

# BREAKING THE LIKELIHOOD–QUALITY TRADE-OFF IN DIFFUSION MODELS BY MERGING PRETRAINED EXPERTS

Yasin Esfandiari<sup>1\*</sup> Stefan Bauer<sup>2,3</sup> Sebastian U. Stich<sup>4</sup> Andrea Dittadi<sup>2,3,5</sup>

<sup>1</sup>Saarland University <sup>2</sup>Helmholtz AI <sup>3</sup>Technical University of Munich

<sup>4</sup>CISPA Helmholtz Center for Information Security <sup>5</sup>MPI for Intelligent Systems, Tübingen

## ABSTRACT

Diffusion models for image generation often exhibit a trade-off between perceptual sample quality and data likelihood: training objectives emphasizing high-noise denoising steps yield realistic images but poor likelihoods, whereas likelihood-oriented training overweights low-noise steps and harms visual fidelity. We introduce a simple plug-and-play sampling method that combines two pre-trained diffusion experts by switching between them along the denoising trajectory. Specifically, we apply an image-quality expert at high noise levels to shape global structure, then switch to a likelihood expert at low noise levels to refine pixel statistics. The approach requires no retraining or fine-tuning—only the choice of an intermediate switching step. On CIFAR-10 and ImageNet32, the merged model consistently matches or outperforms its base components, improving or preserving both likelihood and sample quality relative to each expert alone. These results demonstrate that expert switching across noise levels is an effective way to break the likelihood–quality trade-off in image diffusion models.

## 1 INTRODUCTION

Diffusion models are a class of probabilistic generative models that learn to approximate a data distribution by reversing a forward noising process through a learned denoising procedure (Sohl-Dickstein et al., 2015; Ho et al., 2020; Nichol & Dhariwal, 2021). They have recently achieved state-of-the-art results, e.g., in image generation (Dhariwal & Nichol, 2021; Tang et al., 2024; Kim et al., 2024), density estimation (Kingma et al., 2021), and in text-to-image and text-to-video generation tasks (Esser et al., 2024; Polyak et al., 2024).

For image data, likelihood and perceptual quality are often misaligned in practice (Theis et al., 2015), that is, strong performance on one does not necessarily imply good performance on the other. Notably, Kim et al. (2021) report an inverse correlation between likelihood and sample quality as measured via the Fréchet Inception Distance (FID). As a result, models that aim to maximize likelihood typically optimize a lower bound on it, whereas models prioritizing perceptual quality modify the training objective, e.g., by reweighting contributions from different time steps in the diffusion process. This trade-off implies that models producing visually appealing samples often achieve lower likelihoods, while those optimized for likelihood tend to generate less realistic images. Because likelihood and FID capture complementary aspects of generative modeling—the statistical fidelity of the data versus its perceptual realism—balancing both is crucial for developing diffusion models that accurately represent the data distribution while producing convincing visual samples.

In this paper, we aim to overcome the likelihood–FID trade-off by designing a model that can generate images with both high perceptual quality and strong likelihood. To do this, we start from two key empirical observations reported in the literature: (1) Higher noise levels are associated with perceptual image quality. For example, DDPM (Ho et al., 2020) employs a simplified objective that down-weights the loss at lower noise levels, allowing the model to focus on the more challenging denoising steps at higher noise levels. Similarly, Kim et al. (2021) showed that accurate score

\*Work done while at Helmholtz AI. Correspondence to [yaes00001@stud.uni-saarland.de](mailto:yaes00001@stud.uni-saarland.de).

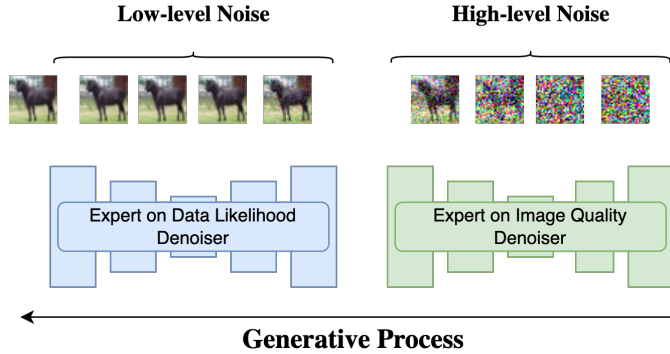


Figure 1: Diagram of our merged model where at an intermediate time  $\eta \in [0, 1]$  we switch between denoisers. Note that the likelihood model is only used for almost imperceptible noise levels. This significantly improves the likelihood, which is sensitive to low-level color statistics, while leaving the FID unaffected.

prediction at high noise levels is crucial for generating realistic samples. (2) Likelihood is highly sensitive to low-level pixel statistics (Zheng et al., 2023b; Kim et al., 2021), whereas perceptual quality is primarily determined by global image structure rather than fine-grained pixel details. Supporting this view, Kingma & Dhariwal (2018) and Kingma & Gao (2024) showed that training on 5-bit images, which effectively discards fine details, can lead to better visual quality.

Motivated by these insights, we propose a simple approach that merges two pretrained diffusion experts—one specialized in image quality and the other in likelihood. Specifically, we use EDM (Karras et al., 2022) as the image-quality expert for high noise levels, and VDM (Kingma et al., 2021) as the likelihood expert for low noise levels. An overview of the merged model is provided in Fig. 1. Starting from noise, the model first denoises up to a chosen intermediate step using the image-quality expert, producing a high-fidelity yet slightly noisy sample. The process then switches to the likelihood expert, which refines the sample to improve likelihood while preserving perceptual quality. By appropriately selecting the switching point along the denoising trajectory, the model achieves strong performance in both FID and likelihood, effectively overcoming the trade-off between the two.

The remainder of this paper is organized as follows. Section 2 introduces the necessary preliminaries on diffusion models. Section 3 presents our framework for adapting pretrained models for reuse across different processes, enabling the merging of multiple experts. Section 4 describes the experimental setup and reports quantitative and qualitative results on CIFAR-10 and ImageNet32. Finally, Section 5 discusses related work, and Section 6 concludes with limitations and directions for future research.

## 2 PRELIMINARIES

Diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020) are a class of generative models that learn to reverse a diffusion process that gradually perturbs data with noise. Let  $\mathbf{x} \in \mathbb{R}^d$  denote a data point drawn from an unknown distribution  $q_{\text{data}}$ . The forward process is a continuous-time stochastic process  $(\mathbf{z}_t)_{t \in [0, 1]}$  in  $\mathbb{R}^d$  initialized from a simple conditional distribution  $q(\mathbf{z}_0 | \mathbf{x}) = \mathcal{N}(\mathbf{z}_0; \alpha_0 \mathbf{x}, \sigma_0^2 \mathbf{I})$  with scalar parameters  $\alpha_0, \sigma_0$ . A common choice for the forward dynamics is an Ornstein–Uhlenbeck SDE with a time-dependent but data-independent drift:

$$d\mathbf{z}_t = f_t \mathbf{z}_t dt + g_t d\mathbf{w}_t, \quad (1)$$

where  $\mathbf{w}_t$  is a standard Wiener process and  $f_t, g_t$  are scalar functions of time. Under this construction, the marginal density of  $\mathbf{z}_t$  conditional on data is

$$q(\mathbf{z}_t | \mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}), \quad (2)$$

where  $\alpha_t, \sigma_t \in \mathbb{R}_{>0}$  are smooth scalar-valued functions of  $t$  defining the *noise schedule*. Choosing  $(\alpha_t, \sigma_t)$  determines the SDE coefficients via:

$$f_t = \frac{d \log \alpha_t}{dt}, \quad g_t^2 = \alpha_t^2 \frac{d}{dt} \left[ \frac{\sigma_t^2}{\alpha_t^2} \right]. \quad (3)$$

We assume that the *signal-to-noise ratio* (SNR),  $\alpha_t^2/\sigma_t^2$ , is monotonically decreasing in  $t$ , which makes the diffusion coefficient  $g_t$  from Eq. (3) well defined.

