Improved Techniques for Maximum Likelihood Estimation for Diffusion ODEs

Kaiwen Zheng [*1] **Cheng Lu** [*1] **Jianfei Chen** [1] **Jun Zhu** [1 2]

Abstract

Diffusion models have exhibited excellent performance in various domains. The probability flow ordinary differential equation (ODE) of diffusion models (i.e., diffusion ODEs) is a particular case of continuous normalizing flows (CNFs), which enables deterministic inference and exact likelihood evaluation. However, the likelihood estimation results by diffusion ODEs are still far from those of the state-of-the-art likelihood-based generative models. In this work, we propose several improved techniques for maximum likelihood estimation for diffusion ODEs, including both training and evaluation perspectives. For training, we propose velocity parameterization and explore variance reduction techniques for faster convergence. We also derive an error-bounded high-order flow matching objective for finetuning, which improves the ODE likelihood and smooths its trajectory. For evaluation, we propose a novel training-free truncated-normal dequantization to fill the training-evaluation gap commonly existing in diffusion ODEs. Building upon these techniques, we achieve state-of-the-art likelihood estimation results on image datasets (2.56 on CIFAR-10, 3.43/3.69 on ImageNet-32) without variational dequantization or data augmentation.

1. Introduction

Likelihood is an important metric to evaluate density estimation models, and accurate likelihood estimation is the key for many applications such as data compression (Ho et al., 2021; Helminger et al., 2020; Kingma et al., 2021; Yang & Mandt, 2022), anomaly detection (Chen et al., 2018c; Dias et al., 2020) and out-of-distribution detection (Serrà et al., 2020; Xiao et al., 2020). Many deep generative models can compute tractable likelihood, including autoregressive models (Oord et al., 2016; Salimans et al., 2017; Chen et al., 2018b), variational auto-encoders (VAE) (Kingma & Welling, 2014; Vahdat & Kautz, 2020), normalizing flows (Dinh et al., 2017; Kingma & Dhariwal, 2018; Ho et al., 2019) and diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021c;a; Karras et al., 2022). Among these models, recent work named variational diffusion models (VDM) (Kingma et al., 2021) achieves state-of-the-art likelihood estimation performance on standard image density estimation benchmarks, which is a variant of diffusion models.

There are two types of diffusion models, one is based on the reverse stochastic differential equation (SDE) (Song et al., 2021c), named as *diffusion SDE*; the other is based on the probability flow ordinary differential equation (ODE) (Song et al., 2021c), named as *diffusion ODE*. These two types of diffusion models define and evaluate the likelihood in different manners: diffusion SDE can be understood as an infinitely-deep VAE (Huang et al., 2021) and can only compute a variational lower bound of the likelihood (Song et al., 2021c; Kingma et al., 2021); while diffusion ODE is a variant of continuous normalizing flows (Chen et al., 2018a) and can compute the exact likelihood by ODE solvers. Thus, it is natural to hypothesize that the likelihood performance of diffusion ODEs may be better than that of diffusion SDEs. However, all existing methods for training diffusion ODEs (Song et al., 2021b; Lu et al., 2022a; Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022; Liu et al., 2022b) cannot even achieve a comparable likelihood performance with VDM, which belongs to diffusion SDEs. It still remains largely open whether diffusion ODEs are also great likelihood estimators.

Real-world data is usually discrete, and evaluating the likelihood of discrete data by diffusion ODEs needs to first perform a dequantization process (Dinh et al., 2017; Salimans et al., 2017) to make sure the input data of diffusion ODEs is continuous. In this work, we observe that previous likelihood evaluation of diffusion ODEs has flaws in the dequantization process: the uniform dequantization (Song et al., 2021b) causes a large training-evaluation gap, and the variational dequantization (Ho et al., 2019; Song et al., 2021b) requires additional training overhead and is hard to train to the optimal.

[*]Equal contribution [1]Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Center, Tsinghua-Bosch Joint ML Center, THBI Lab, Tsinghua University [2]Pazhou Lab (Huangpu), Guangzhou, China. Correspondence to: Jun Zhu <dcszj@tsinghua.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

In this work, we propose several improved techniques, including both the evaluation perspective and training perspective, to allow the likelihood estimation by diffusion ODEs to outperform the existing state-of-the-art likelihood estimators. In the aspect of evaluation, we propose a training-free dequantization method dedicated to diffusion models by a carefully-designed truncated-normal distribution, which can fit diffusion ODEs well and improve the likelihood evaluation by a large margin compared to uniform dequantization. We also introduce an importance weighted likelihood estimator to get a tighter bound. In the aspect of training, we split our training into pretraining and finetuning phases. For pretraining, we propose a new model parameterization method including velocity parameterization, which is an extended version of flow matching (Lipman et al., 2022) with practical modifications, and log-signal-to-noise-ratio timed parameterization. Besides, we find a simple yet efficient importance sampling strategy for variance reduction. Together, our pretraining has a faster convergence speed compared to previous work. For finetuning, we propose an error-bounded high-order flow matching objective, which not only improves the ODE likelihood but also results in smoother trajectories. Together, we name our framework Improved Diffusion ODE (i-DODE).

We conduct ablation studies to demonstrate the effectiveness of separate parts. Our experimental results empirically achieve the state-of-the-art likelihood on image datasets (2.56 on CIFAR-10, 3.43/3.69 on ImageNet-32), surpassing the previous best ODEs of 2.90 and 3.48/3.82, with the superiority that we use no data augmentation and throw away the need for training variational dequantization models.

2. Diffusion Models

2.1. Diffusion ODEs and Maximum Likelihood Training

Suppose we have a d-dimensional data distribution $q_0(\boldsymbol{x}_0)$. Diffusion models (Ho et al., 2020; Song et al., 2021c) gradually diffuse the data by a forward stochastic differential equation (SDE) starting from $\boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0)$:

$$\mathrm{d}\boldsymbol{x}_t = f(t)\boldsymbol{x}_t\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}_t, \quad \boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0), \tag{1}$$

where $f(t), g(t) \in \mathbb{R}$ are manually designed noise schedules and $\boldsymbol{w}_t \in \mathbb{R}^d$ is a standard Wiener process. The forward process $\{\boldsymbol{x}_t\}_{t\in[0,T]}$ is accompanied with a series of marginal distributions $\{q_t\}_{t\in[0,T]}$, so that $q_T(\boldsymbol{x}_T) \approx \mathcal{N}(\boldsymbol{x}_T|\boldsymbol{0}, \sigma_T^2\boldsymbol{I})$ with some constant $\sigma_T > 0$. Since this is a simple linear SDE, the transition kernel is an analytical Gaussian (Song et al., 2021c): $q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\alpha_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I})$, where the coefficients satisfy $f(t) = \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}$, $g^2(t) = \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\sigma_t^2$ (Kingma et al., 2021). Under some regularity conditions (Anderson, 1982), the forward process has an equivalent probability flow ODE (Song et al., 2021c):

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = f(t)\boldsymbol{x}_t - \frac{1}{2}g^2(t)\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t), \tag{2}$$

which produces the same marginal distribution q_t at each time t as that in Eqn. (1). The only unknown term $\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t)$ is the *score function* of q_t. By parameterizing a *score network* $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ to predict the time-dependent $\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t)$, we can replace the true score function, resulting in the *diffusion ODE* (Song et al., 2021c):

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = f(t)\boldsymbol{x}_t - \frac{1}{2}g^2(t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t), \tag{3}$$

with the associated marginal distributions $\{p_t\}_{t\in[0,T]}$. Diffusion ODEs are special cases of *continuous normalizing flows* (CNFs) (Chen et al., 2018a), thus can perform exact inference of the latents and exact likelihood evaluation.

Though traditional maximum likelihood training methods for CNFs (Grathwohl et al., 2019) are feasible for diffusion ODEs, the training costs of these methods are quite expensive and hard to scale up because of the requirement of solving ODEs at each iteration. Instead, a more practical way is to match the generative probability flow $\{p_t\}_{t\in[0,T]}$ with $\{q_t\}_{t\in[0,T]}$ by a simulation-free approach. Specifically, Lu et al. (2022a) proves that $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$ can be formulated by $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}}) = D_{\mathrm{KL}}(q_T \parallel p_T^{\mathrm{ODE}}) + \mathcal{J}_{\mathrm{ODE}}(\theta)$, where

$$\mathcal{J}_{\mathrm{ODE}}(\theta) := \frac{1}{2}\int_0^T g(t)^2\mathbb{E}_{q_t(\boldsymbol{x}_t)}\Big[(\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t))^\top (\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t))\Big]\mathrm{d}t \tag{4}$$

However, computing $\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x}_t)$ requires solving another ODE and is also expensive (Lu et al., 2022a). To minimize $\mathcal{J}_{\mathrm{ODE}}(\theta)$ in a simulation-free manner, Lu et al. (2022a) also proposes a combination of $g^2(t)$ weighted first-order and high-order score matching objectives. Particularly, the first-order score matching objective is

$$\mathcal{J}_{\mathrm{SM}}(\theta) := \int_0^T \frac{g^2(t)}{2\sigma_t^2}\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\|\sigma_t\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) + \boldsymbol{\epsilon}\|_2^2\right]\mathrm{d}t, \tag{5}$$

where $\boldsymbol{x}_t = \alpha_t\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}$, $\boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}|\boldsymbol{0}, \boldsymbol{I})$.

2.2. Log-SNR Timed Diffusion Models

Diffusion models have manually designed noise schedule α_t, σ_t, which has high freedom and affects the performance. Even for restricted design space such as Variance Preserving (VP) (Song et al., 2021c), which constrains the noise schedule by $\alpha_t^2 + \sigma_t^2 = 1$, we could still have various choices about how fast α_t, σ_t changes w.r.t time t. To decouple the specific schedule form, variational diffusion

models (VDM) (Kingma et al., 2021) use a negative log-signal-to-noise-ratio (log-SNR) for the time variable and can greatly simplify both noise schedules and training objectives. Specifically, denote $\gamma_t = -\text{log-SNR}(t) = -\log \frac{\alpha_t^2}{\sigma_t^2}$, the change-of-variable relation from γ to t is

$$\frac{\mathrm{d}\gamma}{\mathrm{d}t} = \frac{g^2(t)}{\sigma_t^2}, \tag{6}$$

and replace the time subscript with γ, we get the simplified score matching objective with likelihood weighting:

$$\mathcal{J}_{\text{SM}}(\theta) = \frac{1}{2}\int_{\gamma_0}^{\gamma_T} \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\|\sigma_t \boldsymbol{s}_\theta(\boldsymbol{x}_\gamma, \gamma) + \boldsymbol{\epsilon}\|_2^2\right] \mathrm{d}\gamma \tag{7}$$

This result is in accordance with the continuous diffusion loss in Kingma et al. (2021).

2.3. Dequantization for Density Estimation

Many real-world datasets usually contain discrete data, such as images or texts. In such cases, learning a continuous density model to these discrete data points will cause degenerate results (Uria et al., 2013) and cannot provide meaningful density estimations. A common solution is *dequantization* (Dinh et al., 2017; Salimans et al., 2017; Ho et al., 2019). Specifically, suppose $\boldsymbol{x}_0$ is 8-bit discrete data scaled to $[-1, 1]$. Dequantization methods assume that we have trained a continuous model distribution p_{model} for $\boldsymbol{x}_0$, and define the discrete model distribution by

$$P_{\text{model}}(\boldsymbol{x}_0) \coloneqq \int_{[-\frac{1}{256}, \frac{1}{256})^d} p_{\text{model}}(\boldsymbol{x}_0 + \boldsymbol{u})\mathrm{d}\boldsymbol{u}.$$

To train $P_{\text{model}}(\boldsymbol{x}_0)$ by maximum likelihood estimation, variational dequantization (Ho et al., 2019) introduces a dequantization distribution $q(\boldsymbol{u}|\boldsymbol{x}_0)$ and jointly train p_{model} and $q(\boldsymbol{u}|\boldsymbol{x}_0)$ by a variational lower bound:

$$\log P_{\text{model}}(\boldsymbol{x}_0) \geq \mathbb{E}_{q(\boldsymbol{u}|\boldsymbol{x}_0)}[\log p_{\text{model}}(\boldsymbol{x}_0 + \boldsymbol{u}) - \log q(\boldsymbol{u}|\boldsymbol{x}_0)]\,.$$

A simple way for $q(\boldsymbol{u}|\boldsymbol{x}_0)$ is uniform dequantization, where we set $q(\boldsymbol{u}|\boldsymbol{x}_0) = \mathcal{U}(-\frac{1}{256}, \frac{1}{256})$.

3. Diffusion ODEs with Truncated-Normal Dequantization

In this section, we discuss the challenges of training diffusion ODEs with dequantization and propose a training-free dequantization method for diffusion ODEs.

3.1. Challenges for Diffusion ODEs with Dequantization

We first discuss the challenges for diffusion ODEs with dequantization in this section.

Truncation introduces an additional gap. Theoretically, we want to train diffusion ODEs by minimizing $D_{\text{KL}}(q_0 \parallel p_0)$ and use $p_0(\boldsymbol{x}_0)$ for the continuous model distribution. However, as $\sigma_0 = 0$, we have $\gamma_0 = -\infty$. Due to this, it is shown in previous work (Song et al., 2021c; Kim et al., 2022) that there are numerical issues near $t = 0$ for both training and sampling, so we cannot directly compute the model distribution p_0 at time 0. In practice, a common solution is to choose a small starting time $\epsilon > 0$ for improving numerical stability. The training objective then becomes minimizing $D_{\text{KL}}(q_\epsilon \parallel p_\epsilon)$, which is equivalent to

$$\max_\theta \mathbb{E}_{q_0(\boldsymbol{x}_0)q_{0\epsilon}(\boldsymbol{x}_\epsilon|\boldsymbol{x}_0)}[\log p_\epsilon(\boldsymbol{x}_\epsilon)], \tag{8}$$

and $\mathbb{E}_{q_0(\boldsymbol{x}_0)} \log p_\epsilon(\boldsymbol{x}_0)$ is directly used to evaluate the data likelihood. However, as $p_\epsilon \neq p_0$, such a method will introduce an additional gap due to the mismatch between training ($\mathbb{E}_{q_\epsilon(\boldsymbol{x}_\epsilon)}[\log p_\epsilon(\boldsymbol{x}_\epsilon)]$) and testing ($\mathbb{E}_{q_0(\boldsymbol{x}_0)}[\log p_\epsilon(\boldsymbol{x}_0)]$), which may degrade the likelihood evaluation performance.

Uniform dequantization causes a train-test mismatch. After choosing ϵ, the continuous model distribution is defined by $p_{\text{model}}(\boldsymbol{x}) \coloneqq p_\epsilon(\boldsymbol{x})$. Let $q(\boldsymbol{u}|\boldsymbol{x}_0)$ be a dequantization distribution with support over $\boldsymbol{u} \in [-\frac{1}{256}, \frac{1}{256})^d$. The variational lower bound for the discrete model density $P_0(\boldsymbol{x}_0)$ is:

$$\begin{aligned}\mathbb{E}_{q_0(\boldsymbol{x}_0)}[\log P_0(\boldsymbol{x}_0)] \geq\ & \mathbb{E}_{q_0(\boldsymbol{x}_0)q(\boldsymbol{u}|\boldsymbol{x}_0)}\left[\log p_\epsilon(\boldsymbol{x}_0 + \boldsymbol{u})\right] \\ & - \mathbb{E}_{q_0(\boldsymbol{x}_0)q(\boldsymbol{u}|\boldsymbol{x}_0)}\left[\log q(\boldsymbol{u}|\boldsymbol{x}_0)\right].\end{aligned}$$

One widely-used choice for $q(\boldsymbol{u}|\boldsymbol{x}_0)$ is uniform distribution (uniform dequantization). However, this leads to a training-evaluation gap: for training, we fit p_ϵ to the distribution $q_\epsilon(\boldsymbol{x}_\epsilon)$, which is a Gaussian distribution near each discrete data point $\boldsymbol{x}_0$ because $\boldsymbol{x}_\epsilon = \alpha_\epsilon \boldsymbol{x}_0 + \sigma_\epsilon \boldsymbol{\epsilon}$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$; while for evaluation, we test p_ϵ on uniform dequantized data $\boldsymbol{x}_0 + \boldsymbol{u}$. Such a gap will also degrade the likelihood evaluation performance and is not well-studied.

