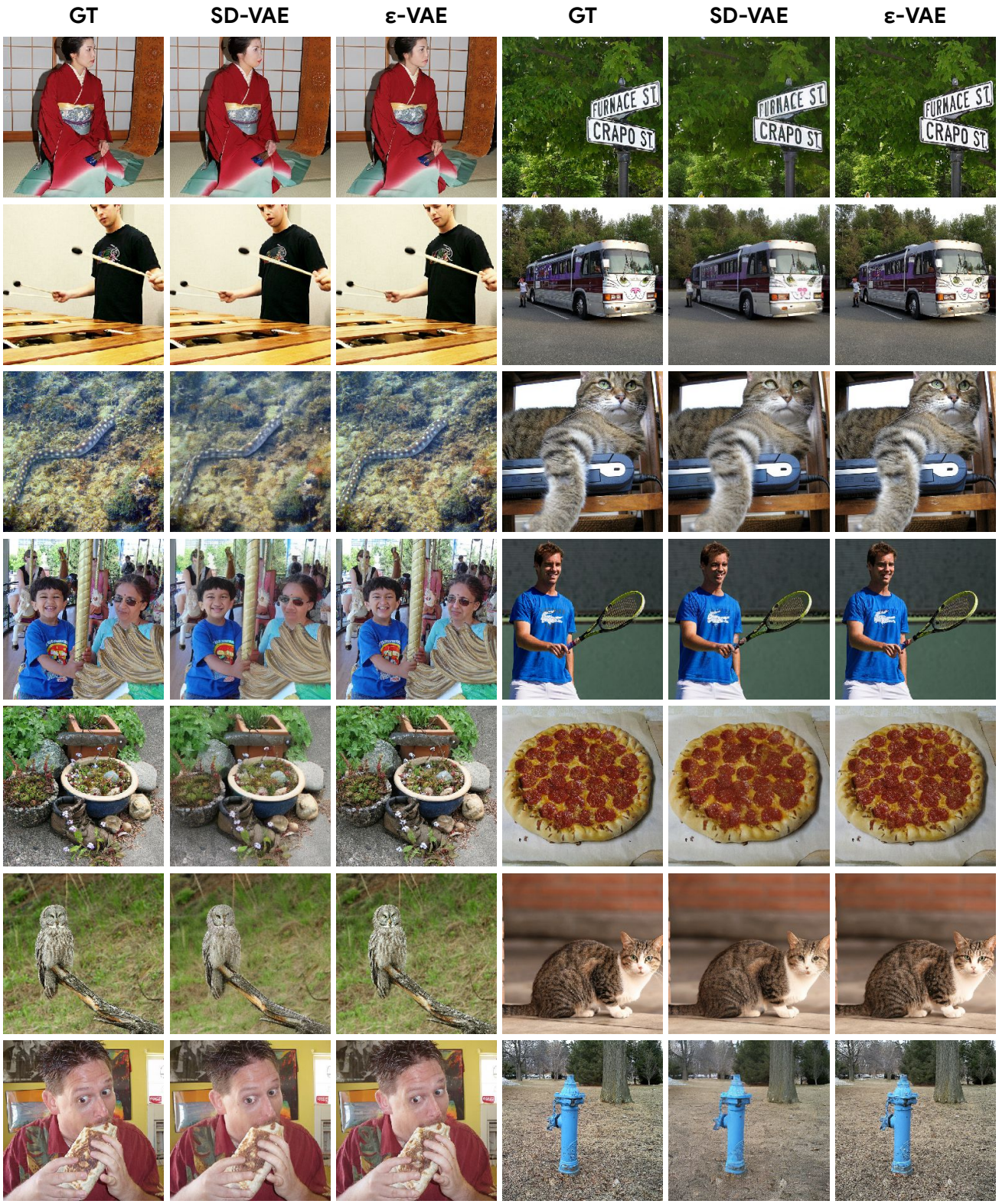


Figure 7. **Image reconstruction results under the SD-VAE configuration (Rombach et al., 2022) at the resolution of** 256×256**.** ϵ-VAE produces significantly better visual details than SD-VAE when reconstructing local regions with complex textures or structures, such as human faces and small texts. *Best viewed when zoomed-in and in color*.

Table 10. **Benchmarking class-conditional image generation on ImageNet** 256×256**.** We use the DiT-XL/2 architecture (Esser et al., 2024) for latent diffusion models and apply classifier-free guidance (Ho & Salimans, 2022).

Downsample factor	Method	FID↓
8×8	SD-VAE (Rombach et al., 2022)	3.51
	ϵ-VAE-SD (M)	<u>2.83</u>
	ϵ-VAE-SD (H)	**2.69**

gression, from plain diffusion ADM to ϵ-VAE, underscores the significance of our proposals and their impact.

Model scaling. We investigate the impact of model scaling by comparing VAE and ϵ-VAE across five model variants, all trained and evaluated at a resolution of 128×128, as summarized in Tab. 8. The results demonstrate that ϵ-VAE consistently achieves significantly better rFID scores than VAE, with an average relative improvement of over 40%, and even the smallest ϵ-VAE model outperforms VAE at largest scale. While the U-Net-based decoder of ϵ-VAE has about twice as many parameters as standard decoder of VAE, grouping models by similar sizes, highlighted in different colors, shows that performance gains are not simply due to increased model parameters.

Tab. 9 presents the unconditional image generation results of VAE and ϵ-VAE at resolutions of 128×128 and 256×256. In addition to FID, we report Inception Score (IS) (Salimans et al., 2016) and Precision/Recall (Kynkäänniemi et al., 2019) as secondary metrics. The results show that ϵ-VAE consistently outperforms VAE across all model scales. Notably, ϵ-VAE (B) surpasses VAE (H), consistent with our earlier findings in Sec. 4.1. These results further demonstrate the effectiveness of ϵ-VAE from the generation perspective.

Results with classifier-free guidance. We provide additional results with classifier-free guidance (Ho & Salimans, 2022) under the 8×8 downsample factor in Tab. 10. We find that ϵ-VAE (M) performs relatively 20% better than SD-VAE and further improvements are obtained after we scale up our model to ϵ-VAE (H). These results are consistent with the results without classifier-free guidance in Tab. 4, confirming the effectiveness of our model.

E. Additional visual results

We provide additional visual comparisons between ϵ-VAE and SD-VAE at the resolution of 256×256 (Fig. 7). Our observations indicate that ϵ-VAE delivers significantly better visual quality than SD-VAE, particularly when reconstructing local regions with complex textures or structures, such as human faces and small text.