If we could sample from  $q(\mathbf{z}_1)$  and the marginal scores  $\nabla_{\mathbf{z}_t} \log q(\mathbf{z}_t)$  were known, we could obtain samples from  $q(\mathbf{z}_0)$  by simulating a deterministic or stochastic process backward in time. In the stochastic settings, several reverse-time SDEs are possible, with different diffusion coefficients  $g_t$ . A common choice uses the same  $g_t$  as in the noising process (Song et al., 2020b):

$$d\mathbf{z}_t = [f_t \mathbf{z}_t - g_t^2 \nabla_{\mathbf{z}_t} \log q(\mathbf{z}_t)] dt + g_t d\tilde{\mathbf{w}}_t, \quad \mathbf{z}_1 \sim q(\mathbf{z}_1), \quad (4)$$

where  $\tilde{\mathbf{w}}_t$  denotes a reverse-time Wiener process. Another common alternative is to define a deterministic process known as the *probability flow ODE* (PF ODE) (Song et al., 2020b):

$$\frac{d\mathbf{z}_t}{dt} = f_t \mathbf{z}_t - \frac{1}{2} g_t^2 \nabla_{\mathbf{z}_t} \log q(\mathbf{z}_t), \quad \mathbf{z}_1 \sim q(\mathbf{z}_1). \quad (5)$$

Both Eqs. (4) and (5) share the same time-marginals  $q(\mathbf{z}_t)$  as the forward process and therefore yield exact samples from  $q(\mathbf{z}_0)$ . Generating data then requires a decoder  $q(\mathbf{x}|\mathbf{z}_0) \propto q(\mathbf{z}_0|\mathbf{x})q_{\text{data}}(\mathbf{x})$  which is typically unavailable.

In practice, we replace the intractable ingredients by: (i) a *prior*  $p(\mathbf{z}_1) \approx \int q(\mathbf{z}_1|\mathbf{x})q_{\text{data}}(\mathbf{x})d\mathbf{x}$ , usually  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ ; (ii) a *score estimator*  $\mathbf{s}_\theta(\mathbf{z}_t, t) \approx \nabla \log q(\mathbf{z}_t)$ ; (iii) a *likelihood function*  $p(\mathbf{x}|\mathbf{z}_0) \approx q(\mathbf{x}|\mathbf{z}_0)$ , e.g., chosen to be proportional to  $q(\mathbf{z}_0|\mathbf{x})$  (Kingma et al., 2021). By using these three approximations to define the generative model, sampling a new data point  $\mathbf{x}$  proceeds by (i) drawing  $\mathbf{z}_1 \sim p(\mathbf{z}_1)$ , (ii) integrating Eq. (4) or Eq. (5) with the approximate score  $\mathbf{s}_\theta$  to obtain  $\mathbf{z}_0$ , and (iii) sampling  $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z}_0)$ .

The generative model can be learned by minimizing an upper bound on the negative log-likelihood:

$$-\log p_\theta(\mathbf{x}) \leq \underbrace{D_{\text{KL}}(q(\mathbf{z}_1|\mathbf{x})||p(\mathbf{z}_1))}_{\text{Prior loss}} + \underbrace{\mathbb{E}_{q(\mathbf{z}_0|\mathbf{x})}[-\log p(\mathbf{x}|\mathbf{z}_0)]}_{\text{Reconstruction loss}} + \underbrace{\mathcal{L}_{\text{diff}}(\mathbf{x}; \theta)}_{\text{Diffusion loss}} \quad (6)$$

$$\mathcal{L}_{\text{diff}}(\mathbf{x}; \theta) := \frac{1}{2} \mathbb{E}_{t \sim \mathcal{U}(0,1), \mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{x})} \left[ g_t^2 \|\mathbf{s}_\theta(\mathbf{z}_t, t) - \nabla \log q(\mathbf{z}_t|\mathbf{x})\|^2 \right]. \quad (7)$$

In this setting, the noising process, the prior distribution, and the likelihood  $p(\mathbf{x}|\mathbf{z}_0)$  are fixed. Learning the generative model then amounts to learning the score estimator by minimizing  $\mathcal{L}_{\text{diff}}$  over the training data. The minimizer of this objective corresponds to the *marginal score*  $\nabla \log q(\mathbf{z}_t)$  (Vintcent, 2011). This result provides a theoretical justification for the score-based approach: although the model is trained via a tractable regression objective against the conditional score, the procedure yields an accurate estimate of the desired marginal score. Consequently, the training objective not only aligns with maximum likelihood principles but also remains computationally efficient and theoretically well grounded. Moreover, this objective can be equivalently reformulated through various parameterizations of the score, such as predicting the original data, the added noise, the velocity (Salimans & Ho, 2022), or the PF ODE vector field (Lipman et al., 2022; Liu et al., 2022b)—see, e.g., Kingma & Gao (2024) for an overview.

Besides enabling deterministic sampling from diffusion models, the PF ODE (5) allows us to compute a tighter bound than Eq. (6). The exact likelihood on any  $\mathbf{z}_0$  can be computed via the instantaneous change of variables formula (Chen et al., 2018):

$$\log p_\theta(\mathbf{z}_0) = \log p(\mathbf{z}_1) + \int_0^1 \nabla \cdot \mathbf{h}_\theta(\mathbf{z}_t, t) dt \quad (8)$$

where  $\mathbf{h}_\theta(\mathbf{z}_t, t)$  is the vector field of the PF ODE (5) with  $\nabla \log q(\mathbf{z}_t)$  replaced by  $\mathbf{s}_\theta(\mathbf{z}_t, t)$ . The data log-likelihood can then be bounded as follows:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}_0|\mathbf{x})} [\log p(\mathbf{x}|\mathbf{z}_0) + \log p_\theta(\mathbf{z}_0) - \log q(\mathbf{z}_0|\mathbf{x})], \quad (9)$$

which can be estimated by Monte Carlo sampling since  $\log p_\theta(\mathbf{z}_0)$  can be computed with Eq. (8) and the other two terms can be designed to be tractable. The bound gap is exactly  $D_{\text{KL}}(q(\mathbf{z}_0|\mathbf{x})||p(\mathbf{z}_0|\mathbf{x}))$ , which is typically negligible in practice. When  $q(\mathbf{z}_0|\mathbf{x})$  is a *dequantization* distribution (Ho et al., 2019; Zheng et al., 2023b),  $p(\mathbf{x}|\mathbf{z}_0) = 1$  almost surely under  $q(\mathbf{z}_0|\mathbf{x})$ , and thus the  $\log p(\mathbf{x}|\mathbf{z}_0)$  term in Eq. (9) vanishes.

### 3 MERGING EXPERTS

In this section, we discuss the problem of *merging* multiple pretrained diffusion models and demonstrate how this can be applied to our specific objective: combining two pretrained experts, each specialized in one of two fundamental aspects of generative modeling—density estimation and perceptual image quality. Our approach is based on reformulating the diffusion dynamics in terms of the signal-to-noise ratio (SNR), which provides a natural framework for aligning and integrating models trained under different noise schedules. When considering standard diffusion or flow models defined by Gaussian probability paths, this formulation can be applied directly. We begin by presenting a general method for adapting a pretrained score model to a new stochastic process. We then show how this procedure can be used to merge the two expert models of interest—one optimized for likelihood and the other for perceptual fidelity.

Let the *target forward process* be defined by data-conditional marginals  $q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\alpha_t\mathbf{x}, \sigma_t^2\mathbf{I})$  with a smooth noise schedule  $t \mapsto (\alpha_t, \sigma_t)$ . We define its negative log-SNR as  $\gamma_t := -\log \frac{\alpha_t^2}{\sigma_t^2}$  which increases monotonically with  $t$ . Given a score estimate  $\mathbf{s}_\theta(\mathbf{z}_t, t) \approx \nabla \log q(\mathbf{z}_t)$ , new data samples can be approximately generated using the reverse SDE or the PF ODE associated with this target process (Eqs. (4) and (5)). In our merged model, rather than training a new score network, we employ score estimates obtained from pretrained expert models.

Each pretrained model has been trained under its own noising process with marginals  $\tilde{q}(\tilde{\mathbf{z}}_u|\mathbf{x}) = \mathcal{N}(\tilde{\alpha}_u\mathbf{x}, \tilde{\sigma}_u^2\mathbf{I})$  and corresponding noise schedule  $u \mapsto (\tilde{\alpha}_u, \tilde{\sigma}_u)$  and negative log-SNR  $\tilde{\gamma}_u$ . During training, the model learned a score estimator  $\tilde{\mathbf{s}}_\theta(\tilde{\mathbf{z}}_u, u) \approx \nabla_{\tilde{\mathbf{z}}_u} \log \tilde{q}(\tilde{\mathbf{z}}_u)$  or an equivalent quantity (e.g., noise, data, velocity (Salimans & Ho, 2022), or PF ODE vector field (Lipman et al., 2022; Liu et al., 2022b)) that can be converted to a score. The score model was trained on inputs distributed as  $\tilde{q}(\tilde{\mathbf{z}}_u) = \int q_{\text{data}}(\mathbf{x}) \mathcal{N}(\tilde{\mathbf{z}}_u; \tilde{\alpha}_u\mathbf{x}, \tilde{\sigma}_u^2\mathbf{I}) d\mathbf{x}$ . Our goal is to reuse this pretrained score function within the target process at any desired time  $t$ .