In addition, another way for dequantization is to train a variational dequantization model $q_\phi(\boldsymbol{u}|\boldsymbol{x}_0)$ (Ho et al., 2019; Song et al., 2021b) but it will need additional costs and is hard to train (Kim et al., 2022).

3.2. Training-Free Dequantization by Truncated Normal

In this section, we show that there exists a training-free dequantization distribution that fits diffusion ODEs well.

As discussed in Sec. 3.1, the gap between training and testing of diffusion ODEs is due to the difference between the training input $\boldsymbol{x}_\epsilon = \alpha_\epsilon \boldsymbol{x}_0 + \sigma_\epsilon \boldsymbol{\epsilon}$ (where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$) and the testing input $\boldsymbol{x}_0 + \boldsymbol{u}$. To fill such a gap, we can choose

a dequantization distribution $q(\boldsymbol{u}|\boldsymbol{x}_0)$ which satisfies

$$\boldsymbol{x}_0 + \boldsymbol{u} \approx \alpha_\epsilon \boldsymbol{x}_0 + \sigma_\epsilon \boldsymbol{\epsilon}, \quad \boldsymbol{u} \in \left[-\frac{1}{256}, \frac{1}{256}\right)^d. \tag{9}$$

For small enough ϵ, we have $\alpha_\epsilon \approx 1$, then Eqn. (9) becomes $\boldsymbol{u} \approx \frac{\sigma_\epsilon}{\alpha_\epsilon}\boldsymbol{\epsilon}$. We also need to ensure the support of $q(\boldsymbol{u}|\boldsymbol{x}_0)$ is $[-\frac{1}{256}, \frac{1}{256})^d$, i.e. the random variable $\frac{\sigma_\epsilon}{\alpha_\epsilon}\boldsymbol{\epsilon}$ is approximately within $[-\frac{1}{256}, \frac{1}{256})^d$. To this end, we choose the variational dequantization distribution by a truncated normal distribution as follows:

$$q(\boldsymbol{u}|\boldsymbol{x}_0) = \mathcal{TN}(\boldsymbol{0}, \frac{\sigma_\epsilon^2}{\alpha_\epsilon^2}\boldsymbol{I}, -\frac{1}{256}, \frac{1}{256}) \tag{10}$$

where $\mathcal{TN}(\boldsymbol{x}|\boldsymbol{\mu}, \sigma^2\boldsymbol{I}, a, b)$ is a truncated-normal distribution with mean $\boldsymbol{\mu}$, covariance $\sigma^2\boldsymbol{I}$, and bounds $[a, b]$ in each dimension. Moreover, such truncated-normal dequantization provides a guideline for choosing the start time ϵ: To avoid large deviation from the truncation by $\frac{1}{256}$, we need to ensure that $\frac{\alpha_\epsilon}{\sigma_\epsilon}\boldsymbol{u} \approx \boldsymbol{\epsilon}$ in most cases. We leverage the 3-σ principle for standard normal distribution and let ϵ to satisfy $\frac{\alpha_\epsilon}{\sigma_\epsilon}\boldsymbol{u} \in [-3, 3]^d$. As $\boldsymbol{u} \in [-\frac{1}{256}, \frac{1}{256})$, the critical start time ϵ satisfies that the negative log-SNR $\gamma_\epsilon = -\log\frac{\alpha_\epsilon^2}{\sigma_\epsilon^2} \approx -13.3$. Surprisingly, such choice of γ_ϵ is exactly the same as the γ_{min} in Kingma et al. (2021) which instead is obtained by training. Such dequantization distribution can ensure the conditions in Eqn. (9) and we validate in Sec. 6 that such dequantization can provide a tighter variational bound yet with no additional training costs. We summarize the likelihood evaluation by such dequantization distribution in the following theorem.

Theorem 3.1 (Variational Bound under Truncated-Normal Dequantization)**.** *Suppose we use the truncated-normal dequantization in Eqn. (10), then the discrete model distribution has the following variational bound:*

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{q(\hat{\boldsymbol{\epsilon}})}\left[\log p_\epsilon(\hat{\boldsymbol{x}}_\epsilon)\right] + \frac{d}{2}(1 + \log(2\pi\sigma_\epsilon^2)) + d\log Z - d\frac{\tau}{\sqrt{2\pi}Z}\exp(-\frac{1}{2}\tau^2)$$

where

$$\tau = \frac{\alpha_\epsilon}{256\sigma_\epsilon}, \quad Z = erf\left(\frac{\tau}{\sqrt{2}}\right)$$
$$\hat{\boldsymbol{x}}_\epsilon = \alpha_\epsilon\boldsymbol{x}_0 + \sigma_\epsilon\hat{\boldsymbol{\epsilon}}, \quad \hat{\boldsymbol{\epsilon}} \sim \mathcal{TN}\left(\hat{\boldsymbol{\epsilon}}\,|\boldsymbol{0}, \boldsymbol{I}, -\tau, \tau\right).$$

Besides, we also have the following importance weighted likelihood estimator by using K i.i.d. samples by using Jensen's inequality as in Burda et al. (2015). As K increases, the estimator gives a tighter bound, which enables more precise likelihood estimation.

Corollary 3.2 (Importance Weighted Variational Bound under Truncated-Normal Dequantization)**.** *Suppose we use the truncated-normal dequantization in Eqn. (10), then the discrete model distribution has the following importance weighted variational bound:*

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{\prod_{i=1}^K q(\hat{\boldsymbol{\epsilon}}^{(i)})}\left[\log\left(\frac{1}{K}\sum_{i=1}^K \frac{p_\epsilon(\hat{\boldsymbol{x}}_\epsilon^{(i)})}{q(\hat{\boldsymbol{\epsilon}}^{(i)})}\right)\right] + d\log\sigma_\epsilon$$

where

$$\hat{\boldsymbol{x}}_\epsilon^{(i)} = \alpha_\epsilon\boldsymbol{x}_0 + \sigma_\epsilon\hat{\boldsymbol{\epsilon}}^{(i)}, \quad \hat{\boldsymbol{\epsilon}}^{(i)} \sim \mathcal{TN}\left(\hat{\boldsymbol{\epsilon}}^{(i)}\,|\boldsymbol{0}, \boldsymbol{I}, -\tau, \tau\right)$$
$$q(\hat{\boldsymbol{\epsilon}}) = \frac{1}{(2\pi Z^2)^{\frac{d}{2}}}\exp(-\frac{1}{2}\|\hat{\boldsymbol{\epsilon}}\|_2^2), \quad Z = erf\left(\frac{\tau}{\sqrt{2}}\right).$$

Remark 3.3. Another way to bridge the discrete-continuous gap is *variational perspective*. We can view the process from discrete $\boldsymbol{x}_0$ to continuous $\boldsymbol{x}_\epsilon$ as a variational autoencoder, where the prior $p_\epsilon(\boldsymbol{x}_\epsilon)$ is modeled by diffusion ODE. The dequantization and variational perspectives of diffusion ODEs have a close relationship both theoretically and empirically, and we detailedly discuss them in Appendix A.

4. Practical Techniques for Improving the Likelihood of Diffusion ODEs

In this section, we propose some practical techniques for improving the likelihood of diffusion ODEs, including parameterization, a high-order training objective, and variance reduction by importance sampling. For simplicity, we denote $\dot{f}_x = \frac{\mathrm{d}f(x)}{\mathrm{d}x}$ for any scalar function $f(x)$.

4.1. Velocity Parameterization

While the score matching objective $\mathcal{J}_{\text{SM}}(\theta)$ only depends on the noise schedule, the training process is affected by many aspects such as network parameterization (Song et al., 2021c; Karras et al., 2022). For example, the noise predictor $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ is widely used to replace the score predictor $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$, since the noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ has unit variance and is easier to fit, while $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) = -\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)/\sigma_t$ is pathological and explosive near $t = 0$ (Song et al., 2021c).

In this work, we consider another network parameterization which is to directly predict the drift of the diffusion ODE. The parameterized model is defined by

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{v}_\theta(\boldsymbol{x}_t, t) \coloneqq f(t)\boldsymbol{x}_t - \frac{1}{2}g^2(t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \tag{11}$$

By rewriting the (first-order) score matching objective in Eqn. (5), $\mathcal{J}_{\text{SM}}(\theta)$ is equivalent to:

$$\mathcal{J}_{\text{FM}}(\theta) \coloneqq \int_0^T \frac{2}{g^2(t)}\mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v}\|_2^2\right]\mathrm{d}t, \tag{12}$$

where $\boldsymbol{v} = \dot{\alpha}_t\boldsymbol{x}_0 + \dot{\sigma}_t\boldsymbol{\epsilon}$ is the velocity to predict. Given unlimited model capacity, the optimal $\boldsymbol{v}^*$ is

$$\boldsymbol{v}^*(\boldsymbol{x}_t, t) = f(t)\boldsymbol{x}_t - \frac{1}{2}g^2(t)\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t), \tag{13}$$

which is the drift of probability flow ODE in Eqn. (2).

We give an intuitive explanation for $\mathcal{J}_{\text{FM}}$ in Appendix D that the prediction target $\boldsymbol{v}$ is the tangent (velocity) of the diffusion path, and we name $\boldsymbol{v}_\theta$ as *velocity parameterization*. Besides, we show it empirically alleviates the *imbalance problem* in noise prediction.

In addition, we prove the equivalence between different predictors and different matching objectives for general noise schedules in Appendix B. We also show in Appendix E that the flow matching method Lipman et al. (2022); Albergo & Vanden-Eijnden (2022); Liu et al. (2022b) and related techniques for improving the sample quality of diffusion models in Karras et al. (2022); Salimans & Ho (2022); Ho et al. (2022) can all be reformulated in velocity parameterization. To be consistent, we still call $\mathcal{J}_{\text{FM}}$ as flow matching. It's an extended version of Lipman et al. (2022) with likelihood weighting and several practical modifications as detailed in Section 4.3.

4.2. Error-bounded Second-Order Flow Matching

According to Chen et al. (2018a), the ODE likelihood of Eqn. (11) can be evaluated by solving the following differential equation from ϵ to T:

$$\frac{\mathrm{d}\log p_t(\boldsymbol{x}_t)}{\mathrm{d}t} = -\mathrm{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{v}_\theta(\boldsymbol{x}_t, t)). \tag{14}$$

As $\mathcal{J}_{\text{FM}}$ in Eqn. (12) can only restrict the distance between $\boldsymbol{v}_\theta$ and $\boldsymbol{v}^*$, but not the divergence $\mathrm{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{v}_\theta)$ and $\mathrm{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{v}^*)$. The precision and smoothness of the trace $\mathrm{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{v}_\theta(\boldsymbol{x}_t, t))$ affects the likelihood performance and the number of function evaluations for sampling. For simulation-free training of $\mathrm{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{v}_\theta(\boldsymbol{x}_t, t))$, we propose an error-bounded trace of second-order flow matching, where the second-order error is bounded by the proposed objective and first-order error.

Theorem 4.1. *(Error-Bounded Trace of Second-Order Flow Matching) Suppose we have a first-order velocity estimator $\hat{\boldsymbol{v}}_1(\boldsymbol{x}_t, t)$, we can learn a second-order trace velocity model $\boldsymbol{v}_2^{trace}(\cdot, t; \theta) : \mathbb{R}^d \to \mathbb{R}$ which minimizes*

$$\mathbb{E}_{q_t(\boldsymbol{x}_t)}\left[\left|\boldsymbol{v}_2^{trace}(\boldsymbol{x}_t, t; \theta) - \mathrm{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{v}^*(\boldsymbol{x}_t, t))\right|^2\right],$$

by optimizing

$$\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\left|\boldsymbol{v}_2^{trace}(\boldsymbol{x}_t, t; \theta) - \frac{\dot\sigma_t}{\sigma_t}d + \boldsymbol{\ell}_1\right|^2\right]. \tag{15}$$

where

$$\boldsymbol{\ell}_1(\boldsymbol{\epsilon}, \boldsymbol{x}_0, t) \coloneqq \frac{2}{g^2(t)}\|\hat{\boldsymbol{v}}_1(\boldsymbol{x}_t, t) - \boldsymbol{v}\|_2^2,$$
$$\boldsymbol{x}_t = \alpha_t\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}, \quad \boldsymbol{v} = \dot\alpha_t\boldsymbol{x}_0 + \dot\sigma_t\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}).$$

Moreover, denote the first-order flow matching error as $\delta_1(\boldsymbol{x}_t, t) \coloneqq \|\hat{\boldsymbol{v}}_1(\boldsymbol{x}_t, t) - \boldsymbol{v}^*(\boldsymbol{x}_t, t)\|_2$, then $\forall \boldsymbol{x}_t, \theta$, the estimation error for $\boldsymbol{v}_2^{trace}(\boldsymbol{x}_t, t; \theta)$ can be bounded by:

$$\begin{aligned}&|\boldsymbol{v}_2^{trace}(\boldsymbol{x}_t, t; \theta) - \mathrm{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{v}^*(\boldsymbol{x}_t, t))| \\ \le\ &|\boldsymbol{v}_2^{trace}(\boldsymbol{x}_t, t; \theta) - \boldsymbol{v}_2^{trace}(\boldsymbol{x}_t, t; \theta^*)| + \frac{2}{g^2(t)}\delta_1^2(\boldsymbol{x}_t, t).\end{aligned}$$

The proof is provided in Appendix F. In practice, we choose $\boldsymbol{v}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta) = \mathrm{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{v}_\theta(\boldsymbol{x}_t, t))$ for self-regularizing. As for scalability, we use Hutchinson's trace estimator (Hutchinson, 1990) to unbiasedly estimate the trace, and use forward-mode automatic differentiation to compute Jacobian-vector product (Lu et al., 2022a).

4.3. Timing by Log-SNR and Normalizing Velocity

In practice, we make two modifications to improve the performance. First, we use negative log-SNR γ_t to time the diffusion process. Still, we parameterize $\boldsymbol{v}_\theta(\boldsymbol{x}_\gamma, \gamma)$ to predict the drift of the γ timed diffusion ODE i.e. $\frac{\mathrm{d}\boldsymbol{x}_\gamma}{\mathrm{d}\gamma} = \boldsymbol{v}_\theta(\boldsymbol{x}_\gamma, \gamma)$, so the corresponding predictor $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{v}_\theta(\boldsymbol{x}_\gamma, \gamma)\frac{\mathrm{d}\gamma}{\mathrm{d}t}$. Second, the velocity of the diffusion path $\boldsymbol{v} = \dot\alpha_t\boldsymbol{x}_0 + \dot\sigma_t\boldsymbol{\epsilon}$ may have different scales at different t, so we propose to predict the normalized velocity $\tilde{\boldsymbol{v}} = \boldsymbol{v}/\sqrt{\dot\alpha_t^2 + \dot\sigma_t^2}$, with the parameterized network $\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{v}_\theta(\boldsymbol{x}_t, t)/\sqrt{\dot\alpha_t^2 + \dot\sigma_t^2}$, which is equal to $\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma, \gamma) = \boldsymbol{v}_\theta(\boldsymbol{x}_\gamma, \gamma)/\sqrt{\dot\alpha_\gamma^2 + \dot\sigma_\gamma^2}$. The objective in Eqn. (12) reduces to

$$\mathcal{J}_{\text{FM}}(\theta) = \int_{\gamma_0}^{\gamma_T} 2\frac{\dot\alpha_\gamma^2 + \dot\sigma_\gamma^2}{\sigma_\gamma^2}\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\|\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma, \gamma) - \tilde{\boldsymbol{v}}\|_2^2\mathrm{d}\gamma.$$

And the corresponding second-order objective:

$$\begin{aligned}\mathcal{J}_{\text{FM,tr}} = \int_{\gamma_0}^{\gamma_T} 2\frac{\dot\alpha_\gamma^2 + \dot\sigma_\gamma^2}{\sigma_\gamma^2}\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\Bigg(&\sigma_\gamma\mathrm{tr}(\nabla\tilde{\boldsymbol{v}}_\theta) - \frac{\dot\sigma_\gamma}{\sqrt{\dot\alpha_\gamma^2 + \dot\sigma_\gamma^2}}d \\ &+ \frac{2\sqrt{\dot\alpha_\gamma^2 + \dot\sigma_\gamma^2}}{\sigma_\gamma}\|\tilde{\boldsymbol{v}}_\theta^{(s)}(\boldsymbol{x}_\gamma, \gamma) - \tilde{\boldsymbol{v}}\|_2^2\Bigg)^2\mathrm{d}\gamma\end{aligned} \tag{16}$$

where $\tilde{\boldsymbol{v}}_\theta^{(s)}$ is the stop-gradient version of $\tilde{\boldsymbol{v}}_\theta$, since we only use the parameterized first-order velocity predictor as an estimator. Our final formulation of parameterized diffusion ODE is

$$\frac{\mathrm{d}\boldsymbol{x}_\gamma}{\mathrm{d}\gamma} = \sqrt{\dot\alpha_\gamma^2 + \dot\sigma_\gamma^2}\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma, \gamma) \tag{17}$$

4.4. Variance Reduction with Importance Sampling

The flow matching is conducted for all γ in $[\gamma_0, \gamma_T]$ through an integral. In practice, the evaluation of the integral is time-consuming, and Monte-Carlo methods are used to unbiasedly estimate the objective by uniformly sampling γ. In

this case, the variance of the Monte-Carlo estimator affects the optimization process. Thus, a continuous importance distribution $p(\gamma)$ can be proposed for variance reduction. Denote $\mathcal{L}_\theta(\boldsymbol{x}_0, \boldsymbol{\epsilon}, \gamma,) = 2\frac{\dot{\alpha}_\gamma^2+\dot{\sigma}_\gamma^2}{\sigma_\gamma^2}\|\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma, \gamma) - \tilde{\boldsymbol{v}}\|_2^2$, then

$$\mathcal{J}_{\text{FM}}(\theta) = \mathbb{E}_{\gamma\sim p(\gamma)}\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\frac{\mathcal{L}_\theta(\boldsymbol{x}_0, \boldsymbol{\epsilon}, \gamma)}{p(\gamma)}\right] \tag{18}$$

We propose to use two types of importance sampling (IS), and empirically compare them for faster convergence.

Designed IS Intuitively, we can chooise $p(\gamma) \propto \frac{\dot{\alpha}_\gamma^2+\dot{\sigma}_\gamma^2}{\sigma_\gamma^2}$. This way, the coefficients of $\|\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma, \gamma) - \tilde{\boldsymbol{v}}\|_2^2$ is a time-invariant constant, and the velocity matching error is not amplified or shrank at any γ. This is similar to the IS in Song et al. (2021b), where the $g^2(t)/\sigma_t^2$ weighting before the noise matching error $\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\epsilon}\|_2^2$ is cancelled, and it corresponds to uniform γ under our parameterization.

For noise schedules used in this paper, we can obtain closed-form sampling procedures using *inverse transform sampling*, see Appendix C.

Learned IS The variance of the Monte-Carlo estimator depends on the learned network $\tilde{\boldsymbol{v}}_\theta$. To minimize the variance, we can parameterize the IS with another network and treat the variance as an objective. Actually, learning $p(\gamma)$ is equivalent to learning a monotone mapping $\gamma(t) : [0, 1] \to [\gamma_0, \gamma_T]$, which is inverse cumulative distribution function of $p(\gamma)$. We can uniformly sample t, and regard the IS as change-of-variable from γ to t.

$$\mathcal{J}_{\text{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\gamma'(t)\mathcal{L}_\theta(\boldsymbol{x}_0, \boldsymbol{\epsilon}, \gamma(t))\right] \tag{19}$$

Suppose we parameterize $\gamma(t)$ with η. Denote $\mathcal{L}_{\theta,\eta}(\boldsymbol{x}_0, \boldsymbol{\epsilon}, t) = \gamma'_\eta(t)\mathcal{L}_\theta(\boldsymbol{x}_0, \boldsymbol{\epsilon}, \gamma_\eta(t))$, which is a Monte-Carlo estimator of $\mathcal{J}_{\text{FM}}(\theta)$. Since its variance $\text{Var}_{t,\boldsymbol{\epsilon},\boldsymbol{x}_0}[\mathcal{L}_{\theta,\eta}(\boldsymbol{x}_0, \boldsymbol{\epsilon}, t)] = \mathbb{E}_{t,\boldsymbol{\epsilon},\boldsymbol{x}_0}[\mathcal{L}^2_{\theta,\eta}(\boldsymbol{x}_0, \boldsymbol{\epsilon}, t)] - \mathcal{J}^2_{\text{FM}}(\theta)$ and $\mathcal{J}_{\text{FM}}(\theta)$ is invariant to $\gamma_\eta(t)$, we can minimize $\mathbb{E}_{t,\boldsymbol{\epsilon},\boldsymbol{x}_0}[\mathcal{L}^2_{\theta,\eta}(\boldsymbol{x}_0, \boldsymbol{\epsilon}, t)]$ for variance reduction.

While this approach seeks the optimal IS, they cause extra overhead by introducing an IS network, requiring complex gradient operation or additional training steps. Thus, we only use it as a reference to test the optimality of our designed IS. We simplify the variance reduction in Kingma et al. (2021), and propose an *adaptive IS* algorithm, which is detailed in Appendix H. Empirically, we show that designed IS is a more preferred approach since it is training-free and achieves a similar convergence speed to learned IS.

5. Related Work

Diffusion models, also known as score-based generative models (SGMs), have achieved state-of-the-art sample quality and likelihood (Dhariwal & Nichol, 2021; Karras et al., 2022; Kingma et al., 2021) among deep generative models, yielding extensive downstream applications such as speech and singing synthesis (Chen et al., 2021; Liu et al., 2022a), conditional image generation (Ramesh et al., 2022; Rombach et al., 2022), guided image editing (Meng et al., 2022; Nichol et al., 2022), unpaired image-to-image translation (Zhao et al., 2022) and inverse problem solving (Chung et al., 2022; Kawar et al., 2022).

Diffusion ODEs are special formulations of neural ODEs and can be viewed as continuous normalizing flows (Chen et al., 2018a). Training of diffusion ODEs can be categorized into simulation-based and simulation-free methods. The former utilizes the exact likelihood evaluation formula of ODE (Chen et al., 2018a), which leads to a maximum likelihood training procedure (Grathwohl et al., 2019). However, it involves expensive ODE simulations for forward and backward propagation and may result in unnecessary complex dynamics (Finlay et al., 2020) since it only cares about the model distribution at $t = 0$. The latter trains neural ODEs by matching their trajectories to a predefined path, such as the diffusion process. This approach is proposed in Song et al. (2021c), and extended in Lu et al. (2022a); Lipman et al. (2022); Albergo & Vanden-Eijnden (2022); Liu et al. (2022b). We propose velocity parameterization which is an extension of Lipman et al. (2022) with practical modifications and claim that the paths used in Lipman et al. (2022); Albergo & Vanden-Eijnden (2022); Liu et al. (2022b) are special cases of noise schedule. Aiming at maximum likelihood training, we also get inspiration from Lu et al. (2022a). We additionally apply likelihood weighting and propose to finetune the model with high-order flow matching.

Variance reduction techniques are commonly used for training diffusion models. Nichol & Dhariwal (2021) proposes an importance sampling (IS) for discrete-time diffusion models by maintaining the historical losses at each time step and building the proposal distribution based on them. Song et al. (2021b) designs an IS to cancel out the weighting before the noise matching loss. Kingma et al. (2021) proposes a variance reduction method that is equivalent to learning a parameterized IS. We simply their procedure and propose an adaptive IS scheme for ablation. By empirically comparing different IS methods, we find a designed and analytical IS distribution that achieves a good performance-efficiency trade-off.

6. Experiments

In this section, we present our training procedure and experiment settings, and our ablation studies to demonstrate how our techniques improve the likelihood of diffusion ODEs.

We implement our methods based on the open-source code-

base of Kingma et al. (2021) implemented with JAX Bradbury et al. (2018), and use similar network and hyperparameter settings. We first train the model by optimizing our first-order flow matching objective $\min_\theta \mathcal{J}_{\text{FM}}(\theta)$ for enough iterations, so that the first-order velocity prediction has little error. Then, we finetune the pretrained first-order model using a mixture of first-order and second-order flow matching objectives $\min_\theta \mathcal{J}_{\text{FM}}(\theta) + \lambda \mathcal{J}_{\text{FM,tr}}(\theta)$. The finetune process converges in much fewer iterations than pretraining. Finally, we evaluate the likelihood on the test set using the variational bound under our proposed truncated-normal dequantization. The detailed training configurations are provided in Appendix I.

Our training and evaluation procedure is feasible for any noise schedule $\alpha_\gamma, \sigma_\gamma$. We choose two special noise schedules:

Variance Preserving (VP) $\alpha_\gamma^2 + \sigma_\gamma^2 = 1$. This schedule is widely used in diffusion models, which yields a process with a fixed variance of one when the initial distribution has unit variance.

Straight Path (SP) $\alpha_\gamma + \sigma_\gamma = 1$. This schedule is used in Lipman et al. (2022); Albergo & Vanden-Eijnden (2022); Liu et al. (2022b), where they call it OT path and claim it leads to better dynamics since the pairwise diffusion paths are straight lines. We simply regard it as a special kind of noise schedule.

Under these two schedules, $\alpha_\gamma, \sigma_\gamma$ are uniquely determined by γ, and we do not have any extra hyperparameters. They also have corresponding objectives and designed IS, which can be expressed in closed form (see Appendix C for details). We train our i-DODE on CIFAR-10 (Krizhevsky et al., 2009) and ImageNet-32[1] (Deng et al., 2009), which are two popular benchmarks for generative modeling and density estimation.

6.1. Likelihood and Samples

Table 1 shows our experiment results on CIFAR-10 and ImageNet-32 datasets. Our models are pretrained with velocity parameterization, designed IS, and finetuned with second-order flow matching. We report the likelihood values using our truncated-normal dequantization with the importance weighted estimator under $K = 20$. To compute the FID values, we apply an adaptive-step ODE solver to draw samples from the diffusion ODEs. We also report the NFE during the sampling process, which reflects the smoothness of the dynamics.

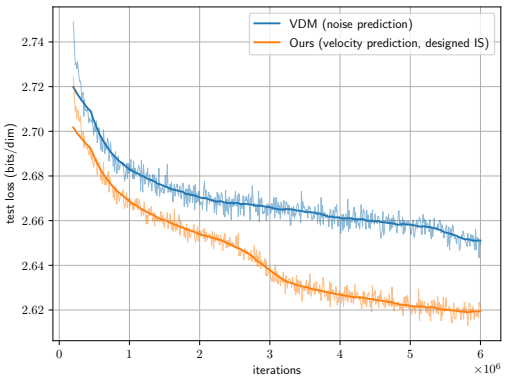


Figure 1. Test loss curve in the pretraining phase, compared to VDM (Kingma et al., 2021). We compute the loss on the test set by the SDE likelihood bound in Kingma et al. (2021).

Combining our training techniques and dequantization, we exceed the likelihood of previous ODEs, especially by a large margin on CIFAR-10. In Figure 1, we compare our pretraining phase to VDM (Kingma et al., 2021), which indicates that our techniques achieve 2x∼3x times of previous convergence speed. We do not observe the superiority of SP to VP such as lower FID and NFE as in Lipman et al. (2022). We suspect it may result from maximum likelihood training, which put more emphasis on the high log-SNR region. More theoretical comparisons with Lipman et al. (2022) are given in Appendix E.2.

Randomly generated samples from our models are provided in Appendix J. Since we use network architecture and techniques targeted at the likelihood, our FID is worse than the state-of-the-art, which can be improved by designing time weighting to emphasize the training at small log-SNR levels (Kingma et al., 2021) or using high-quality sampling algorithms such as PC sampler (Song et al., 2021c).

6.2. Ablations

Due to the expensive time cost of pretraining, we only conduct ablation studies on CIFAR-10 under the VP schedule. First, we test our techniques for pretraining when training from scratch. We plot the training curves with noise predictor (Kingma et al., 2021) and velocity predictor, then further implement our IS strategies (Figure 2). We find that velocity parameterization and IS both accelerate the training process, while designed IS performs slightly worse than adaptive IS. Considering the extra time cost for learning the IS network, we conclude that designed IS is a better choice for large-scale pretraining. Then we visualize different IS by plotting the mapping from uniform t to importance sampled

[1]There are two different versions of ImageNet32 and ImageNet64 datasets. For fair comparisons, we use both versions of ImageNet32, one is downloaded from `https://image-net.org/data/downsample/Imagenet32_train.zip`, following Lipman et al. (2022), and the other is downloaded from `http://image-net.org/small/train_32x32.tar` (old version, no longer available), following Song et al. (2021b) and Kingma et al. (2021). The former dataset applies anti-aliasing and is easier for maximum likelihood training.

Table 1. Negative log-likelihood (NLL) in bits/dim (BPD), sample quality (FID scores) and number of function evaluations (NFE) on CIFAR-10 and ImageNet 32x32. For fair comparisons, we list NLL results of previous ODEs without variational dequantization or data augmentation, and FID/NFE results obtained by adaptive-step ODE solver. Results with "/" means they are not reported in the original papers. †For VDM, since they have no ODE formulation, the FID score is obtained by 1000 step discretization of their SDE. We report their corresponding ODE result in the ablation study. *Corresponding to the old version ImageNet-32 dataset.

Model	CIFAR-10			ImageNet-32		
	NLL ↓	FID ↓	NFE ↓	NLL ↓	FID ↓	NFE ↓
VDM (Kingma et al., 2021)	2.65	7.60†	1000	3.72*	/	/
(Previous ODE)						
FFJORD (Grathwohl et al., 2019)	3.40	/	/	/	/	/
ScoreSDE (Song et al., 2021c)	2.99	2.92	/	/	/	/
ScoreFlow (Song et al., 2021b)	2.90	5.40	/	3.82*	10.18*	/
Soft Truncation (Kim et al., 2022)	3.01	3.96	/	3.90*	8.42*	/
Flow Matching (Lipman et al., 2022)	2.99	6.35	142	3.53	5.31	122
Stochastic Interp.(Albergo & Vanden-Eijnden, 2022)	2.99	10.27	/	3.48	8.49	/
i-DODE (SP) (**ours**)	**2.56**	11.20	162	3.44/**3.69***	10.31	138
i-DODE (VP) (**ours**)	2.57	10.74	126	**3.43**/3.70*	9.09	152

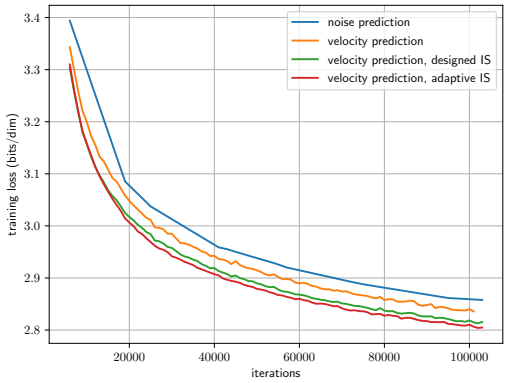
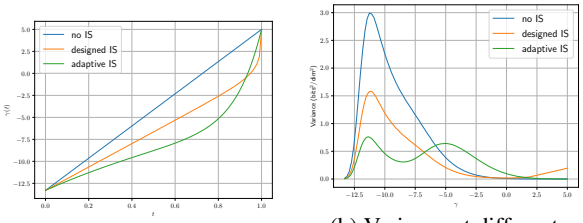


Figure 2. Training curve from scratch for ablation. We compute the loss on the training set by the SDE likelihood bound in Kingma et al. (2021).