At time  $t$  of the target process, the state  $\mathbf{z}_t$  has a noise level  $\gamma_t$ . To use a pretrained model at this point, we match the noise levels of the two processes by equating their negative log-SNRs:

$$\tilde{\gamma}_u = \gamma_t \quad u = \tilde{\gamma}^{-1}(\gamma_t) \quad (10)$$

i.e., we identify the expert’s time  $u$  that corresponds to the same effective noise level as that of the target process at time  $t$ . This mapping is well-defined only where such a  $u$  exists, i.e., when the desired value of  $\gamma_t$  lies within the range  $\tilde{\gamma}([0, 1])$  of the expert’s noise schedule. Finally, we rescale  $\mathbf{z}_t$  so that it follows the same distribution as the expert’s training data:

$$\tilde{\mathbf{z}}_u := \frac{\tilde{\alpha}_u}{\alpha_t} \mathbf{z}_t = \frac{\tilde{\alpha}_u}{\alpha_t} (\alpha_t \mathbf{x} + \sigma_t \epsilon) = \tilde{\alpha}_u \mathbf{x} + \tilde{\sigma}_u \epsilon, \quad \text{with } \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

which is distributed as  $\tilde{q}(\tilde{\mathbf{z}}_u)$ , as required. Now,  $\tilde{\mathbf{s}}_\theta(\tilde{\mathbf{z}}_u, u)$  approximates  $\nabla \log \tilde{q}(\tilde{\mathbf{z}}_u)$ , whereas our goal is to approximate  $\nabla \log q(\mathbf{z}_t)$ . By a change of variables, we have:

$$\nabla_{\mathbf{z}_t} \log q(\mathbf{z}_t) = \nabla_{\tilde{\mathbf{z}}_t} \log \tilde{q}(\tilde{\mathbf{z}}_u) = \frac{\tilde{\alpha}_u}{\alpha_t} \nabla_{\tilde{\mathbf{z}}_u} \log \tilde{q}(\tilde{\mathbf{z}}_u). \quad (11)$$

Hence, the score of the target process can be approximated by:

$$\mathbf{s}_\theta(\mathbf{z}_t, t) = \frac{\tilde{\alpha}_u}{\alpha_t} \tilde{\mathbf{s}}_\theta\left(\frac{\tilde{\alpha}_u}{\alpha_t} \mathbf{z}_t, \tilde{\gamma}^{-1}(\gamma_t)\right). \quad (12)$$

In the experiments presented in this work, we restrict our attention to variance-preserving (VP) processes, for which  $\alpha_t^2 + \sigma_t^2 = 1$  for all  $t$ . Under this condition, the equality of SNRs implies that  $\tilde{\sigma}_u = \sigma_t$  and  $\tilde{\alpha}_u = \alpha_t$ . Consequently, the scaling factor cancels out and the adaptation simplifies to a straightforward time remapping:

$$\mathbf{s}_\theta(\mathbf{z}_t, t) = \tilde{\mathbf{s}}_\theta(\mathbf{z}_t, \tilde{\gamma}^{-1}(\gamma_t)). \quad (13)$$

Having established how to adapt a pretrained expert to different noise processes, we now turn to the case involving multiple pretrained experts. By applying the adaptation procedure described above, each expert can generate a score estimate that remains valid across any noise level it was trained on. To construct a unified model, these estimates can then be combined in an ensemble fashion, e.g., by taking a convex combination of the scores from the individual models.

In this work, we focus on the special case in which a hard switch is applied between two models, i.e., we assign a weight of 1 to exactly one expert at any given time, effectively employing only a single pretrained expert at each step. The concrete instantiation of this procedure with pretrained VDM (Kingma et al., 2021) and EDM (Karras et al., 2022) models is presented in Section 4.

## 4 EXPERIMENTS

To assess the effectiveness of this hybrid approach, we conduct a series of experiments combining the two pretrained diffusion experts through a hard switch during denoising. We first describe the experimental configuration, then analyze quantitative and qualitative results demonstrating how the switching threshold mediates the trade-off between perceptual quality and data likelihood.

### 4.1 EXPERIMENTAL SETUP

We use pretrained EDM (Karras et al., 2022) and VDM (Kingma et al., 2021) models and define a hard switch between them during the denoising process. The EDM model operates at high noise levels  $\gamma_t$  to enhance perceptual quality, while the VDM model is applied at low noise levels to improve likelihood (see Fig. 1). This configuration is motivated by prior findings indicating that likelihood is primarily influenced by denoising at low noise levels, whereas perceptual fidelity benefits from accurate denoising at high noise levels (Kim et al., 2021; Zheng et al., 2023b).

Let  $\gamma_t^{\text{EDM}}$  and  $\gamma_t^{\text{VDM}}$  denote the respective training noise schedules (negative log-SNR) of EDM and VDM. We introduce a switching time  $\eta \in [0, 1]$  and define a target process with noise schedule  $\gamma_t$  such that:

$$\gamma_0 = \gamma_0^{\text{VDM}} < \gamma_0^{\text{EDM}} \leq \gamma_\eta \leq \gamma_1^{\text{VDM}} < \gamma_1^{\text{EDM}} = \gamma_1 .$$

The EDM and VDM models were trained over  $\gamma^{\text{VDM}} \in [-13.3, 5]$  and  $\gamma^{\text{EDM}} \in [-12.43, 8.764]$ , respectively, implying that the merged model operates over the combined range  $\gamma \in [-13.3, 8.764]$ . For simplicity, we adopt a linear schedule, following Kingma et al. (2021):

$$\gamma_t := \gamma_0^{\text{VDM}} + t (\gamma_1^{\text{EDM}} - \gamma_0^{\text{VDM}}) , \quad t \in [0, 1] . \quad (14)$$

This formulation constrains the feasible switching time  $\eta$  to the interval

$$\eta_{\min} = \frac{\gamma_0^{\text{EDM}} - \gamma_0^{\text{VDM}}}{\gamma_1^{\text{EDM}} - \gamma_0^{\text{VDM}}} \approx 0.0394 , \quad \eta_{\max} = \frac{\gamma_1^{\text{VDM}} - \gamma_0^{\text{VDM}}}{\gamma_1^{\text{EDM}} - \gamma_0^{\text{VDM}}} \approx 0.8294 . \quad (15)$$

In our experiments, we vary the switching thresholds within the range  $\eta \in [\eta_{\min}, \eta_{\max}]$ , thereby restricting the denoising process to noise levels covered by both models. All models use the variance-preserving (VP) formulation, i.e.,  $\alpha_t^2 + \sigma_t^2 = 1$ , and operate directly in pixel space.

We report the performance of the original EDM and VDM models as baselines and further include results from previously published approaches to provide a comprehensive evaluation. The base models are evaluated under their native  $\gamma$  ranges. Our experiments are conducted on CIFAR-10 (Krizhevsky & Hinton, 2009) and ImageNet32 (Deng et al., 2009). For consistency with prior work, we adopt the original ImageNet32 variant (Van Den Oord et al., 2016) and denote it with an asterisk (\*) in our comparisons, as some other works use the updated official version (Chrabaszcz et al., 2017). Further experimental details are provided in Appendix A.

Sample quality is evaluated using the Fréchet Inception Distance (FID) (Heusel et al., 2017) on 50k generated samples following the EDM evaluation protocol of Karras et al. (2022). We also report test log-likelihood in bits per dimension (BPD) using two methods. The first estimates the standard variational lower bound (6). The second computes the exact log-likelihood of  $\mathbf{z}_0$  by integrating the PF ODE (5) while tracking the log-density with Eq. (8), and subsequently applies truncated normal dequantization (Zheng et al., 2023b). This yields the bound (9) which is tighter than (6).

### 4.2 RESULTS

We now present experimental results evaluating our merged diffusion framework, focusing on how the threshold parameter  $\eta$  mediates the trade-off between likelihood and perceptual image quality.