(a) $\gamma(t)$ (b) Variance at different log-SNR levels.

Figure 3. Visualization of importance sampling: (a) The inverse cumulative distribution function $\gamma(t)$ of the proposal distribution $p(\gamma)$, which maps uniform t to importance sampled γ (b) The variance of Monte-Carlo estimator Var $[\gamma'(t)\mathcal{L}_\theta(\boldsymbol{x}_0, \boldsymbol{\epsilon}, \gamma(t))]$ at different noise levels, estimated using 32 data samples $\boldsymbol{x}_0$ and 100 noise samples $\boldsymbol{\epsilon}$. The peak variance is achieved around $\gamma = -11.2$.

γ, as well as the variance at different noise levels on the pretrained model (Figure 3). We show that the IS reduces the variance by sampling more in high log-SNR regions.

Table 2. Ablation study when converged. We report negative log-likelihood (NLL) in bits/dim (BPD), sample quality (FID scores), and number of function evaluations (NFE) after our pretraining and finetuning phase. We evaluate NLL by uniform (U) and truncated-normal (TN) dequantization without importance weight. We retrain VDM and evaluate its ODE form.

Model	NLL (U)	NLL (TN)	FID	NFE
VDM (Kingma et al., 2021)	2.78	2.64	8.65	213
Pretrain (**ours**)	2.75	2.61	10.66	248
+ Finetune (**ours**)	2.74	**2.60**	10.74	**126**

Next, we test our pretraining, finetuning and evaluation on the converged model (Table 2). As stated before, our pretraining has faster loss descent and converges to a higher likelihood than VDM. Based on it, our finetuning slightly improves the ODE likelihood and smooths the flow, leading to much less NFE when sampling. Our truncated-normal dequantization is also a key factor for precise likelihood computing, which surpasses previous uniform dequantization by a large margin.

In agreement with Song et al. (2021b), our improvements in likelihood lead to slightly worse FIDs. We also argue that the degeneration is small in terms of visual quality. We provide additional samples in the Appendix J for comparison.

7. Conclusion

We propose improved techniques for simulation-free maximum likelihood training and likelihood evaluation of diffusion ODEs. Our training stage involves improved pretraining and additional finetuning, which results in fast convergence, high likelihood and smooth trajectory. We improve the likelihood evaluation with novel truncated-normal dequantization, which is training-free and tailor-made for diffusion ODEs. Empirically, we achieve state-of-the-art likelihood on image datasets without variational dequanti-

zation or data augmentation and make a breakthrough on CIFAR-10 compared to previous ODEs. Due to resource limitations, we didn't explore tuning of hyperparameters and network architectures, which are left for future work.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (2020AAA0106302); NSF of China Projects (Nos. 62061136001, 61620106010, 62076145, U19B2034, U1811461, U19A2081, 6197222, 62106120, 62076145); a grant from Tsinghua Institute for Guo Qiang; the High Performance Computing Center, Tsinghua University. J.Z was also supported by the New Cornerstone Science Foundation through the XPLORER PRIZE.

References

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

Anderson, B. D. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., et al. Jax: composable transformations of python+ numpy programs. *Version 0.2*, 5:14–24, 2018.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 6572–6583, 2018a.

Chen, X., Mishra, N., Rohaninejad, M., and Abbeel, P. Pixelsnail: An improved autoregressive generative model. In *International Conference on Machine Learning*, pp. 864–872. PMLR, 2018b.

Chen, Z., Yeo, C. K., Lee, B. S., and Lau, C. T. Autoencoder-based network anomaly detection. In *2018 Wireless telecommunications symposium (WTS)*, pp. 1–5. IEEE, 2018c.

Choi, K., Meng, C., Song, Y., and Ermon, S. Density ratio estimation via infinitesimal classification. In *International Conference on Artificial Intelligence and Statistics*, pp. 2552–2573. PMLR, 2022.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

Dhariwal, P. and Nichol, A. Q. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

Dias, M. L., Mattos, C. L. C., da Silva, T. L., de Macedo, J. A. F., and Silva, W. C. Anomaly detection in trajectory data with normalizing flows. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

Dormand, J. R. and Prince, P. J. A family of embedded Runge-Kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

Finlay, C., Jacobsen, J.-H., Nurbekyan, L., and Oberman, A. How to train your neural ode: the world of jacobian and kinetic regularization. In *International conference on machine learning*, pp. 3154–3164. PMLR, 2020.

Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., and Duvenaud, D. Ffjord: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019.

Helminger, L., Djelouah, A., Gross, M., and Schroers, C. Lossy image compression with normalizing flows. *arXiv preprint arXiv:2008.10486*, 2020.

Ho, J., Chen, X., Srinivas, A., Duan, Y., and Abbeel, P. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pp. 2722–2730. PMLR, 2019.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J.,

et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Ho, Y.-H., Chan, C.-C., Peng, W.-H., Hang, H.-M., and Domański, M. Anfic: Image compression using augmented normalizing flows. *IEEE Open Journal of Circuits and Systems*, 2:613–626, 2021.

Huang, C.-W., Lim, J. H., and Courville, A. A variational perspective on diffusion-based generative models and score matching. In *Advances in Neural Information Processing Systems*, 2021.

Hutchinson, M. F. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450, 1990.

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

Kawar, B., Elad, M., Ermon, S., and Song, J. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, 2022.

Kim, D., Shin, S., Song, K., Kang, W., and Moon, I.-C. Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. In *International Conference on Machine Learning*, pp. 11201–11228. PMLR, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Dhariwal, P. Glow: generative flow with invertible $1\times$ 1 convolutions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 10236–10245, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. In *Advances in Neural Information Processing Systems*, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Liu, J., Li, C., Ren, Y., Chen, F., and Zhao, Z. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11020–11028, 2022a.

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

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

Lu, C., Zheng, K., Bao, F., Chen, J., Li, C., and Zhu, J. Maximum likelihood training for score-based diffusion odes by high order denoising score matching. In *International Conference on Machine Learning*, pp. 14429–14460. PMLR, 2022a.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, 2022b.

Meng, C., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. SDEdit: Image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Nichol, A. Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., Mcgrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pp. 16784–16804. PMLR, 2022.

Oord, A. v. d., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., and Kavukcuoglu, K. Conditional image generation with pixelcnn decoders. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 4797–4805, 2016.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

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

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations*, 2017.

Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J. F., and Luque, J. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International Conference on Learning Representations*, 2020.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 32, pp. 11895–11907, 2019.

Song, Y., Garg, S., Shi, J., and Ermon, S. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pp. 574–584. PMLR, 2020.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1415–1428, 2021b.

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

Uria, B., Murray, I., and Larochelle, H. RNADE: The real-valued neural autoregressive density-estimator. *Advances in Neural Information Processing Systems*, 26, 2013.

Vahdat, A. and Kautz, J. Nvae: a deep hierarchical variational autoencoder. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 19667–19679, 2020.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Xiao, Z., Yan, Q., and Amit, Y. Likelihood regret: an out-of-distribution detection score for variational auto-encoder. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 20685–20696, 2020.

Xu, Y., Liu, Z., Tegmark, M., and Jaakkola, T. S. Poisson flow generative models. In *Advances in Neural Information Processing Systems*, 2022.

Xu, Y., Liu, Z., Tian, Y., Tong, S., Tegmark, M., and Jaakkola, T. Pfgm++: Unlocking the potential of physics-inspired generative models. *arXiv preprint arXiv:2302.04265*, 2023.

Yang, R. and Mandt, S. Lossy image compression with conditional diffusion models. *arXiv preprint arXiv:2209.06950*, 2022.

Zhao, M., Bao, F., Li, C., and Zhu, J. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. In *Advances in Neural Information Processing Systems*, 2022.

A. Different perspective of diffusion ODEs for bridging the gap between discrete and continuous data

Suppose the discrete data $\boldsymbol{X}_0$ to be modelled are 8-bit integers $\{0, 1, \dots, 255\}$. Following the common transform in diffusion models, we first normalize it to range [-1,1] by the mapping $\boldsymbol{x}_0 = \frac{\boldsymbol{X}_0 + \frac{1}{2} - 128}{128}$. In the following discussions, we consider the model distribution $P_0(\boldsymbol{x}_0)$ on transformed discrete data $\boldsymbol{x}_0$, which is equal to $P_0(\boldsymbol{X}_0)$ since the scaling does not alter the discrete probability.

A.1. Dequantization perspective

The discrete data $\boldsymbol{x}_0$ has a uniform gap $\frac{1}{128}$ between two consecutive values on each dimension. We can define the discrete model distribution as

$$P_0(\boldsymbol{x}_0) = \int_{\boldsymbol{u} \in [-\frac{1}{256}, \frac{1}{256}]^d} p_\epsilon(\boldsymbol{x} + \boldsymbol{u}) \mathrm{d}\boldsymbol{u} \tag{20}$$

where p_ϵ is the diffusion ODE defined at time ϵ. Then, we can introduce a dequantization distribution $q(\boldsymbol{u}|\boldsymbol{x}_0)$ with support over $[-\frac{1}{256}, \frac{1}{256})^d$. Treating q as an approximate posterior, we obtain the following variational bound (Ho et al., 2019):

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{q(\boldsymbol{u}|\boldsymbol{x}_0)} \left[\log p_\epsilon(\boldsymbol{x}_0 + \boldsymbol{u}) - \log q(\boldsymbol{u}|\boldsymbol{x}_0)\right] \tag{21}$$

The ODE term $\log p_\epsilon(\boldsymbol{x}_0 + \boldsymbol{u})$ can be evaluated exactly by solving another ODE called "Instantaneous Change of Variables" (Chen et al., 2018a). As for the posterior $\log q(\boldsymbol{u}|\boldsymbol{x}_0)$, we can derive closed-form solutions for predefined posterior formulation. We provide the details for uniform dequantization and our proposed truncated-normal dequantization.

Uniform dequantization We simply use uniform posterior $q(\boldsymbol{u}|\boldsymbol{x}_0) = \mathcal{U}(-\frac{1}{256}, \frac{1}{256})$. In this case, $\log q(\boldsymbol{u}|\boldsymbol{x}_0) = d \log 128$ is a constant, and the bound becomes

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{\boldsymbol{u} \sim \mathcal{U}(-\frac{1}{256}, \frac{1}{256})} \left[\log p_\epsilon(\boldsymbol{x}_0 + \boldsymbol{u})\right] - d \log 128 \tag{22}$$

Similar to Burda et al. (2015), we can also sample multiple $\boldsymbol{u}$ to derive a tighter bound, which is called importance weighted estimator:

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(K)} \sim \mathcal{U}(-\frac{1}{256}, \frac{1}{256})} \left[\log \left(\frac{1}{K} \sum_{i=1}^{K} p_\epsilon(\boldsymbol{x}_0 + \boldsymbol{u}^{(i)})\right)\right] - d \log 128 \tag{23}$$

However, this dequantization will cause a training-evaluation gap. For training, we fit p_ϵ to the distribution of $\boldsymbol{x}_\epsilon = \alpha_\epsilon \boldsymbol{x}_0 + \sigma_\epsilon \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. For evaluation, we test p_ϵ on uniform dequantized $\boldsymbol{x}_0 + \boldsymbol{u}, \boldsymbol{u} \sim \mathcal{U}(-\frac{1}{256}, \frac{1}{256})$. This gap will degenerate the likelihood performance, as we will show later.

Truncated-normal dequantization To bridge the training-evaluation gap, we test p_ϵ on $\hat{\boldsymbol{x}}_\epsilon = \alpha_\epsilon \boldsymbol{x}_0 + \sigma_\epsilon \hat{\boldsymbol{\epsilon}}$, where $\hat{\boldsymbol{\epsilon}}$ obeys a truncated-normal distribution to make sure the range of $\boldsymbol{u}$ on each dimension does not exceed $[-\frac{1}{256}, \frac{1}{256}]$. Specifically, denote $\tau \coloneqq \frac{\alpha_\epsilon}{256 \sigma_\epsilon}$, we define the truncated-normal distribution as

$$\hat{\boldsymbol{\epsilon}} \sim \mathcal{TN}\left(\hat{\boldsymbol{\epsilon}} \,|\boldsymbol{0}, \boldsymbol{I}, -\tau, \tau\right) \tag{24}$$

Let

$$\boldsymbol{u} \coloneqq \frac{\sigma_\epsilon}{\alpha_\epsilon} \hat{\boldsymbol{\epsilon}} \in \left[-\frac{1}{256}, \frac{1}{256}\right] \tag{25}$$

By the change of variables for probability density, we have

$$\log p_\epsilon(\boldsymbol{x}_0 + \boldsymbol{u}) = \log p_\epsilon \left(\boldsymbol{x}_0 + \frac{\sigma_\epsilon}{\alpha_\epsilon} \hat{\boldsymbol{\epsilon}}\right) = \log p_\epsilon(\hat{\boldsymbol{x}}_\epsilon) + d \log \alpha_\epsilon \tag{26}$$

$$\log q(\boldsymbol{u}|\boldsymbol{x}_0) = \log q \left(\frac{\sigma_\epsilon}{\alpha_\epsilon} \hat{\boldsymbol{\epsilon}}\right) = \log q(\hat{\boldsymbol{\epsilon}}) + d \log \frac{\alpha_\epsilon}{\sigma_\epsilon} \tag{27}$$

where $q(\hat{\boldsymbol{\epsilon}})$ is the probability distribution function of truncated-normal distributions

$$q(\hat{\boldsymbol{\epsilon}}) = \frac{1}{(2\pi Z^2)^{\frac{d}{2}}} \exp(-\frac{1}{2}\|\hat{\boldsymbol{\epsilon}}\|_2^2), \quad Z \coloneqq \Phi(\tau) - \Phi(-\tau) = \text{erf}\left(\frac{\tau}{\sqrt{2}}\right) \tag{28}$$

Here $\Phi(\cdot)$ is the cumulative distribution function of standard normal distribution, and $\text{erf}(\cdot)$ is the error function. Combining the equations above, the bound is reduced to

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{q(\hat{\boldsymbol{\epsilon}})}\left[\log p_\epsilon(\hat{\boldsymbol{x}}_\epsilon) - \log q(\hat{\boldsymbol{\epsilon}})\right] + d\log\sigma_\epsilon \tag{29}$$

Further, we can derive closed-form solutions for the entropy term of truncated-normal distribution:

$$-\mathbb{E}_{q(\hat{\boldsymbol{\epsilon}})}[\log q(\hat{\boldsymbol{\epsilon}})] = \mathcal{H}(q(\hat{\boldsymbol{\epsilon}})) = d\log(\sqrt{2\pi e}) + d\log Z - d\frac{\tau}{\sqrt{2\pi}Z}\exp(-\frac{1}{2}\tau^2) \tag{30}$$

and we finally obtain the exact form of the bound:

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{\hat{\boldsymbol{\epsilon}}\sim\mathcal{TN}(\mathbf{0},\boldsymbol{I},-\tau,\tau)}\left[\log p_\epsilon(\hat{\boldsymbol{x}}_\epsilon)\right] + \frac{d}{2}(1+\log(2\pi\sigma_\epsilon^2)) + d\log Z - d\frac{\tau}{\sqrt{2\pi}Z}\exp(-\frac{1}{2}\tau^2) \tag{31}$$

where the ODE log-likelihood $\log p_\epsilon(\hat{\boldsymbol{x}}_\epsilon)$ can also be evaluated exactly. Similarly, we have the corresponding importance weighted estimator by modifying Eqn. (29):

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{\hat{\boldsymbol{\epsilon}}^{(1)},\ldots,\hat{\boldsymbol{\epsilon}}^{(K)}\sim\mathcal{TN}(\mathbf{0},\boldsymbol{I},-\tau,\tau)}\left[\log\left(\frac{1}{K}\sum_{i=1}^{K}\frac{p_\epsilon(\hat{\boldsymbol{x}}_\epsilon^{(i)})}{q(\hat{\boldsymbol{\epsilon}}^{(i)})}\right)\right] + d\log\sigma_\epsilon \tag{32}$$

where $\hat{\boldsymbol{x}}_\epsilon^{(i)} \coloneqq \alpha_\epsilon\boldsymbol{x}_0 + \sigma_\epsilon\hat{\boldsymbol{\epsilon}}^{(i)}$, and $q(\hat{\boldsymbol{\epsilon}})$ is expressed in Eqn. (28).

In our experiments, we choose the start time $\gamma_\epsilon = -13.3$. Under this setting, we have $\tau \approx 3$, and the truncated-normal distribution $\mathcal{TN}(\mathbf{0}, \boldsymbol{I}, -\tau, \tau)$ is almost the same as the standard normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ due to the 3-σ principle. Thus, $\boldsymbol{x}_\epsilon$ used in training and $\hat{\boldsymbol{x}}_\epsilon$ used in testing are virtually identically distributed, resulting in a negligible training-evaluation gap.

A.2. Variational perspective

From the variational perspective, we can view the transition from discrete $\boldsymbol{x}_0$ to continuous $\boldsymbol{x}_\epsilon$ as a variational autoencoder, where the prior $p_\epsilon(\boldsymbol{x}_\epsilon)$ is modeled by diffusion ODE, and the approximate posterior $q_{0\epsilon}(\boldsymbol{x}_\epsilon|\boldsymbol{x}_0)$ is the analytical Gaussian transition kernel in the forward diffusion process at the start. We have the variational bound:

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{q_{0\epsilon}(\boldsymbol{x}_\epsilon|\boldsymbol{x}_0)}\left[\log p_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon) + \log p_\epsilon(\boldsymbol{x}_\epsilon) - \log q_{0\epsilon}(\boldsymbol{x}_\epsilon|\boldsymbol{x}_0)\right] \tag{33}$$

where $q_{0\epsilon}(\boldsymbol{x}_\epsilon|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_\epsilon|\alpha_\epsilon\boldsymbol{x}_0, \sigma_\epsilon^2\boldsymbol{I})$. We want to use the reconstruction term $p_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon)$ to approximate $q_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon)$. Note that

$$q_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon) = \frac{q_{0\epsilon}(\boldsymbol{x}_\epsilon|\boldsymbol{x}_0)q_0(\boldsymbol{x}_0)}{q_\epsilon(\boldsymbol{x}_\epsilon)} \tag{34}$$

for small enough ϵ, we have $q_0(\boldsymbol{x}_0) \approx q_\epsilon(\boldsymbol{x}_\epsilon)$, so $q_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon) \propto q_{0\epsilon}(\boldsymbol{x}_\epsilon|\boldsymbol{x}_0) = \prod_i q_{0\epsilon}(\boldsymbol{x}_{\epsilon,i}|\boldsymbol{x}_{0,i})$, where i represents the i-th dimension. Thus, we also choose $p_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon)$ as a factorized distribution, following Kingma et al. (2021):

$$p_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon) = \prod_i p_{\epsilon 0}(\boldsymbol{x}_{0,i}|\boldsymbol{x}_{\epsilon,i}) \tag{35}$$

where each

$$p_{\epsilon 0}(\boldsymbol{x}_{0,i}|\boldsymbol{x}_{\epsilon,i}) \propto q_{0\epsilon}(\boldsymbol{x}_{\epsilon,i}|\boldsymbol{x}_{0,i}) \propto \exp\left(-\frac{(\boldsymbol{x}_{\epsilon,i} - \alpha_\epsilon\boldsymbol{x}_{0,i})^2}{2\sigma_\epsilon^2}\right) \tag{36}$$

As $\boldsymbol{x}_0$ is a discrete variable, the probability can be computed by softmax, so we have

$$\log p_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon) = \sum_{i=1}^{d}\log\text{softmax}_{j=0}^{255}\left(-\frac{(\boldsymbol{x}_{\epsilon,i} - \alpha_\epsilon j)^2}{2\sigma_\epsilon^2}\right)[\boldsymbol{x}_{0,i}] \tag{37}$$

Table 3. Likelihood results under different bound and number of importance samples K. $K = 1$ means we do not use importance weighted estimator.

NLL	Uniform Dequantization			Variational			Truncated-Normal Dequantization		
	$K = 1$	$K = 5$	$K = 20$	$K = 1$	$K = 5$	$K = 20$	$K = 1$	$K = 5$	$K = 20$
CIFAR-10 (VP)	2.74	2.72	2.71	2.60	2.59	2.58	2.60	2.58	2.57
CIFAR-10 (SP)	2.81	2.79	2.78	2.61	2.59	2.58	2.60	2.57	2.56
ImageNet-32 (VP)	3.52	3.51	3.50	3.46	3.44	3.44	3.45	3.44	3.43
ImageNet-32 (SP)	3.57	3.56	3.55	3.48	3.47	3.46	3.47	3.45	3.44

Besides, the Gaussian entropy term can be computed exactly

$$-\mathbb{E}_{q_{0\epsilon}(\boldsymbol{x}_\epsilon|\boldsymbol{x}_0)}[\log q_{0\epsilon}(\boldsymbol{x}_\epsilon|\boldsymbol{x}_0)] = \mathcal{H}(q_{0\epsilon}(\boldsymbol{x}_\epsilon|\boldsymbol{x}_0)) = \frac{d}{2}(1 + \log(2\pi\sigma_\epsilon^2)) \tag{38}$$

and the bound is reduced to

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\log p_\epsilon(\boldsymbol{x}_\epsilon) + \log p_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon)\right] + \frac{d}{2}(1 + \log(2\pi\sigma_\epsilon^2)) \tag{39}$$

where $\boldsymbol{x}_\epsilon = \alpha_\epsilon \boldsymbol{x}_0 + \sigma_\epsilon \boldsymbol{\epsilon}$, $\log p_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon)$ is given in Eqn. (37) and $\log p_\epsilon(\boldsymbol{x}_\epsilon)$ is the exact ODE likelihood. We also have the importance weighted estimator by modifying Eqn. (33):

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{\boldsymbol{\epsilon}^{(1)},\ldots,\boldsymbol{\epsilon}^{(K)}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\log\left(\frac{1}{K}\sum_{i=1}^{K}\frac{p_\epsilon(\boldsymbol{x}_\epsilon)p_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon)}{q_{0\epsilon}(\boldsymbol{x}_\epsilon|\boldsymbol{x}_0)}\right)\right] \tag{40}$$

A.3. Practical connections and results

Let us consider the bound without importance weighted estimator. By observing the bound in Eqn. (31) for truncated-normal dequantization and the bound in Eqn. (39) for variational perspective, we can find that they have similar formulations. Suppose we use $\gamma_\epsilon = -13.3$, we have $\tau \approx 3.01869$, $Z \approx 0.9974613$, and the bound in Eqn. (31) is approximately

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{\hat{\boldsymbol{\epsilon}}\sim\mathcal{TN}(\mathbf{0},\boldsymbol{I},-\tau,\tau)}\left[\log p_\epsilon(\hat{\boldsymbol{x}}_\epsilon)\right] + \frac{d}{2}(1 + \log(2\pi\sigma_\epsilon^2)) - 0.01522 \times d \tag{41}$$

Next, consider the variational perspective. Though the reconstruction term $\log p_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon)$ in Eqn. (39) depends on the data distribution, empirically it is nearly a constant $\log p_{\epsilon 0}(\boldsymbol{x}_0|\boldsymbol{x}_\epsilon) \approx -0.01 \times d$. So we have the approximate bound

$$\log P_0(\boldsymbol{x}_0) \geq \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\log p_\epsilon(\boldsymbol{x}_\epsilon)\right] + \frac{d}{2}(1 + \log(2\pi\sigma_\epsilon^2)) - 0.01 \times d \tag{42}$$

We note the only difference is that our proposed truncated-normal dequantization uses $\hat{\boldsymbol{x}}_\epsilon$ rather than $\boldsymbol{x}_\epsilon$ for ODE likelihood evaluation, and there is a small constant difference in the bound.

Remark A.1. For high-dimensional data such as images, directly comparing log-likelihood may suffer from scaling issues by the dimension. In practice, we usually compare the BPD (bits/dim) by

$$\text{BPD} = \mathbb{E}_{\boldsymbol{x}_0\sim q_0}\left[\frac{-\log P_0(\boldsymbol{x}_0)}{d\log 2}\right] \tag{43}$$

where q_0 is the data distribution. Since BPD averages the log-likelihood on each dimension, scaling dimensionality has no effect on the final result.

We test the two types of dequantization and the variational perspective on our final models, using different numbers of importance samples K. The results are listed in Table 3. Empirically, truncated-normal dequantization performs slightly better than variational, while uniform dequantization gives a bad likelihood due to the large training-evaluation gap. We also observe that increasing K further improves the results by giving a tighter bound.

Remark A.2. Since uniform dequantized data has a larger noise level than truncated-normal dequantized data, we find evaluating $\log p_\epsilon(\boldsymbol{x}_0 + \boldsymbol{u})$ at start time $\gamma_\epsilon = -13.3$ leads to bad likelihood. Thus, we tune γ_ϵ for uniform dequantization (Figure 4), and eventually choose $\gamma_\epsilon = -12.0, -11.9, -11.7, -11.6$ for CIFAR-10 (VP), CIFAR-10 (SP), ImageNet-32 (VP), ImageNet-32 (SP) respectively.

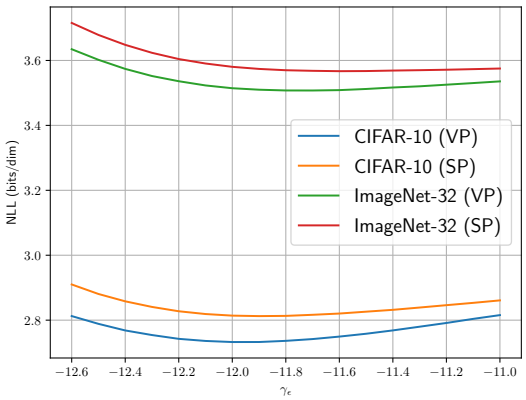


Figure 4. The likelihood evaluation results under uniform dequantization for different start times γ_ϵ. To plot the curve, we estimate the likelihood using the first 1024 test samples for CIFAR-10, and the first 512 test samples for ImageNet-32.

B. Equivalence of different predictors and matching objectives

We have the following theorem which demonstrates that different predictors are mutually transformable by a time-dependent skip connection, and they can be trained in a simulation-free approach by equivalent matching objectives.

Theorem B.1. *Let $\boldsymbol{x}_0$ be the sample from data distribution, and $\boldsymbol{\epsilon}$ be the sample from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Denote $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$, $\boldsymbol{v} = \dot{\alpha}_t \boldsymbol{x}_0 + \dot{\sigma}_t \boldsymbol{\epsilon}$. Suppose we have four kinds of predictors parameterized by θ and corresponding matching objectives with positive time weighting function $w(t)$:*

- *score predictor $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ and score matching loss $\mathcal{J}_{SM}(\theta, w(t)) = \mathbb{E}_t\left[w(t)\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}[\|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)\|_2^2]\right]$*
- *noise predictor $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ and noise matching loss $\mathcal{J}_{NM}(\theta, w(t)) = \mathbb{E}_t\left[w(t)\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\epsilon}\|_2^2]\right]$*
- *data predictor $\boldsymbol{x}_\theta(\boldsymbol{x}_t, t)$ and data matching loss $\mathcal{J}_{DM}(\theta, w(t)) = \mathbb{E}_t\left[w(t)\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}[\|\boldsymbol{x}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\|_2^2]\right]$*
- *velocity predictor $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$ and flow matching loss $\mathcal{J}_{FM}(\theta, w(t)) = \mathbb{E}_t\left[w(t)\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}[\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v}\|_2^2]\right]$*

For any $w(t)$, if we denote the optimal (ground-truth) predictors that minimize the corresponding matching losses as $\boldsymbol{s}^*(\boldsymbol{x}_t, t), \boldsymbol{\epsilon}^*(\boldsymbol{x}_t, t), \boldsymbol{x}^*(\boldsymbol{x}_t, t), \boldsymbol{v}^*(\boldsymbol{x}_t, t)$ respectively, then they are equivalent by the following relations:

$$\begin{aligned}
\boldsymbol{\epsilon}^*(\boldsymbol{x}_t, t) &= -\sigma_t \boldsymbol{s}^*(\boldsymbol{x}_t, t) \\
\boldsymbol{x}^*(\boldsymbol{x}_t, t) &= \frac{1}{\alpha_t}\boldsymbol{x}_t + \frac{\sigma_t^2}{\alpha_t}\boldsymbol{s}^*(\boldsymbol{x}_t, t) \\
\boldsymbol{v}^*(\boldsymbol{x}_t, t) &= f(t)\boldsymbol{x}_t - \frac{1}{2}g^2(t)\boldsymbol{s}^*(\boldsymbol{x}_t, t)
\end{aligned} \tag{44}$$

where $\boldsymbol{s}^*(\boldsymbol{x}_t, t) = \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$ is the ground-truth score.

Proof. For any positive weighting $w(t)$, the overall optimum of the matching loss $\mathbb{E}_t\left[w(t)\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}[\|\cdot\|_2^2]\right]$ is achieved when the optimum of the inner expectation $\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}[\|\cdot\|_2^2]$ is achieved for any t. For fixed t, by denoising score matching (Vincent, 2011), we know minimizing $\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}[\|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)\|_2^2]$ is equivalent to minimizing $\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}[\|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2] = \mathbb{E}_{q(\boldsymbol{x}_t)}\mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2]$, where $\log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) = -\frac{\boldsymbol{\epsilon}}{\sigma_t}$. The inner expectation is a minimum mean square error problem, so the optimal score predictor satisfies

$$\boldsymbol{s}^*(\boldsymbol{x}_t, t) = \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\nabla_{\boldsymbol{x}} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)] = -\frac{1}{\sigma_t}\mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{\epsilon}] \tag{45}$$

Similarly, for $\mathcal{J}_{\text{NM}}(\theta, w(t))$, the optimal noise predictor satisfies

$$\boldsymbol{\epsilon}^*(\boldsymbol{x}_t, t) = \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{\epsilon}] = -\sigma_t \boldsymbol{s}^*(\boldsymbol{x}_t, t) \tag{46}$$

For $\mathcal{J}_{\text{DM}}(\theta, w(t))$, the optimal data predictor satisfies

$$\begin{aligned}\boldsymbol{x}^*(\boldsymbol{x}_t, t) &= \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0] \\ &= \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\frac{\boldsymbol{x}_t - \sigma_t\boldsymbol{\epsilon}}{\alpha_t}\right] \\ &= \frac{1}{\alpha_t}\boldsymbol{x}_t - \frac{\sigma_t}{\alpha_t}\mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{\epsilon}] \\ &= \frac{1}{\alpha_t}\boldsymbol{x}_t + \frac{\sigma_t^2}{\alpha_t}\boldsymbol{s}^*(\boldsymbol{x}_t, t)\end{aligned} \tag{47}$$

For $\mathcal{J}_{\text{FM}}(\theta, w(t))$, the optimal velocity predictor satisfies

$$\begin{aligned}\boldsymbol{v}^*(\boldsymbol{x}_t, t) &= \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\dot{\alpha}_t\boldsymbol{x}_0 + \dot{\sigma}_t\boldsymbol{\epsilon}] \\ &= \dot{\alpha}_t\mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0] + \dot{\sigma}_t\mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{\epsilon}] \\ &= \frac{\dot{\alpha}_t}{\alpha_t}\boldsymbol{x}_t + \left(\frac{\dot{\alpha}_t}{\alpha_t}\sigma_t^2 - \sigma_t\dot{\sigma}_t\right)\boldsymbol{s}^*(\boldsymbol{x}_t, t) \\ &= f(t)\boldsymbol{x}_t - \frac{1}{2}g^2(t)\boldsymbol{s}^*(\boldsymbol{x}_t, t)\end{aligned} \tag{48}$$

□

The equivalence of optimal predictors also implies the equivalence of parameterized predictors. From the above theorem, we know $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$ and $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ are related by $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) = f(t)\boldsymbol{x}_t + \frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$. In practice, we use γ timing. From the relationship $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{v}_\theta(\boldsymbol{x}_\gamma, \gamma)\frac{\mathrm{d}\gamma}{\mathrm{d}t}, \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_\gamma, \gamma)$, we obtain the noise predictor expressed by $\boldsymbol{v}_\theta(\boldsymbol{x}_\gamma, \gamma)$

$$\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_\gamma, \gamma) = 2\frac{\boldsymbol{v}_\theta(\boldsymbol{x}_\gamma, \gamma) - \frac{\dot{\alpha}_\gamma}{\alpha_\gamma}\boldsymbol{x}_\gamma}{\sigma_\gamma} \tag{49}$$

Further, we can replace $\boldsymbol{v}_\theta(\boldsymbol{x}_\gamma, \gamma)$ with the normalized velocity predictor $\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma, \gamma) = \boldsymbol{v}_\theta(\boldsymbol{x}_\gamma, \gamma)/\sqrt{\dot{\alpha}_\gamma^2 + \dot{\sigma}_\gamma^2}$.