Table 1: Test likelihood (ODE) in bits/dimension (BPD) on CIFAR-10 and ImageNet32.

|            |     | Threshold |              |      |      |      |             |      |             |      |      |              |      |
|------------|-----|-----------|--------------|------|------|------|-------------|------|-------------|------|------|--------------|------|
|            |     | EDM       | $\eta_{min}$ | 0.1  | 0.2  | 0.3  | 0.4         | 0.5  | 0.6         | 0.7  | 0.8  | $\eta_{max}$ | VDM  |
| CIFAR-10   | NLL | 3.21      | 3.09         | 2.83 | 2.69 | 2.63 | <b>2.62</b> | 2.62 | 2.63        | 2.63 | 2.63 | 2.64         | 2.64 |
|            | NFE | 204       | 232          | 234  | 236  | 253  | 259         | 254  | 251         | 257  | 273  | 274          | 248  |
| ImageNet32 | NLL | 4.04      | 3.96         | 3.80 | 3.76 | 3.74 | 3.72        | 3.72 | <b>3.72</b> | 3.72 | 3.72 | 3.72         | 3.72 |
|            | NFE | 195       | 185          | 192  | 186  | 180  | 180         | 196  | 210         | 220  | 232  | 236          | 205  |

Table 2: FID@50k on CIFAR-10 and ImageNet32 with deterministic (ODE) sampling.

|            |     | Threshold |              |      |      |             |      |             |      |      |      |              |      |
|------------|-----|-----------|--------------|------|------|-------------|------|-------------|------|------|------|--------------|------|
|            |     | EDM       | $\eta_{min}$ | 0.1  | 0.2  | 0.3         | 0.4  | 0.5         | 0.6  | 0.7  | 0.8  | $\eta_{max}$ | VDM  |
| CIFAR-10   | FID | 2.02      | 2.04         | 2.05 | 2.03 | <b>2.01</b> | 2.14 | 2.82        | 4.75 | 6.86 | 7.67 | 7.73         | 9.37 |
|            | NFE | 125       | 145          | 147  | 159  | 169         | 173  | 193         | 221  | 239  | 226  | 238          | 206  |
| ImageNet32 | FID | 7.38      | 7.43         | 7.44 | 7.39 | 7.26        | 6.98 | <b>6.58</b> | 6.72 | 7.15 | 7.15 | 7.11         | 9.85 |
|            | NFE | 120       | 140          | 144  | 150  | 166         | 169  | 180         | 204  | 207  | 189  | 189          | 158  |

**Quantitative evaluation.** We investigate how the threshold parameter  $\eta$  in our merged model governs the balance between data likelihood and perceptual image quality. Table 1 reports negative log-likelihood (NLL) in bits per dimension (BPD) computed using the bound in Eq. (9), which relies on the PF ODE Eq. (5) and the truncated normal dequantization method proposed by Zheng et al. (2023b). The corresponding FID scores for unconditional image generation using ODE-based sampling are reported in Table 2. We vary  $\eta$  from  $\eta_{min}$  to  $\eta_{max}$  (see Eq. (15)).

Figure 2 visualizes the effect of  $\eta$  on both likelihood and perceptual quality. For likelihood, we report both the ODE-based bound (9) and the variational bound (6) which does not require ODE integration. For perceptual quality, we include FID scores obtained from deterministic sampling with an adaptive step size ODE solver, and from stochastic sampling with 256 sampling steps. For ease of interpretation, in Fig. 2, we denote pure EDM as  $\eta = 0$  and pure VDM as  $\eta = 1$ , although these values are not technically realizable in the merged model.

Varying  $\eta$  produces a clear and consistent trade-off between likelihood and perceptual quality. On CIFAR-10, the best overall operating point occurs at  $\eta = 0.3$ . Increasing  $\eta$  to 0.4 further improves the likelihood beyond the VDM baseline, with only a slight degradation in FID (2.02 to 2.14). On ImageNet32 (see Fig. 4 in Appendix B),  $\eta = 0.5$  matches the VDM baseline in likelihood while surpassing EDM in FID. **Overall, these results indicate that a single threshold  $\eta$  can outperform both base models across metrics, demonstrating that our approach effectively breaks the apparent trade-off between likelihood and perceptual quality.**

To contextualize these results, Table 3 compares our method with other approaches designed to achieve both high likelihood and strong perceptual quality. For likelihood evaluation, we use PF-ODE

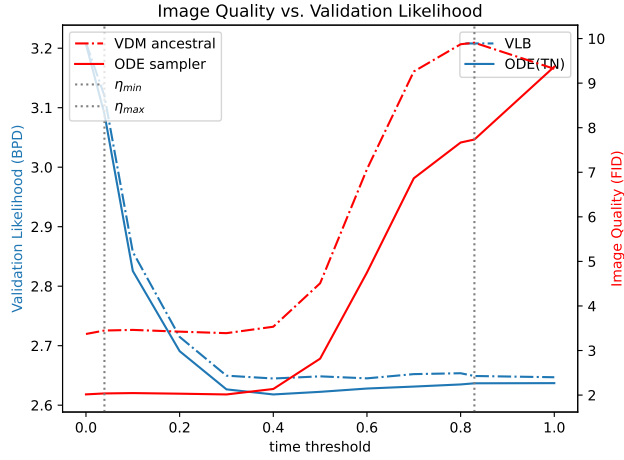


Figure 2: Likelihood-quality trade-off on CIFAR-10. Likelihood is measured in BPD using PF ODE integration with truncated normal dequantization (*ODE (TN)*) or the variational lower bound (*VLB*). Perceptual quality is measured with FID with both deterministic (*ODE sampler*) and stochastic (*VDM ancestral*) integration. The x-axis corresponds to the switching threshold  $\eta$  between the models. The EDM and VDM base models correspond to  $\eta = 0$  and  $\eta = 1$ , respectively.



Table 3: Comparison of our method with prior approaches targeting both high likelihood and strong perceptual quality. Unless otherwise noted, NLL is evaluated using truncated normal dequantization and PF ODE integration. Alternative settings are indicated as follows: Uniform Deq.<sup>†</sup>, Variational Deq.<sup>‡</sup>, VLB<sup>✓</sup>, Data Augmentation<sup>⊕</sup>, ImageNet32 (old version)\*.

| Model   | CIFAR-10               |        |     | ImageNet32               |        |     |
|---|------------------------|--------|-----|--------------------------|--------|-----|
|   | NLL(↓)                 | FID(↓) | NFE | NLL(↓)                   | FID(↓) | NFE |
| <b>Base Models</b>                                  |                        |        |     |                          |        |     |
| VDM (Kingma et al., 2021)                           | 2.65 <sup>✓</sup>      | 7.41   | -   | 3.72* <sup>✓</sup>       | -      | -   |
| EDM (w/ Heun Sampler) (Karras et al., 2022)         | -                      | 1.97   | 35  | -                        | -      | -   |
| <b>Focused on both FID and NLL</b>                  |                        |        |     |                          |        |     |
| Soft Truncation (Kim et al., 2021)                  | 3.01 <sup>†</sup>      | 3.96   | -   | 3.90* <sup>†</sup>       | 8.42*  | -   |
| CTM (⊕ - randomflip) (Kim et al., 2023)             | 2.43 <sup>†</sup>      | 1.87   | 2   | -                        | -      | -   |
| <b>Ours</b>   |                        |        |     |                          |        |     |
| VDM (our evaluation, $\gamma \in [-13.3, 5]$ )      | 2.64/2.66 <sup>✓</sup> | 9.37   | 206 | 3.72*/3.72* <sup>✓</sup> | 9.85*  | 158 |
| EDM (our evaluation, $\gamma \in [-12.43, 8.764]$ ) | 3.21                   | 2.02   | 125 | 4.04*                    | 7.38*  | 120 |
| Ours NLL ( $\eta = 0.4$ , CIFAR-10)                 | 2.62                   | 2.14   | 173 | -                        | -      | -   |
| Ours ( $\eta = 0.3$ , CIFAR-10)                     | 2.63                   | 2.01   | 169 | -                        | -      | -   |
| Ours ( $\eta = 0.5$ , ImageNet32)                   | -                      | -      | -   | 3.72*                    | 6.58*  | 180 |

likelihood estimation with truncated normal dequantization, while image quality is assessed using ODE-based sampling. Our approach outperforms Soft Truncation (Kim et al., 2021), which also seeks to balance these objectives. Consistency Trajectory Models (CTM, Kim et al., 2023) improve both metrics by combining multiple loss functions, including GAN-based objectives, and data augmentation. In contrast, our method relies solely on standard denoising objectives. A broader comparison is presented in Table 5 (Appendix B), which includes models achieving state-of-the-art results on either metric, providing a reference for current performance limits. While we do not claim state-of-the-art performance, our experiments show that merging two pretrained diffusion models—one optimized for perceptual image quality and the other for likelihood—consistently improves both metrics compared to either model used independently.