Moreover, we can derive the equivalent training objectives under different parameterizations by employing the relations discussed above freely. For example, when we replace the normalized velocity predictor $\tilde{\boldsymbol{v}}_\theta$ with the score predictor $\boldsymbol{s}_\theta$ in the second-order objective Eqn. (16), we can obtain the second-order denoising score matching similar to Lu et al. (2022a). However, though theoretically equivalent, the actual performance of these objectives highly depends on the specific model architecture, hyperparameters and parameterization, and the authors of Lu et al. (2022a) find that their high-order denoising score matching objectives only work for VE schedule, but degenerate the performance of pretrained models with VP schedule.

C. Specifications under VP and SP schedule

As stated in Section 4.3, using γ timing and normalized velocity predictor $\tilde{\boldsymbol{v}}_\theta$, the likelihood weighted first-order and second-order flow matching objectives are reformulated as:

$$\mathcal{J}_{\text{FM}} = \int_{\gamma_0}^{\gamma_T} 2\frac{\dot{\alpha}_\gamma^2 + \dot{\sigma}_\gamma^2}{\sigma_\gamma^2}\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\|\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma, \gamma) - \tilde{\boldsymbol{v}}\|_2^2\mathrm{d}\gamma \tag{50}$$

$$\mathcal{J}_{\text{FM,tr}} = \int_{\gamma_0}^{\gamma_T} 2\frac{\dot{\alpha}_\gamma^2 + \dot{\sigma}_\gamma^2}{\sigma_\gamma^2}\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left(\sigma_\gamma\mathrm{tr}(\nabla\tilde{\boldsymbol{v}}_\theta) - \frac{\dot{\sigma}_\gamma}{\sqrt{\dot{\alpha}_\gamma^2 + \dot{\sigma}_\gamma^2}}d + \frac{2\sqrt{\dot{\alpha}_\gamma^2 + \dot{\sigma}_\gamma^2}}{\sigma_\gamma}\|\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma, \gamma) - \tilde{\boldsymbol{v}}\|_2^2\right)^2\mathrm{d}\gamma \tag{51}$$

where $\boldsymbol{v} = \dot{\alpha}_\gamma\boldsymbol{x}_0 + \dot{\sigma}_\gamma\boldsymbol{\epsilon}, \tilde{\boldsymbol{v}} = \boldsymbol{v}/\sqrt{\dot{\alpha}_\gamma^2 + \dot{\sigma}_\gamma^2}$. For VP and SP schedule, since $\gamma = \log(\sigma_\gamma^2/\alpha_\gamma^2)$, using their schedule properties, $\alpha_\gamma, \sigma_\gamma$ are deterministic functions of γ without any hyperparameters. Thus, we can derive their specific

Table 4. Specification of related values and objectives under VP and SP schedule.

Formula	VP	SP
α_γ	$\sqrt{\frac{1}{1+\exp(\gamma)}}$	$\frac{1}{1+\exp(\gamma/2)}$
σ_γ	$\sqrt{\frac{1}{1+\exp(-\gamma)}}$	$\frac{1}{1+\exp(-\gamma/2)}$
$\dot{\alpha}_\gamma$	$-\frac{1}{2}\alpha_\gamma\sigma_\gamma^2$	$-\frac{1}{2}\alpha_\gamma\sigma_\gamma$
$\dot{\sigma}_\gamma$	$\frac{1}{2}\alpha_\gamma^2\sigma_\gamma$	$\frac{1}{2}\alpha_\gamma\sigma_\gamma$
$\sqrt{\dot{\alpha}_\gamma^2+\dot{\sigma}_\gamma^2}$	$\frac{1}{2}\alpha_\gamma\sigma_\gamma$	$\frac{1}{\sqrt{2}}\alpha_\gamma\sigma_\gamma$
$\tilde{\boldsymbol{v}}$	$\alpha_\gamma\boldsymbol{\epsilon}-\sigma_\gamma\boldsymbol{x}_0$	$\frac{\boldsymbol{\epsilon}-\boldsymbol{x}_0}{\sqrt{2}}$
$\mathcal{J}_{\text{FM}}$	$\frac{1}{2}\int_{\gamma_0}^{\gamma_T}\alpha_\gamma^2\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\|\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma,\gamma)-\tilde{\boldsymbol{v}}\|_2^2\mathrm{d}\gamma$	$\int_{\gamma_0}^{\gamma_T}\alpha_\gamma^2\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\|\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma,\gamma)-\tilde{\boldsymbol{v}}\|_2^2\,\mathrm{d}\gamma$
$\mathcal{J}_{\text{FM,tr}}$	$\frac{1}{2}\int_{\gamma_0}^{\gamma_T}\alpha_\gamma^2\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left(\sigma_\gamma\mathrm{tr}(\nabla\tilde{\boldsymbol{v}}_\theta)-\alpha_\gamma d+\alpha_\gamma\|\hat{\tilde{\boldsymbol{v}}}_\theta-\tilde{\boldsymbol{v}}\|_2^2\right)^2\mathrm{d}\gamma$	$\int_{\gamma_0}^{\gamma_T}\alpha_\gamma^2\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left(\sigma_\gamma\mathrm{tr}(\nabla\tilde{\boldsymbol{v}}_\theta)-\frac{1}{\sqrt{2}}d+\sqrt{2}\alpha_\gamma\|\hat{\tilde{\boldsymbol{v}}}_\theta-\tilde{\boldsymbol{v}}\|_2^2\right)^2\mathrm{d}\gamma$
$\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_\gamma,\gamma)$	$\sigma_\gamma\boldsymbol{x}_\gamma+\alpha_\gamma\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma,\gamma)$	$\boldsymbol{x}_\gamma+\sqrt{2}\alpha_\gamma\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma,\gamma)$

objectives and equivalent predictors using the formula for general noise schedules. We summarize them in Table C, where $\hat{\tilde{\boldsymbol{v}}}_\theta$ denotes the stop-gradient version of $\tilde{\boldsymbol{v}}_\theta$.

Next, we derive the designed IS procedure. We want to choose a proposal distribution $p(\gamma)\propto\frac{\dot{\alpha}_\gamma^2+\dot{\sigma}_\gamma^2}{\sigma_\gamma^2}$, which is proportional α_γ^2 for VP and SP. Since we have explicit expressions for the density, we utilize *inverse transform sampling* to design a sampling procedure. Concretely, we take uniform samples of a number $t\in[0,1]$, and solve the following equation about γ_t:

$$\frac{1}{Z}\int_{\gamma_0}^{\gamma_t}\alpha_\gamma^2\mathrm{d}\gamma=t,\quad Z=\int_{\gamma_0}^{\gamma_1}\alpha_\gamma^2\mathrm{d}\gamma \tag{52}$$

Here we assume maximum time $T=1$, and Z is a normalizing constant.

VP We have (omit the constant of the indefinite integral)

$$\int\alpha_\gamma^2\mathrm{d}\gamma=\log\frac{1}{1+\exp(-\gamma)}=\log\alpha_\gamma^2 \tag{53}$$

Then the equation for inverse transform sampling is

$$\log\frac{1}{1+\exp(-\gamma_t)}-\log\alpha_{\gamma_0}^2=Zt,\quad Z=\log\frac{\sigma_{\gamma_1}^2}{\sigma_{\gamma_0}^2} \tag{54}$$

The solution has a closed-form expression, which gives the inverse transformation from t to γ

$$\gamma_t=\log\frac{1}{\exp(-Zt)/\sigma_{\gamma_0}^2-1},\quad t\sim\mathcal{U}(0,1) \tag{55}$$

SP We have (omit the constant of the indefinite integral)

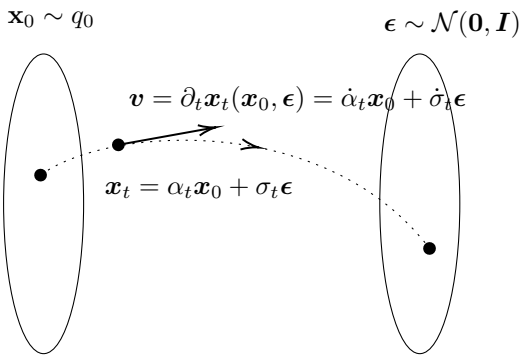
$$\int\alpha_\gamma^2\mathrm{d}\gamma=-2\left(\log(1+\exp(-\gamma/2))+\frac{1}{1+\exp(-\gamma/2)}\right) \tag{56}$$

Denote $F(\gamma)=-\log(1+\exp(-\gamma/2))-(1+\exp(-\gamma/2))^{-1}$, then the equation for inverse transform sampling is

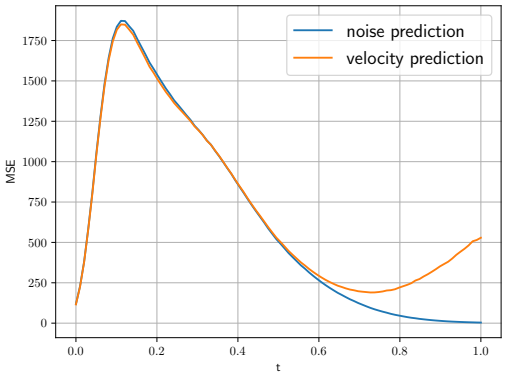
$$\frac{F(\gamma_t)-F(\gamma_0)}{F(\gamma_1)-F(\gamma_0)}=t \tag{57}$$

The solution has no closed-form expressions. Similar to the implementation in Song et al. (2021b), we use the bisection method to find the root.

D. Illustration of velocity prediction and imbalance problem



(a) Illustration of velocity prediction. Left ellipse: $\boldsymbol{x}_0$ sampled from the data distribution. Right ellipse: $\boldsymbol{\epsilon}$ sampled from standard Gaussian distribution. By independently drawing a pair $(\boldsymbol{x}_0, \boldsymbol{\epsilon})$, we can construct a diffusion path using the noise schedule.



(b) Mean square loss at different time t. We plot $\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\epsilon}\|_2^2\right]$ and $\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\|\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_t, t) - \tilde{\boldsymbol{v}}\|_2^2\right]$ for noise and velocity prediction on our pretrained model, tested on 32 data samples $\boldsymbol{x}_0$ and 20 noise samples $\boldsymbol{\epsilon}$.

Figure 5. Illustration of velocity prediction and imbalance problem.

First, we give an intuitive illustration of our velocity parameterization and corresponding flow matching objective in Section 4.1. As shown in Figure 5(a), for each pair $(\boldsymbol{x}_0, \boldsymbol{\epsilon})$ where $\boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, let $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$. As t increases, $\boldsymbol{x}_t$ moves from $\boldsymbol{x}_0$ to $\boldsymbol{\epsilon}$ gradually, forming a diffusion path in the sample space, and $\boldsymbol{v}$ is the velocity $\frac{\partial \boldsymbol{x}_t(\boldsymbol{x}_0, \boldsymbol{\epsilon})}{\partial t}$ across the path. Thus, minimizing $\mathcal{J}_{\text{FM}}$ is to predict the expected velocity for all possible $(\boldsymbol{x}_0, \boldsymbol{\epsilon})$ pairs.

Next, we interpret the superiority of velocity prediction from the perspective of balanced prediction difficulty. Intuitively, the noise prediction model suffers from an imbalance problem: at small t, $\boldsymbol{x}_t$ is similar to data, and extracting the insignificant noise component is hard; at large t, $\boldsymbol{x}_t$ is similar to noise, so the noise prediction is easy and has a small error. Velocity prediction, on the other hand, has a property that the prediction target $\boldsymbol{v}$ is less relevant to input $\boldsymbol{x}_t$. In Fig. 5(b) we empirically confirm it on our pretrained model. We plot the mean square prediction error (MSE) w.r.t. time t, which shows that velocity prediction alleviates the imbalance problem by enlarging the training at large t. Since the overall error is a weighted combination of the MSE at different t and is invariant to the parameterization, we can conclude that under noise prediction, the MSE is lower near $t = 1$, but is imposed a larger weight, so it has a larger gradient variance.

E. Relationship between velocity parameterization and other works

In this section, we demonstrate how the techniques in related works (Karras et al., 2022; Lipman et al., 2022; Salimans & Ho, 2022; Ho et al., 2022) can be reformulated as velocity parameterization.

E.1. Interpretation by preconditioning

Works that aim at improving the sample quality of diffusion models also consider the network parameterizations that adaptively mix signal and noise. Karras et al. (2022) proposes to precondition the neural network with a time-dependent skip connection that allows it to estimate either data $\boldsymbol{x}_0$ or noise $\boldsymbol{\epsilon}$, or something in between. Similarly, we write the noise predictor $\boldsymbol{\epsilon}_\theta(\cdot)$ in the following formulation:

$$\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_\gamma, \gamma) = c_{\text{skip}}(\gamma)\boldsymbol{x}_\gamma + c_{\text{out}}(\gamma)F_\theta(c_{\text{in}}(\gamma)\boldsymbol{x}_\gamma, \gamma) \tag{58}$$

where $F_\theta(\cdot)$ is the pure network, $\boldsymbol{x}_\gamma = \alpha_\gamma \boldsymbol{x}_0 + \sigma_\gamma \boldsymbol{\epsilon}$. The flow matching loss can be rewritten as

$$\begin{aligned}
\mathcal{J}_{\text{FM}}(\theta) &= \frac{1}{2}\int_{\gamma_0}^{\gamma_T} \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_\gamma, \gamma) - \boldsymbol{\epsilon}\|_2^2\right] \\
&= \frac{1}{2}\int_{\gamma_0}^{\gamma_T} \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\|c_{\text{skip}}(\gamma)\boldsymbol{x}_\gamma + c_{\text{out}}(\gamma)F_\theta(c_{\text{in}}(\gamma)\boldsymbol{x}_\gamma, \gamma) - \boldsymbol{\epsilon}\|_2^2\right] \\
&= \frac{1}{2}\int_{\gamma_0}^{\gamma_T} \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[c_{\text{out}}(\gamma)^2\|F_\theta(c_{\text{in}}(\gamma)\boldsymbol{x}_\gamma, \gamma) - F_{\text{target}}(\boldsymbol{x}_0, \boldsymbol{\epsilon}, \gamma)\|_2^2\right]
\end{aligned} \tag{59}$$

where

$$F_{\text{target}}(\boldsymbol{x}_0, \boldsymbol{\epsilon}, \gamma) = \frac{\boldsymbol{\epsilon} - c_{\text{skip}}(\gamma)\boldsymbol{x}_\gamma}{c_{\text{out}}(\gamma)} \tag{60}$$

Following first principles in EDM, We derive formulas for $c_{\text{in}}(\gamma), c_{\text{out}}(\gamma), c_{\text{skip}}(\gamma)$ to ensure:

1. The training inputs of $F_\theta(\cdot)$ have unit variance.
2. The effective training target F_{target} has unit variance.
3. We select $c_{\text{skip}}(\gamma)$ to minimize $c_{\text{out}}(\gamma)$, so that the errors of F_θ are amplified as little as possible.