**Qualitative evaluation.** Figure 3 presents qualitative results on CIFAR-10 obtained using the ODE sampler. In this setting, the score term  $\nabla \log q(\mathbf{z}_t)$  in the PF ODE (5) is replaced by the learned score network  $s_\theta$ , and the ODE is integrated backward in time to generate samples. For each row in the figure, we use the same Gaussian latent sample  $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  and vary only the switching threshold  $\eta$  indicated in the column headers. The merged model integrates the ODE backward and transitions from the pretrained EDM model to the pretrained VDM model at the specified threshold  $\eta$ . Consequently, all trajectories are identical up to the switching point.

When sampling exclusively from the EDM model (leftmost column), we observe visually high-quality samples but relatively poor likelihood. As the switch occurs earlier in the denoising trajectory (i.e., as  $\eta$  increases), the likelihood improves while perceptual quality remains largely unchanged, up to a point where excessive reliance on the VDM component begins to degrade image fidelity. This trend aligns precisely with the intended behavior of our approach: early EDM stages produce realistic, high-quality structures, while the subsequent

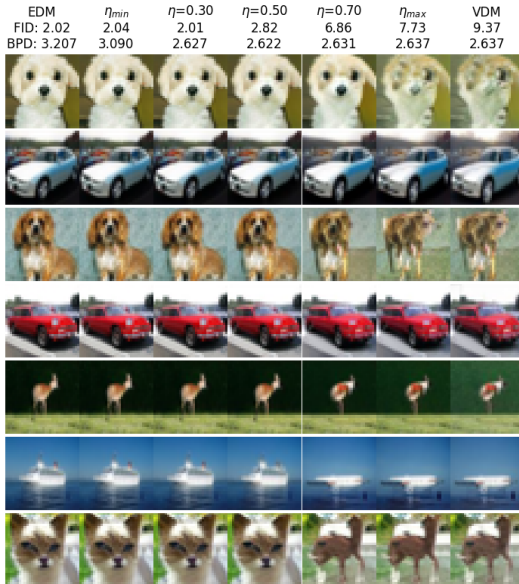


Figure 3: Qualitative comparison on CIFAR-10 using the ODE sampler. Each row starts from the same noise sample  $\mathbf{z}_1$ , while columns vary the threshold  $\eta$  in the merged model. The model follows EDM dynamics up to time  $\eta$  and then switches to VDM for  $t < \eta$ . Increasing  $\eta$  triggers an earlier switch, improving likelihood but gradually reducing perceptual fidelity.

VDM refinement enhances likelihood with only minor perceptual alterations. Notably, despite differences in architecture, training objectives, and weighting of the ELBO, the two base models frequently yield remarkably similar generated samples from the same initial noise, both individually and when combined within the merged framework.

## 5 RELATED WORK

**Likelihood experts.** Several methods focus on improving likelihood. VDM (Kingma et al., 2021) and ScoreFlow (Song et al., 2021) directly optimize (a bound on) the data log-likelihood. i-DODE (Zheng et al., 2023b) introduces *velocity prediction* and proposes an improved likelihood estimation technique. Other works (Sahoo et al., 2023; Nielsen et al., 2023; Bartosh et al., 2024) explore learnable forward processes, whereas our study focuses on standard diffusion models with fixed linear forward noise schedules.

**Sample quality experts.** Many studies improve the perceptual quality of generated samples by introducing better or more efficient samplers (Song et al., 2020a;b; Lu et al., 2022; Zheng et al., 2023a; Zhao et al., 2024; Karras et al., 2022; Zhou et al., 2024), addressing exposure bias (Ning et al., 2023), or applying alternative loss weighting strategies (Kingma & Gao, 2024; Ho et al., 2020). GMEM (Tang et al., 2024) enhances both quality and efficiency by incorporating an external memory bank into a transformer-based model, achieving state-of-the-art FID on CIFAR-10. PaGoDA (Kim et al., 2024), a distillation-based approach, achieves the best known FID on ImageNet32. In this work, we focus on UNet-based diffusion models trained with simpler objectives such as *noise prediction*, and exclude distillation-based methods from our scope.

**Experts on both metrics.** Soft Truncation (Kim et al., 2021) proposes a training strategy that softens fixed truncation into a random variable, adjusting loss weighting across diffusion times to address the likelihood–quality trade-off. While aligned in motivation with our work, their approach requires training from scratch. In contrast, our method directly leverages existing pretrained models. CTM (Kim et al., 2023) uses a combination of loss terms, including an additional GAN loss, along with data augmentation to improve both metrics. In contrast, we address the trade-off from a different perspective, by merging experts trained with the standard denoising objective.

**Mixture-of-Experts.** Mixture-of-Experts (MoE) frameworks have been applied to diffusion models in contexts such as zero-shot text-to-image generation (Balaji et al., 2022; Feng et al., 2023) and controllable image synthesis (Bar-Tal et al., 2023). More recently, MDM (Kang et al., 2024) proposed a MoE strategy where each expert is trained on a specific time interval. While effective, their method employs identical architectures across experts and primarily targets training efficiency and sample quality. To the best of our knowledge, we are the first to address this trade-off by merging pretrained experts specialized separately in likelihood and sample quality.

## 6 CONCLUSION

We proposed a simple yet effective approach for merging pretrained diffusion or flow models to mitigate the trade-off between likelihood and perceptual image quality. By switching between an expert optimized for perceptual fidelity and another optimized for data likelihood, the hybrid model achieves consistent improvements across both objectives. On CIFAR-10 and ImageNet32, it matches or surpasses the performance of its individual components, demonstrating that complementary models can be combined to yield stronger generative behavior without retraining.

A key advantage of the method is its complete reliance on existing pretrained models, requiring no fine-tuning or additional supervision. Its simplicity, however, comes with limitations: performance depends on the characteristics of the merged models, and the optimal switching threshold must be determined empirically. Furthermore, our experiments are restricted to pixel-space diffusion models, leaving room for adaptation to other architectures and training regimes.

Future research may explore automated or learned switching mechanisms, integration with advanced samplers, and extensions to latent or consistency-based diffusion models. Overall, this work highlights model merging as a lightweight yet powerful tool for enhancing pretrained diffusion systems, suggesting new directions for efficient and modular generative modeling.



## REFERENCES

- Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.
- Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.
- Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. *arXiv preprint arXiv:2302.08113*, 2023.
- Grigory Bartosh, Dmitry Vetrov, and Christian A Naesseth. Neural diffusion models. *arXiv preprint arXiv:2310.08337*, 2023.
- Grigory Bartosh, Dmitry Vetrov, and Christian A Naesseth. Neural flow diffusion models: Learnable forward process for improved diffusion modelling. *arXiv preprint arXiv:2404.12940*, 2024.
- Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.
- Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- J.R. Dormand and P.J. Prince. A family of embedded runge-kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980. ISSN 0377-0427.
- Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.
- Zhida Feng, Zhenyu Zhang, Xintong Yu, Yewei Fang, Lanxin Li, Xuyi Chen, Yuxiang Lu, Jiayang Liu, Weichong Yin, Shikun Feng, et al. Ernie-vilg 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10135–10145, 2023.
- Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.
- Matej Grčić, Ivan Grubišić, and Siniša Šegvić. Densely connected normalizing flows. *Advances in Neural Information Processing Systems*, 34:23968–23982, 2021.
- Louay Hazami, Rayhane Mama, and Ragavan Thuraiaratnam. Efficientvdvae: Less is more. *arXiv preprint arXiv:2203.13751*, 2022.
- 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. *Advances in neural information processing systems*, 30, 2017.
- Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pp. 2722–2730. PMLR, 2019.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

- Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021.
- Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.
- Seoungyoon Kang, Yunji Jung, and Hyunjung Shim. Local expert diffusion models for efficient training in denoising diffusion probabilistic models. In *2nd Workshop on Sustainable AI*, 2024.
- Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.
- Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. *arXiv preprint arXiv:2106.05527*, 2021.
- Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.
- Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Yuhta Takida, Naoki Murata, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. Pagoda: Progressive growing of a one-step generator from a low-resolution diffusion teacher. *arXiv preprint arXiv:2405.14822*, 2024.
- Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36, 2024.
- Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.
- Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.
- Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, ON, Canada, 2009.
- Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022a.
- Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022b.
- Aaron Lou and Stefano Ermon. Reflected diffusion models. In *International Conference on Machine Learning*, pp. 22675–22701. PMLR, 2023.
- Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.
- Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.
- Beatrix MG Nielsen, Anders Christensen, Andrea Dittadi, and Ole Winther. Diffenc: Variational diffusion with a learned encoder. *arXiv preprint arXiv:2310.19789*, 2023.
- Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. *arXiv preprint arXiv:2308.15321*, 2023.

- Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.
- Subham Sekhar Sahoo, Aaron Gokaslan, Chris De Sa, and Volodymyr Kuleshov. Diffusion models with learned adaptive noise. *arXiv preprint arXiv:2312.13236*, 2023.
- Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.
- John Skilling. The eigenvalues of mega-dimensional matrices. *Maximum Entropy and Bayesian Methods: Cambridge, England, 1988*, pp. 455–466, 1989.
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.
- Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.
- Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.
- Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.
- Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.
- Yi Tang, Peng Sun, Zhenglin Cheng, and Tao Lin. Generative modeling with explicit memory. *arXiv preprint arXiv:2412.08781*, 2024.
- Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.
- Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021.
- Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016.
- Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.
- Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.
- Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. *Advances in Neural Information Processing Systems*, 36: 55502–55542, 2023a.
- Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Improved techniques for maximum likelihood estimation for diffusion odes. In *International Conference on Machine Learning*, pp. 42363–42389. PMLR, 2023b.
- Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *International Conference on Machine Learning*, 2024.

## A EXPERIMENTAL DETAILS

**Datasets.** We conduct experiments on the CIFAR-10 and ImageNet32 datasets. For CIFAR-10 (Krizhevsky & Hinton, 2009), we use the standard version distributed through PyTorch. For ImageNet32, we use its original version (Deng et al., 2009; Van Den Oord et al., 2016), which is no longer officially distributed but has been made available by Zheng et al. (2023b). Since some prior works report results using the newer official release of ImageNet32 (Chrabaszcz et al., 2017), we indicate, for each baseline, which dataset version their results correspond to.

**Models and training.** We use the publicly available EDM checkpoint for CIFAR-10 (Krizhevsky & Hinton, 2009).<sup>1</sup> For VDM, we train the PyTorch re-implementation<sup>2</sup> based on the architecture described in Kingma et al. (2021). The model is trained for 10 million steps on 8xA100 (40GB) GPUs, with no data augmentation, a fixed linear  $\gamma$  schedule, and a batch size of 128. The resulting model achieves 2.64 BPD on the test set (PF ODE likelihood with TN dequantization) and 2.66 BPD under VLB evaluation.

For ImageNet32, VDM is trained similarly to CIFAR-10, but with 256 channels and a total batch size of 512 for 2 million steps, following Kingma et al. (2021). Since no pretrained EDM model is publicly available for ImageNet32, we train one using the official EDM code, with parameters `--cond 0 --arch ddpmp --duration 1000`. Training is performed for 1000M images with a total batch size of 1024. No hyperparameter tuning was performed in this case.

**Evaluation.** To compute the divergence of the PF ODE vector field when evaluating  $\log p_\theta(\mathbf{z}_0)$  in Eq. (9), we use the Skilling–Hutchinson trace estimator (Skilling, 1989; Hutchinson, 1989):

$$\nabla \cdot \mathbf{h}_\theta(\mathbf{z}_t, t) = \text{tr} \left( \frac{\partial \mathbf{h}_\theta(\mathbf{z}_t, t)}{\partial \mathbf{z}_t} \right) = \mathbb{E}_{p(\epsilon)} \left[ \epsilon^\top \frac{\partial \mathbf{h}_\theta(\mathbf{z}_t, t)}{\partial \mathbf{z}_t} \epsilon \right] \quad (16)$$

where  $\frac{\partial \mathbf{h}_\theta(\mathbf{z}_t, t)}{\partial \mathbf{z}_t}$  is the Jacobian of  $\mathbf{h}_\theta$ , and  $\epsilon$  is a random variable such that  $\mathbb{E}_{p(\epsilon)}[\epsilon] = \mathbf{0}$  and  $\text{Cov}_{p(\epsilon)}[\epsilon] = \mathbf{I}$ . We use the Rademacher distribution for  $p(\epsilon)$  and use the RK45 ODE solver (Dormand & Prince, 1980) with `atol=1e-5` and `rtol=1e-5`, following prior work (Sahoo et al., 2023; Song et al., 2020b; Zheng et al., 2023b). For all evaluations, we use Exponential Moving Average (EMA) weights. FID scores are computed following Karras et al. (2022), with reference statistics calculated from the training sets.

## B ADDITIONAL RESULTS AND VISUALIZATIONS

Table 4 reports likelihood evaluations of the merged model using the Variational Lower Bound (VLB) from Eq. (6). We report the mean and standard deviation over 10 runs. Fig. 4 shows the trade-off between likelihood and perceptual quality on ImageNet32. For likelihood we use both the ODE (Eq. (9), with truncated normal dequantization) and SDE (Table 4) bounds, and for evaluating perceptual quality we use both the ODE sampler and ancestral sampling. The EDM and VDM baselines correspond to  $\eta = 0.0$  and  $\eta = 1.0$ , respectively, and do not involve expert switching. Note again here it’s improper to define them as  $\eta = 0.0$  and  $\eta = 1.0$  but convenient for plotting.

Fig. 5 shows generated samples from our merged model on ImageNet32. For each row, we fix a noise sample  $\mathbf{z}_1 \sim p(\mathbf{z}_1)$  and generate samples with the merged model varying the switching threshold  $\eta$ . In the leftmost column of the figure (corresponding to EDM), the samples exhibit excellent perceptual quality but poor likelihood. As  $\eta$  increases, likelihood improves while image quality remains largely unchanged. Around  $\eta = 0.5$ , the model achieves the same likelihood as the VDM expert while surpassing the EDM baseline in FID.

In Figs. 6 to 9, we present randomly generated samples (without fixed noise samples  $\mathbf{z}_1$ ) from the baseline models and the merged models (across a range of  $\eta$  values), both for CIFAR-10 and ImageNet32, using both deterministic and stochastic sampling.

<sup>1</sup><https://nvlabs-fi-cdn.nvidia.com/edm/pretrained/edm-cifar10-32x32-uncond-vp.pkl>

<sup>2</sup><https://github.com/addtt/variational-diffusion-models>

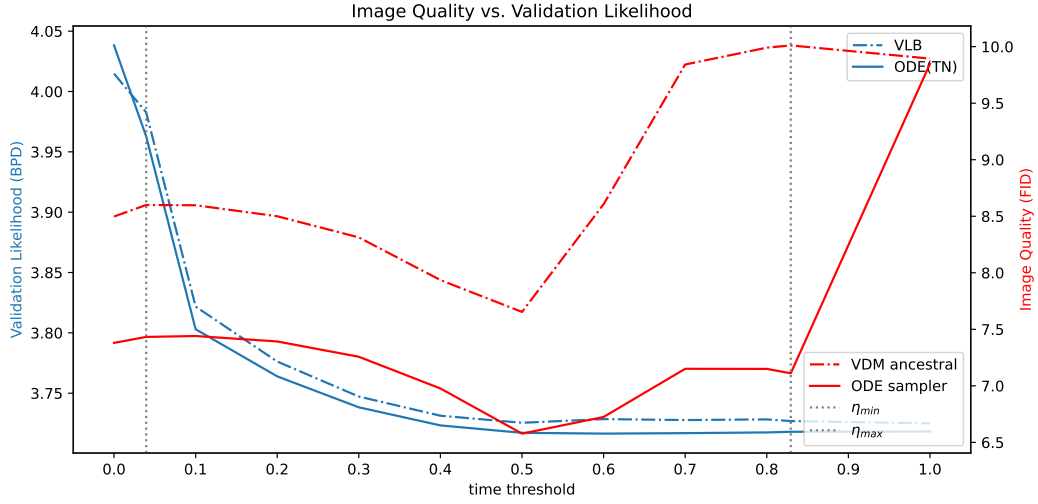


Figure 4: Likelihood–quality trade-off on ImageNet32. Likelihood is measured in BPD using PF ODE integration with truncated normal dequantization (*ODE (TN)*) or the variational lower bound (*VLB*). Perceptual quality is measured with FID with both deterministic (*ODE sampler*) and stochastic (*VDM ancestral*) integration. The x-axis corresponds to the switching threshold  $\eta$  between the models. The EDM and VDM base models correspond to  $\eta = 0$  and  $\eta = 1$ , respectively.



Figure 5: Generated images from our merged model using different thresholds  $\eta$  on ImageNet32 dataset.



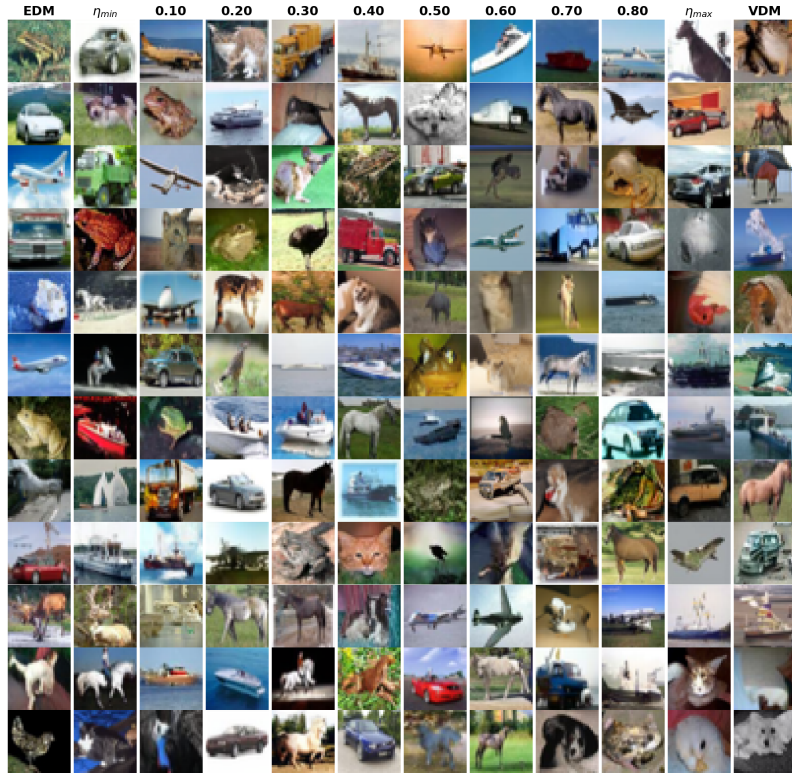


Figure 6: Random samples on CIFAR-10 using ODE sampler (with different  $\eta$ ).

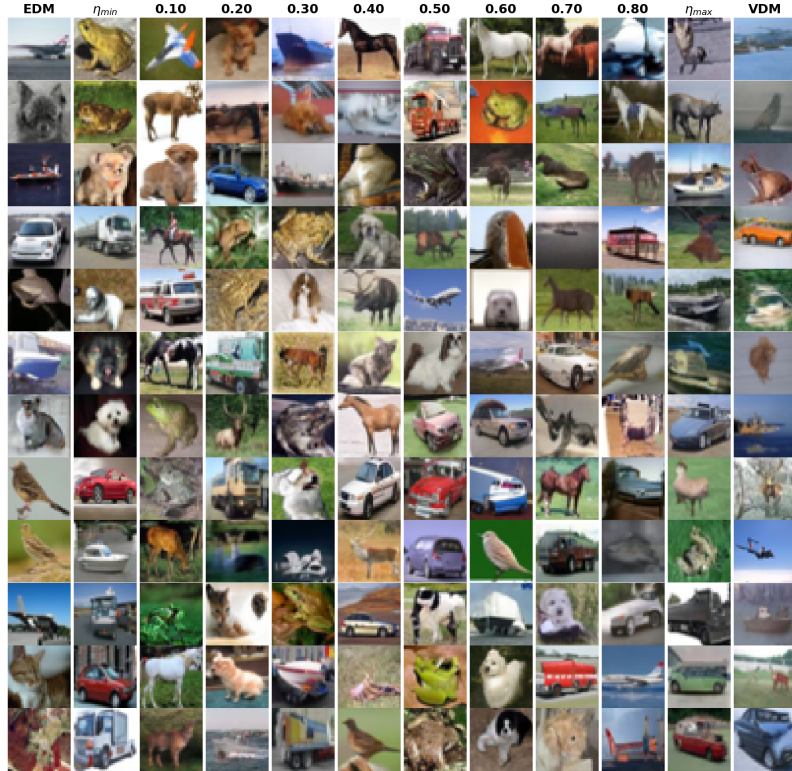


Figure 7: Random samples on CIFAR-10 using VDM ancestral sampler (with different  $\eta$ ).

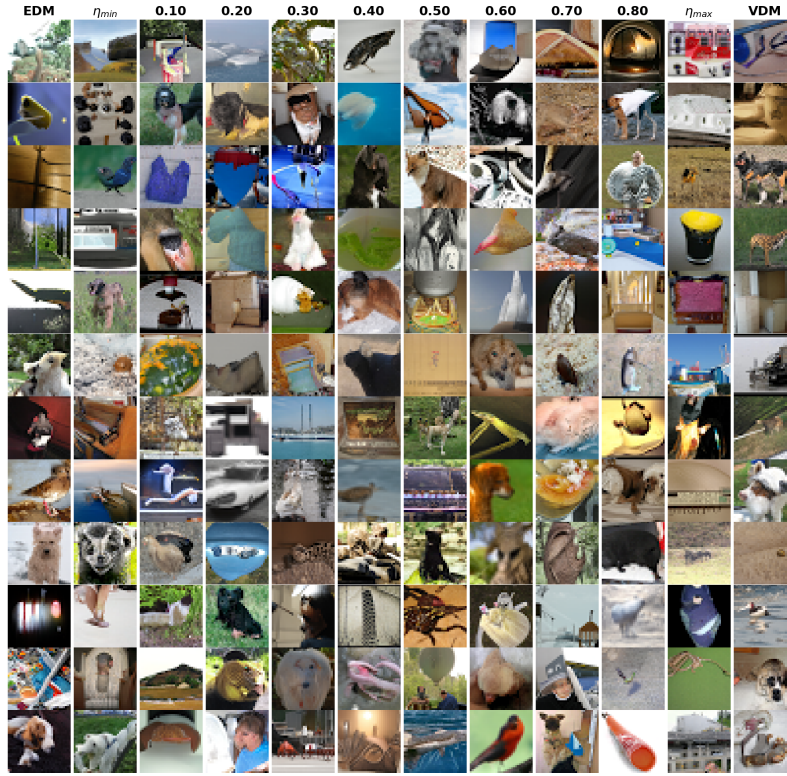


Figure 8: Random samples on ImageNet32 using ODE sampler (with different  $\eta$ ).

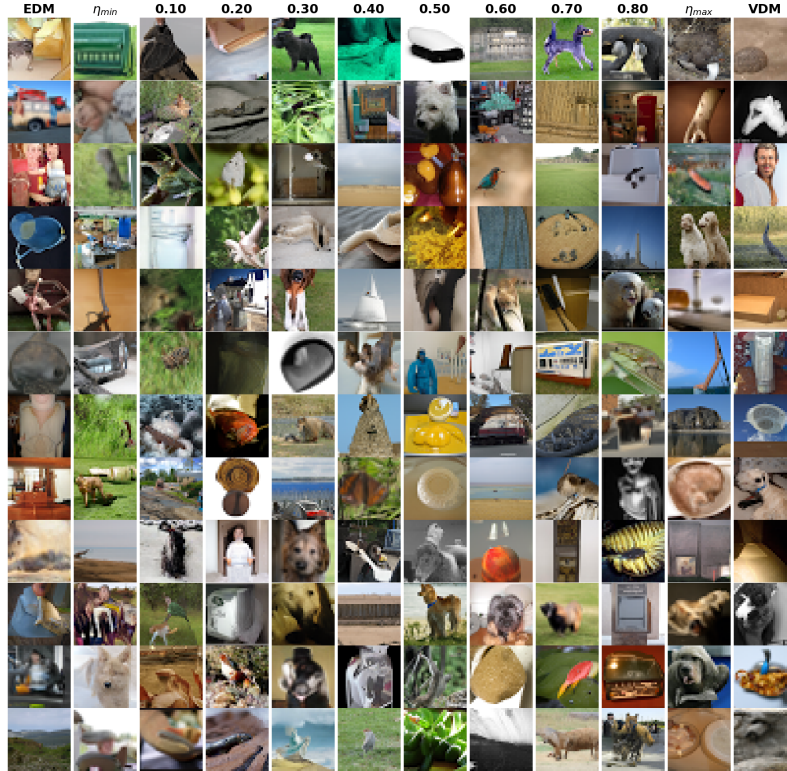


Figure 9: Random samples on ImageNet32 using VDM ancestral sampler (with different  $\eta$ ).