From principle 1, we have

$$\begin{aligned} 1 &= \text{Var}\left[c_{\text{in}}(\gamma)\boldsymbol{x}_\gamma\right] \\ 1 &= \text{Var}\left[c_{\text{in}}(\gamma)(\alpha_\gamma \boldsymbol{x}_0 + \sigma_\gamma \boldsymbol{\epsilon})\right] \\ 1 &= c_{\text{in}}^2(\gamma)(\alpha_\gamma^2\sigma_{\text{data}}^2 + \sigma_\gamma^2) \\ c_{\text{in}}(\gamma) &= \frac{1}{\sqrt{\sigma_\gamma^2 + \sigma_{\text{data}}^2\alpha_\gamma^2}} \end{aligned} \tag{61}$$

From principle 2, we have

$$\begin{aligned} 1 &= \text{Var}\left[F_{\text{target}}(\boldsymbol{x}_0, \boldsymbol{\epsilon}, \gamma)\right] \\ 1 &= \text{Var}\left[\frac{\boldsymbol{\epsilon} - c_{\text{skip}}(\gamma)\boldsymbol{x}_\gamma}{c_{\text{out}}(\gamma)}\right] \\ c_{\text{out}}^2(\gamma) &= \text{Var}\left[\boldsymbol{\epsilon} - c_{\text{skip}}(\gamma)\boldsymbol{x}_\gamma\right] \\ c_{\text{out}}^2(\gamma) &= \text{Var}\left[\boldsymbol{\epsilon} - c_{\text{skip}}(\gamma)(\alpha_\gamma \boldsymbol{x}_0 + \sigma_\gamma \boldsymbol{\epsilon})\right] \\ c_{\text{out}}^2(\gamma) &= \text{Var}\left[(1 - c_{\text{skip}}(\gamma)\sigma_\gamma)\boldsymbol{\epsilon} - c_{\text{skip}}(\gamma)\alpha_\gamma \boldsymbol{x}_0\right] \\ c_{\text{out}}^2(\gamma) &= (1 - c_{\text{skip}}(\gamma)\sigma_\gamma)^2 + c_{\text{skip}}^2(\gamma)\alpha_\gamma^2\sigma_{\text{data}}^2 \end{aligned} \tag{62}$$

From principle 3, we have

$$\begin{aligned} 0 &= \frac{\mathrm{d}c_{\text{out}}^2(\gamma)}{\mathrm{d}c_{\text{skip}}(\gamma)} \\ 0 &= -2\sigma_\gamma(1 - \sigma_\gamma c_{\text{skip}}(\gamma)) + 2\alpha_\gamma^2\sigma_{\text{data}}^2 c_{\text{skip}}(\gamma) \\ c_{\text{skip}}(\gamma) &= \frac{\sigma_\gamma}{\sigma_\gamma^2 + \sigma_{\text{data}}^2\alpha_\gamma^2} \end{aligned} \tag{63}$$

We now substitute Eqn. (63) into Eqn. (62) to obtain the formula for $c_{\text{out}}(\gamma)$:

$$c_{\text{out}}(\gamma) = \frac{\sigma_{\text{data}}\alpha_\gamma}{\sqrt{\sigma_\gamma^2 + \sigma_{\text{data}}^2\alpha_\gamma^2}} \tag{64}$$

If we assume $\sigma_{\text{data}} = 1$ and consider VP schedule, we have $\alpha_\gamma^2 + \sigma_\gamma^2 = 1$, and the coefficients are reduced to

$$c_{\text{in}}(\gamma) = 1, \quad c_{\text{skip}}(\gamma) = \sigma_\gamma, \quad c_{\text{out}}(\gamma) = \alpha_\gamma \tag{65}$$

In this case, the preconditioning is in agreement with our velocity parameterization by $\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma, \gamma) = F_\theta(\boldsymbol{x}_\gamma, \gamma)$. In practice, we find setting $\sigma_{\text{data}} = 0.5$ as in Karras et al. (2022) leads to faster descent of the loss at the start, but slower convergence as the training proceeds.

E.2. Connection to flow matching in Lipman et al. (2022)

Lipman et al. (2022) defines a conditional probability path $p_t(\boldsymbol{x}|\boldsymbol{x}_0)$ that gradually moves the data $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$ to a target distribution $p_1(\boldsymbol{x})$. Note that they use $t = 1$ to represent data distribution and $t = 0$ to represent target distribution. To be consistent, we reverse their time representation. They obtain the marginal probability path by marginalizing over $q(\boldsymbol{x}_0)$:

$$p_t(\boldsymbol{x}) = \int p_t(\boldsymbol{x}|\boldsymbol{x}_0)q(\boldsymbol{x}_0)\mathrm{d}\boldsymbol{x}_0 \tag{66}$$

They want to learn a vector field $\boldsymbol{v}_t(\boldsymbol{x})$, which defines a flow $\boldsymbol{\phi} : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ by

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\phi}_t(\boldsymbol{x}) = \boldsymbol{v}_t(\boldsymbol{\phi}_t(\boldsymbol{x})), \quad \boldsymbol{\phi}_1(\boldsymbol{x}) = \boldsymbol{x} \tag{67}$$

so that the marginal p_t can be generated by the push-forward $p_t = [\boldsymbol{\phi}_t]_* p_1$. In practice, they consider the Gaussian conditional probability paths

$$p_t(\boldsymbol{x}|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_t(\boldsymbol{x}_0), \sigma_t^2(\boldsymbol{x}_0)\boldsymbol{I}) \tag{68}$$

and propose a conditional flow matching (CFM) objective for simulation-free training of $\boldsymbol{v}_t(\boldsymbol{x})$

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(\boldsymbol{x}_0),p_t(\boldsymbol{x}|\boldsymbol{x}_0)}\|\boldsymbol{v}_t(\boldsymbol{x}) - \boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{x}_0)\|_2^2 \tag{69}$$

where

$$\boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{x}_0) = \frac{\sigma_t'(\boldsymbol{x}_0)}{\sigma_t(\boldsymbol{x}_0)}(\boldsymbol{x} - \boldsymbol{\mu}_t(\boldsymbol{x}_0)) + \boldsymbol{\mu}_t'(\boldsymbol{x}_0) \tag{70}$$

Suppose the mean $\boldsymbol{\mu}_t(\boldsymbol{x}_0)$ is linear to $\boldsymbol{x}_0$, and the standard deviation $\sigma_t(\boldsymbol{x}_0)$ is invariant to $\boldsymbol{x}_0$, as the two experimented cases in flow matching. By setting $\boldsymbol{\mu}_t(\boldsymbol{x}_0) = \alpha_t\boldsymbol{x}_0, \sigma_t(\boldsymbol{x}_0) = \sigma_t$, we have

$$\boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{x}_0) = \frac{\dot{\sigma}_t}{\sigma_t}(\boldsymbol{x} - \alpha_t\boldsymbol{x}_0) + \dot{\alpha}_t\boldsymbol{x}_0 = \dot{\alpha}_t\boldsymbol{x}_0 + \dot{\sigma}_t\boldsymbol{\epsilon} \tag{71}$$

where we use $\boldsymbol{x} = \alpha_t\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ since $p_t(\boldsymbol{x}|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}|\alpha_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I})$. Then we can observe that they are corresponding to our notations: the conditional probability path $p_t(\boldsymbol{x}|\boldsymbol{x}_0)$ corresponds to the Gaussian transition kernel $q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)$ of the forward diffusion process; the marginal probability path $p_t(\boldsymbol{x})$ corresponds to the ground-truth marginals $q_t(\boldsymbol{x}_t)$ associated with the forward diffusion process; the matching target $\boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{x}_0)$ in CFM corresponds to the velocity of the diffusion path $\boldsymbol{v} = \dot{\alpha}_t\boldsymbol{x}_0 + \dot{\sigma}_t\boldsymbol{\epsilon}$ in our formulation.

Therefore, the CFM objective in Lipman et al. (2022) is actually velocity parameterization when specific to Gaussian diffusion processes, which is similar to our first-order objective of the pretraining phase. We can express CFM in a simpler form, which is easier to analyze and generalize to any noise schedule. Then by the equivalence of different predictors (Theorem B.1) and the relationship between $f(t), g(t)$ and α_t, σ_t, we have

$$\begin{aligned}
\mathcal{L}_{\text{CFM}}(\theta) &= \int_0^T \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v}\|_2^2\mathrm{d}t \\
&= \int_0^T \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left\|f(t)\boldsymbol{x}_t - \frac{1}{2}g^2(t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - (\dot{\alpha}_t\boldsymbol{x}_0 + \dot{\sigma}_t\boldsymbol{\epsilon})\right\|_2^2\mathrm{d}t \\
&= \int_0^T \frac{1}{4}g^4(t)\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left\|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) + \frac{\boldsymbol{\epsilon}}{\sigma_t}\right\|\mathrm{d}t
\end{aligned} \tag{72}$$

which demonstrates that **the CFM objective not only changes the parameterization but also imposes a different time weighting $w(t) = \frac{1}{4}g^4(t)$ on the original denoising score matching objective**. When the training aims for improving the sample quality (e.g., FID), the optimal choice for $w(t)$ is still an open problem.

Comparing the CFM objective to our first-order objective Eqn. (50), the practical differences are that we use normalized predictor $\tilde{\boldsymbol{v}}_\theta$, γ timing, and apply likelihood weighting. The likelihood weighting refers to time weighting $w(t) = \frac{g^2(t)}{2\sigma_t^2}$ in Eqn. (5) and $w(t) = \frac{2}{g^2(t)}$ in Eqn. (12), which is consistent under different parameterizations and is the theoretically optimal

choice for maximum likelihood training (Song et al., 2021c). Also, changing the time domain from t to γ will not alter the value of the objective, but will affect the variance of Monte-Carlo estimation and the convergence speed, as we have discussed. For example, the OT path in Lipman et al. (2022) is $\alpha_t = 1-t, \sigma_t = 1-(1-\sigma_{\min})(1-t) \approx t$, and the relation between γ and t is $\gamma = \log(\sigma_t^2/\alpha_t^2) = 2\log(t/(1-t))$. Under γ timing, we can decouple the choice of noise schedules to the greatest extent, and regard the change of variable from γ to t as a tunable importance sampling procedure.

Besides, normalizing the field is necessary for stable training of the velocity predictor and is the key to unifying v prediction and preconditioning. Such strategies have also been adopted in more general physics-inspired generative models. For example, Xu et al. (2022; 2023) propose to normalize the Poission field when training Poisson flow generative models.

E.3. Connection to v prediction

In Salimans & Ho (2022); Ho et al. (2022), a technique called "v prediction" is used, which parameterizes a network to predict $\mathbf{v} = \alpha_t \boldsymbol{\epsilon} - \sigma_t \boldsymbol{x}_0$. Assuming a VP schedule following their choice, we have $\alpha_t^2 + \sigma_t^2 = 1$, so by taking the derivative w.r.t. t we have $\alpha_t \dot{\alpha}_t + \sigma_t \dot{\sigma}_t = 0$, then

$$\dot{\alpha}_t = -\frac{\sigma_t \dot{\sigma}_t}{\alpha_t}, \quad \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t} = \frac{\dot{\alpha}_t}{\alpha_t} = -\frac{\sigma_t\dot{\sigma}_t}{\alpha_t^2} \tag{73}$$

so

$$\begin{aligned} g^2(t) &= \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\sigma_t^2 \\ &= 2\sigma_t\dot{\sigma}_t + 2\frac{\sigma_t\dot{\sigma}_t}{\alpha_t^2}\sigma_t^2 \\ &= \frac{2\sigma_t\dot{\sigma}_t}{\alpha_t^2}(\alpha_t^2+\sigma_t^2) \\ &= \frac{2\sigma_t\dot{\sigma}_t}{\alpha_t^2} \end{aligned} \tag{74}$$

and the velocity is

$$\begin{aligned} \boldsymbol{v} &= \dot{\alpha}_t\boldsymbol{x}_0 + \dot{\sigma}_t\boldsymbol{\epsilon} \\ &= \dot{\sigma}_t\boldsymbol{\epsilon} - \frac{\sigma_t\dot{\sigma}_t}{\alpha_t}\boldsymbol{x}_0 \\ &= \frac{\dot{\sigma}_t}{\alpha_t}(\alpha_t\boldsymbol{\epsilon} - \sigma_t\boldsymbol{x}_0) \\ &= \frac{\alpha_t}{2\sigma_t}g^2(t)(\alpha_t\boldsymbol{\epsilon} - \sigma_t\boldsymbol{x}_0) \end{aligned} \tag{75}$$

Besides, we can compute the normalizing factor as

$$\sqrt{\dot{\alpha}_t^2 + \dot{\sigma}_t^2} = \sqrt{\frac{\sigma_t^2\dot{\sigma}_t^2}{\alpha_t^2} + \dot{\sigma}_t^2} = \frac{\dot{\sigma}_t}{\alpha_t} = \frac{\alpha_t}{2\sigma_t}g^2(t) \tag{76}$$

so we have the normalized velocity

$$\tilde{\boldsymbol{v}} = \frac{\boldsymbol{v}}{\sqrt{\dot{\alpha}_t^2 + \dot{\sigma}_t^2}} = \alpha_t\boldsymbol{\epsilon} - \sigma_t\boldsymbol{x}_0 \tag{77}$$

Therefore, $\mathbf{v} = \tilde{\boldsymbol{v}}$, which means that v prediction is a special case of velocity parameterization when the noise schedule is VP.

F. Error-bounded trace of second-order flow matching

Here we provide the proofs for the error-bounded trace of second-order flow matching. First, we provide a lemma that gives the Jacobian of the ground-truth velocity predictor $\boldsymbol{v}^*(\boldsymbol{x}_t, t)$.