Table 4: VLB evaluation in terms of bits per dimension (BPD) on CIFAR-10 and ImageNet32.

| Threshold     | CIFAR-10             |       | ImageNet32           |       |
|---------------|----------------------|-------|----------------------|-------|
|               | Mean( $\downarrow$ ) | Std   | Mean( $\downarrow$ ) | Std   |
| 0.0           | 3.21                 | 0.06  | 4.01                 | 0.004 |
| $\eta_{\min}$ | 3.12                 | 0.005 | 3.98                 | 0.004 |
| 0.1           | 2.86                 | 0.009 | 3.82                 | 0.004 |
| 0.2           | 2.71                 | 0.008 | 3.78                 | 0.004 |
| 0.3           | 2.65                 | 0.007 | 3.75                 | 0.005 |
| 0.4           | 2.64                 | 0.008 | 3.73                 | 0.005 |
| 0.5           | 2.65                 | 0.009 | 3.73                 | 0.004 |
| 0.6           | 2.65                 | 0.007 | 3.73                 | 0.005 |
| 0.7           | 2.65                 | 0.008 | 3.73                 | 0.007 |
| 0.8           | 2.65                 | 0.009 | 3.73                 | 0.005 |
| $\eta_{\max}$ | 2.65                 | 0.008 | 3.73                 | 0.002 |
| 1.0           | 2.65                 | 0.005 | 3.73                 | 0.004 |

Table 5: Extended version of Table 3 with additional methods from the literature. Unless otherwise noted, NLL is evaluated using truncated normal dequantization and PF ODE integration. Alternative settings are indicated as follows: Uniform Deq.<sup>†</sup>, Variational Deq.<sup>‡</sup>, VLB<sup>∇</sup>, Data Augmentation<sup>⊙</sup>, ImageNet32(old version)\*.

| Model   | CIFAR-10               |                     |     | ImageNet32               |                     |     |
|---|------------------------|---------------------|-----|--------------------------|---------------------|-----|
|   | NLL( $\downarrow$ )    | FID( $\downarrow$ ) | NFE | NLL( $\downarrow$ )      | FID( $\downarrow$ ) | NFE |
| <b>Base Models</b>  |                        |                     |     |                          |                     |     |
| VDM (Kingma et al., 2021)   | 2.65 <sup>∇</sup>      | 7.41                | -   | 3.72* <sup>∇</sup>       | -                   | -   |
| EDM (w/ Heun Sampler) (Karras et al., 2022)                               | -                      | 1.97                | 35  | -                        | -                   | -   |
| <b>Focused on both FID-NLL</b>  |                        |                     |     |                          |                     |     |
| Soft Truncation (Kim et al., 2021)  | 3.01 <sup>†</sup>      | 3.96                | -   | 3.90* <sup>†</sup>       | 8.42*               | -   |
| CTM ( $\ominus$ - randomflip) (Kim et al., 2023)                          | 2.43 <sup>†</sup>      | 1.87                | 2   | -                        | -                   | -   |
| ScoreSDE ( $\ominus$ - randomflip) (Song et al., 2020b)                   | 2.99 <sup>†</sup>      | 2.92                | -   | -                        | -                   | -   |
| LSGM (FID) (Vahdat et al., 2021)  | 3.43                   | 2.10                | -   | -                        | -                   | -   |
| DDPM++ cont. (deep, sub-VP) (Song et al., 2020b)                          | 2.99 <sup>†</sup>      | 2.92                | -   | -                        | -                   | -   |
| Reflected Diffusion Models (Lou & Ermon, 2023)                            | 2.68                   | 2.72                | -   | 3.74                     | -                   | -   |
| <b>Focused on FID</b>   |                        |                     |     |                          |                     |     |
| GMEM (Transformer-based) (Tang et al., 2024)                              | -                      | 1.22                | 50  | -                        | -                   | -   |
| PaGoDA (distillation-based) (Kim et al., 2024)                            | -                      | -                   | -   | -                        | 0.79                | 1   |
| SiD (distillation-based) (Zhou et al., 2024)                              | -                      | 1.923               | 1   | -                        | -                   | -   |
| ScoreFlow (VP, FID) (Song et al., 2021)                                   | 3.04 <sup>‡</sup>      | 3.98                | -   | 3.84* <sup>‡</sup>       | 8.34*               | -   |
| PNM (Liu et al., 2022a)   | -                      | 3.26                | -   | -                        | -                   | -   |
| <b>Focused on NLL</b>   |                        |                     |     |                          |                     |     |
| i-DODE (VP) (Zheng et al., 2023b)   | 2.57                   | 10.74               | 126 | 3.43/3.70*               | 9.09                | 152 |
| i-DODE (VP, $\ominus$ ) (Zheng et al., 2023b)                             | 2.42                   | 3.76                | 215 | -                        | -                   | -   |
| Flow Matching (Lipman et al., 2022)                                       | 2.99 <sup>†</sup>      | 6.35                | 142 | 3.53 <sup>†</sup>        | 5.02                | 122 |
| DiffEnc (Nielsen et al., 2023)  | 2.62 <sup>∇</sup>      | 11.1                | -   | 3.46 <sup>∇</sup>        | -                   | -   |
| NDM ( $\ominus$ - horizontalflip) (Bartosh et al., 2023)                  | 2.70 <sup>†</sup>      | -                   | -   | 3.55                     | -                   | -   |
| NFDM (Gaussian q, $\ominus$ - horizontalflip) (Bartosh et al., 2024)      | 2.49 <sup>†</sup>      | 21.88               | 12  | 3.36                     | 24.74               | 12  |
| NFDM (non-Gaussian q, $\ominus$ - horizontalflip) (Bartosh et al., 2024)  | 2.48 <sup>†</sup>      | -                   | -   | 3.34                     | -                   | -   |
| NFDM-OT( $\ominus$ - horizontalflip) (Bartosh et al., 2024)               | 2.62 <sup>†</sup>      | 5.20                | 12  | 3.45                     | 4.11                | 12  |
| ScoreFlow (deep, sub-VP, NLL) (Song et al., 2021)                         | 2.81 <sup>‡</sup>      | 5.40                | -   | 3.76* <sup>‡</sup>       | 10.18*              | -   |
| Stochastic Interp. (Albergo & Vanden-Eijnden, 2022)                       | 2.99 <sup>†</sup>      | 10.27               | -   | 3.48 <sup>†</sup>        | 8.49                | -   |
| MuLAN (w/o IS $k=1$ ) (Sahoo et al., 2023)                                | 2.59                   | -                   | -   | 3.71                     | -                   | -   |
| MuLAN (w/ IS $k=20$ ) (Sahoo et al., 2023)                                | 2.55                   | -                   | -   | 3.67                     | -                   | -   |
| Improved DDPM ( $L_{\text{vib}}$ ) (Nichol & Dhariwal, 2021)              | 2.94 <sup>∇</sup>      | 11.47               | -   | -                        | -                   | -   |
| FFJORD (Grathwohl et al., 2018)   | 3.4                    | -                   | -   | -                        | -                   | -   |
| Improved DDPM ( $L_{\text{vib}}$ ) (Nichol & Dhariwal, 2021)              | 2.94 <sup>∇</sup>      | 11.47               | -   | -                        | -                   | -   |
| ARDM-Upscale 4 (autoregressive) (Hoogeboom et al., 2021)                  | 2.64                   | -                   | -   | -                        | -                   | -   |
| Efficient-VDVAE (Hazami et al., 2022)                                     | 2.87 <sup>∇</sup>      | -                   | -   | 3.58                     | -                   | -   |
| DenseFlow-74-10 (Grcić et al., 2021)                                      | 2.98 <sup>‡</sup>      | 34.90               | -   | 3.63                     | -                   | -   |
| <b>Ours</b>   |                        |                     |     |                          |                     |     |
| VDM (our evaluation, $\gamma \in [-13.3, 5]$ ) (Kingma et al., 2021)      | 2.64/2.66 <sup>∇</sup> | 9.37                | 206 | 3.72*/3.72* <sup>∇</sup> | 9.85*               | 158 |
| EDM (our evaluation, $\gamma \in [-12.43, 8.764]$ ) (Karras et al., 2022) | 3.21                   | 2.02                | 125 | 4.04*                    | 7.38*               | 120 |
| Ours NLL ( $\eta = 0.4$ , CIFAR-10)                                       | 2.62                   | 2.14                | 173 | -                        | -                   | -   |
| Ours ( $\eta = 0.3$ , CIFAR-10)   | 2.63                   | 2.01                | 169 | -                        | -                   | -   |
| Ours ( $\eta = 0.5$ , ImageNet32)   | -                      | -                   | -   | 3.72*                    | 6.58*               | 180 |