Lemma F.1. *Suppose* $(\boldsymbol{x}_0, \boldsymbol{x}_t) \sim q(\boldsymbol{x}_0, \boldsymbol{x}_t)$, *denote* $\boldsymbol{x}_t = \alpha_t\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}, \boldsymbol{v} = \dot{\alpha}_t\boldsymbol{x}_0 + \dot{\sigma}_t\boldsymbol{\epsilon}, \nabla(\cdot) = \nabla_{\boldsymbol{x}_t}(\cdot)$, *we have*

$$\nabla\boldsymbol{v}^*(\boldsymbol{x}_t, t) = \frac{\dot{\sigma}_t}{\sigma_t}\boldsymbol{I} - \frac{2}{g^2(t)}\mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[(\boldsymbol{v}^*(\boldsymbol{x}_t,t) - \boldsymbol{v})(\boldsymbol{v}^*(\boldsymbol{x}_t,t) - \boldsymbol{v})^\top\right] \tag{78}$$

and

$$\operatorname{tr}(\nabla \boldsymbol{v}^*(\boldsymbol{x}_t, t)) = \frac{\dot{\sigma}_t}{\sigma_t} d - \frac{2}{g^2(t)} \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[\|\boldsymbol{v}^*(\boldsymbol{x}_t, t) - \boldsymbol{v}\|_2^2 \right] \tag{79}$$

Proof. First, the gradient of q_{t0} can be calculated as

$$\begin{aligned}
\nabla q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) &= \nabla \frac{q_0(\boldsymbol{x}_0) q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q_t(\boldsymbol{x}_t)} \\
&= q_0(\boldsymbol{x}_0) \frac{q_t(\boldsymbol{x}_t) \nabla q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) \nabla q_t(\boldsymbol{x}_t)}{q_t(\boldsymbol{x}_t)^2} \\
&= \frac{q_0(\boldsymbol{x}_0) q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q_t(\boldsymbol{x}_t)} \left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t) \right) \\
&= q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t) \right) \\
&= \frac{2}{g^2(t)} (\boldsymbol{v}^*(\boldsymbol{x}_t, t) - \boldsymbol{v}) q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)
\end{aligned} \tag{80}$$

where we use the relation between $\boldsymbol{v}^*(\boldsymbol{x}_t, t)$ and $\nabla \log q_t(\boldsymbol{x}_t)$ in Theorem B.1. From Eqn. (48), we know $\boldsymbol{v}^*(\boldsymbol{x}_t, t) = \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{v}]$, and for given $\boldsymbol{x}_0$, we have

$$\nabla \boldsymbol{v} = \nabla \left(\dot{\alpha}_t \boldsymbol{x}_0 + \dot{\sigma}_t \frac{\boldsymbol{x}_t - \alpha_t \boldsymbol{x}_0}{\sigma_t} \right) = \frac{\dot{\sigma}_t}{\sigma_t} \boldsymbol{I} \tag{81}$$

and

$$\mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[(\boldsymbol{v}^*(\boldsymbol{x}_t, t) - \boldsymbol{v}) \boldsymbol{v}^*(\boldsymbol{x}_t, t)^\top \right] = \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[\boldsymbol{v}^*(\boldsymbol{x}_t, t) - \boldsymbol{v} \right] \boldsymbol{v}^*(\boldsymbol{x}_t, t)^\top = \boldsymbol{0} \tag{82}$$

So

$$\begin{aligned}
\nabla \boldsymbol{v}^*(\boldsymbol{x}_t, t) &= \nabla \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \boldsymbol{v} \mathrm{d}\boldsymbol{x}_0 \\
&= \int \nabla q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \boldsymbol{v}^\top + q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \nabla \boldsymbol{v} \mathrm{d}\boldsymbol{x}_0 \\
&= \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \left(\frac{2}{g^2(t)} (\boldsymbol{v}^*(\boldsymbol{x}_t, t) - \boldsymbol{v}) \boldsymbol{v}^\top + \frac{\dot{\sigma}_t}{\sigma_t} \boldsymbol{I} \right) \mathrm{d}\boldsymbol{x}_0 \\
&= \frac{\dot{\sigma}_t}{\sigma_t} \boldsymbol{I} + \frac{2}{g^2(t)} \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[(\boldsymbol{v}^*(\boldsymbol{x}_t, t) - \boldsymbol{v}) \boldsymbol{v}^\top \right] \\
&= \frac{\dot{\sigma}_t}{\sigma_t} \boldsymbol{I} - \frac{2}{g^2(t)} \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[(\boldsymbol{v}^*(\boldsymbol{x}_t, t) - \boldsymbol{v}) (\boldsymbol{v}^*(\boldsymbol{x}_t, t) - \boldsymbol{v})^\top \right]
\end{aligned} \tag{83}$$

The expression for $\operatorname{tr}(\nabla \boldsymbol{v}^*(\boldsymbol{x}_t, t))$ can be easily derived from the above equation. □

Then we prove Theorem 4.1 as follows.

Proof. The optimization in Eqn. (15) can be rewritten as

$$\theta^* = \operatorname*{argmin}_\theta \frac{2\sigma_t^2}{g^2(t)} \mathbb{E}_{q_t(\boldsymbol{x}_t)} \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[\left| \boldsymbol{v}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta) - \frac{\dot{\sigma}_t}{\sigma_t} d + \frac{2}{g^2(t)} \|\hat{\boldsymbol{v}}_1(\boldsymbol{x}_t, t) - \boldsymbol{v}\|_2^2 \right|^2 \right]. \tag{84}$$

For fixed t and $\boldsymbol{x}_t$, minimizing the inner expectation is a minimum mean square error problem for $\boldsymbol{v}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta)$, so the optimal θ^* satisfies

$$\boldsymbol{v}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta^*) = \frac{\dot{\sigma}_t}{\sigma_t} d - \frac{2}{g^2(t)} \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} [\|\hat{\boldsymbol{v}}_1(\boldsymbol{x}_t, t) - \boldsymbol{v}\|_2^2] \tag{85}$$

Using Lemma F.1 and $\boldsymbol{v}^*(\boldsymbol{x}_t, t) = \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{v}]$, we have

$$\begin{aligned}
&\operatorname{tr}(\nabla \boldsymbol{v}^*(\boldsymbol{x}_t, t)) - \boldsymbol{v}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta^*) \\
=&\frac{2}{g^2(t)} \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[\|\hat{\boldsymbol{v}}_1(\boldsymbol{x}_t, t)\|_2^2 - 2\boldsymbol{v}^\top \hat{\boldsymbol{v}}_1(\boldsymbol{x}_t, t) - \|\boldsymbol{v}^*(\boldsymbol{x}_t, t)\|_2^2 + 2\boldsymbol{v}^\top \boldsymbol{v}^*(\boldsymbol{x}_t, t) \right] \\
=&\frac{2}{g^2(t)} \|\hat{\boldsymbol{v}}_1(\boldsymbol{x}_t, t) - \boldsymbol{v}^*(\boldsymbol{x}_t, t)\|_2^2
\end{aligned} \tag{86}$$

Therefore, we can obtain the error bound by

$$\begin{aligned}
\left| \boldsymbol{v}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta) - \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{v}^*(\boldsymbol{x}_t, t)) \right| &\leq \left| \boldsymbol{v}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta) - \boldsymbol{v}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta^*) \right| + \left| \boldsymbol{v}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta^*) - \operatorname{tr}(\nabla \boldsymbol{v}^*(\boldsymbol{x}_t, t)) \right| \\
&= \left| \boldsymbol{v}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta) - \boldsymbol{v}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta^*) \right| + \frac{2}{g^2(t)} \delta_1^2(\boldsymbol{x}_t, t)
\end{aligned} \tag{87}$$

□

where $\delta_1(\boldsymbol{x}_t, t) = \|\hat{\boldsymbol{v}}_1(\boldsymbol{x}_t, t) - \boldsymbol{v}^*(\boldsymbol{x}_t, t)\|_2$ is the first-order estimation error.

G. Difference between our second-order flow matching and the previous time score matching in Choi et al. (2022)

We propose the error-bounded second-order flow matching objective to regularize $-\operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{v}_\theta(\boldsymbol{x}_t, t))$, which is equal to $\frac{\mathrm{d} \log p_t(\boldsymbol{x}_t)}{\mathrm{d}t}$ by the "Instantaneous Change of Variables" formula of CNFs (Chen et al., 2018a). Choi et al. (2022) proposes a joint score matching method to estimate the data score as well as the time score $(\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}), \partial_t \log p_t(\boldsymbol{x}))$, which seems related. However, they are essentially different.

Firstly, the change-of-variable for CNFs describes the total derivative of $\log p_t(\boldsymbol{x}_t)$ w.r.t. $\boldsymbol{x}_t$ which evolves by the ODE flow trajectory, not each fixed data point $\boldsymbol{x} \in \mathbb{R}^d$. However, $\partial_t \log p_t(\boldsymbol{x})$ in Choi et al. (2022) describes the partial derivative of $\log p_t(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathbb{R}^d$, i.e., any fixed data point in the whole space. Specifically, according to the Fokker-Planck equation, we have

$$\partial_t p_t(\boldsymbol{x}) = -\nabla_{\boldsymbol{x}} \cdot (p_t(\boldsymbol{x}) \boldsymbol{v}_\theta(\boldsymbol{x}, t)) \tag{88}$$

It follows that

$$\partial_t \log p_t(\boldsymbol{x}) = -\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v}_\theta(\boldsymbol{x}, t) - \boldsymbol{v}_\theta(\boldsymbol{x}, t)^\top \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) \tag{89}$$

Therefore, the total derivative $\frac{\mathrm{d} \log p_t(\boldsymbol{x}_t)}{\mathrm{d}t}$ we care about is different from the partial derivative $\partial_t \log p_t(x)$ in Choi et al. (2022), and their training objectives are also different (with different optimal solutions).

Moreover, there is another difference: Choi et al. (2022) trains another model to estimate the partial derivative $\partial_t \log p_t(x)$, which is independent of the ODE velocity $\boldsymbol{v}_\theta(\boldsymbol{x}, t)$ (in the form of the parameterized score function $\boldsymbol{s}_\theta(\boldsymbol{x}, t)$). However, our method restricts the parameterized velocity $\boldsymbol{v}_\theta(\boldsymbol{x}, t)$ itself, and does not employ another model.

Finally, the techniques used in Choi et al. (2022) and our work are also different. Choi et al. (2022) estimates the score matching loss for the partial derivative $\partial_t \log p_t(x)$ by the well-known integral-by-parts, which is used to derive the famous *sliced score matching* (Song et al., 2020), to avoid the computation of the score function $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$; However, our method leverages the property of mean square error (that its minimum is conditional mean), which is used to derive the famous *denoising score matching* (Vincent, 2011), to estimate the divergence of $\boldsymbol{v}_\theta(\boldsymbol{x}, t)$. In the score matching literature, sliced score matching and denoising score matching are two rather different techniques. As first-order denoising score matching is widely used in training diffusion models (such as Song et al. (2021c)), our proposed second-order flow matching is also suitable for training diffusion ODEs.

H. Details of our adaptive IS

In this section, we give details of our adaptive IS stated in Section 4.4. First, we parameterize $\gamma_\eta(t)$ similar to Kingma et al. (2021):

$$\gamma_\eta(t) = \gamma_0 + (\gamma_T - \gamma_0)\frac{\tilde{\gamma}_\eta(t) - \tilde{\gamma}_\eta(0)}{\tilde{\gamma}_\eta(1) - \tilde{\gamma}_\eta(0)} \tag{90}$$

where $\tilde{\gamma}_\eta(t)$ is a dense monotone increasing network. Concretely, we use a two-layer fully-connected network $\tilde{\gamma}_\eta(t) = l_2(\phi(l_1(t)))$ where ϕ is the sigmoid activation function, l_1, l_2 are linear layers with positive weight and output units of 1024 and 1.

Algorithm 1 Adaptive importance sampling (single iteration)

Require: velocity network $\boldsymbol{v}_\theta$, IS network $\tilde{\gamma}_\eta$, noise schedule $\alpha_\gamma, \sigma_\gamma$, batch size N
Sample $\boldsymbol{x}_0^{(1)}, \dots, \boldsymbol{x}_0^{(N)}$ from data distribution
Sample $\boldsymbol{\epsilon}^{(1)}, \dots, \boldsymbol{\epsilon}^{(N)}$ from standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$
Sample $t^{(1)}, \dots, t^{(N)}$ from uniform distribution $\mathcal{U}(0, 1)$
Caculate $\gamma_\eta(t^{(i)}), i = 1, \dots, N$ by Eqn. (90)
Fix η, optimize θ to minimize $\frac{1}{N}\sum_{i=1}^N \mathcal{L}_{\theta,\eta}(\boldsymbol{x}_0^{(i)}, \boldsymbol{\epsilon}^{(i)}, t^{(i)})$
Fix θ, optimize η to minimize $\frac{1}{N}\sum_{i=1}^N \mathcal{L}^2_{\theta,\eta}(\boldsymbol{x}_0^{(i)}, \boldsymbol{\epsilon}^{(i)}, t^{(i)})$

Then we present our adaptive IS procedure in Algorithm 1. Kingma et al. (2021) proposes to reuse the gradient $\nabla_\theta \mathcal{L}_{\theta,\eta}(\boldsymbol{x}_0, \boldsymbol{\epsilon}, t)$ to optimize η and avoid a second backpropagation by decomposing the gradient $\nabla_\eta \mathcal{L}^2_{\theta,\eta}(\boldsymbol{x}_0, \boldsymbol{\epsilon}, t)$ using chain-rule. We simply their learning of $\tilde{\gamma}_\eta$ by removing the complex gradient operation in one iteration and propose to alternatively optimize θ and η. It may take extra overhead, but also seeks the optimal IS and is enough for ablation.

I. Experiment details

In this section, we provide details of our experiment settings. Our network, hyperparameters and training are the same for different noise schedules on the same dataset.

Model architectures Our diffusion ODEs are parameterized in terms of the γ-timed normalized velocity predictor $\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}_\gamma, \gamma)$, based on the U-Net structure of Kingma et al. (2021). This architecture is tailor-made for maximum likelihood training, employing special designs such as removing the internal downsampling/upsampling and adding Fourier features for fine-scale prediction. Our configuration for each dataset also follows Kingma et al. (2021): For CIFAR-10, we use U-Net of depth 32 with 128 channels; for ImageNet-32, we still use U-Net of depth 32, but double the number of channels to 256. All our models use a dropout rate of 0.1 in the intermediate layers.

Hyperparameters and training We follow the same default training settings as Kingma et al. (2021). For all our experiments, we use the Adam (Kingma & Ba, 2014) optimizer with learning rate 2×10^{-4}, exponential decay rates of $\beta_1 = 0.9, \beta_2 = 0.99$ and decoupled weight decay (Loshchilov & Hutter, 2019) coefficient of 0.01. We also maintain an exponential moving average (EMA) of model parameters with an EMA rate of 0.9999 for evaluation.

For other hyperparameters, we use fixed start and end times which satisfy $\gamma_\epsilon = -13.3, \gamma_T = 5.0$, which is the default setting in Kingma et al. (2021). In the finetuning stage, we simply set the coefficient λ in the mixed loss $\mathcal{J}_{\text{FM}}(\theta) + \lambda \mathcal{J}_{\text{FM,tr}}(\theta)$ as 0.1 without no further tuning, so that the magnitude of the second-order loss is negligible w.r.t the first-order loss. Since the first-order matching accuracy is critical to the second-order matching, a large λ will make the training unstable or even degenerate the likelihood performance.

All our training processes are conducted on 8 GPU cards of NVIDIA A40 expect for ImageNet-32 (old version). For CIFAR-10, we pretrain the model for 6 million iterations, which takes around 3 weeks. Then we finetune the model for 200k iterations, which takes around 1 day. For ImageNet-32 (new version), we pretrain the model for 2 million iterations, which takes around 2 weeks. Then we finetune the model for 250k iterations, which takes around 3 days. We use a batch size of 128 for both training stages and both datasets.

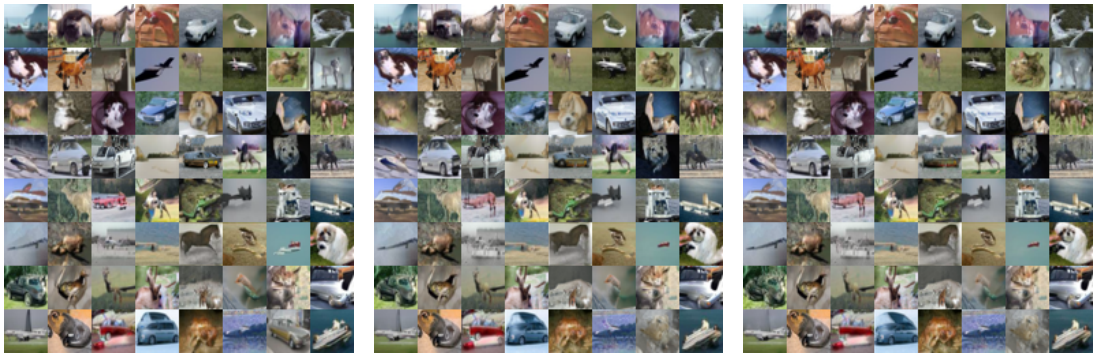
Note that in related works (Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022), experiments on ImageNet-32 (new

version) are conducted at a larger batch size (512 or 1024), which may improve the results. We did not use a larger batch size or train longer due to resource limitations.

For ImageNet-32 (old version), the training processes are conducted on 8 GPU cards of NVIDIA A100 (40GB). We pretrain the model for 2 million iterations using a batch size of 512, which takes around 2 weeks. Then we finetune the model for 500k iterations using a batch size of 128 and accumulate the gradient for every 4 batches, which takes around 2.5 days.

Likelihood and sample quality For likelihood, we use our truncated-normal dequantization. When the number of importance samples $K = 1$, we report the BPD on the test dataset with 5 times repeating to reduce the variance of the trace estimator. When $K = 5$ or $K = 20$, we do not repeat the dataset since the log-likelihood of a data sample is already evaluated multiple times. For sampling, since we are concentrated on ODE, we simply use an adaptive-step ODE solver with RK45 method (Dormand & Prince, 1980) (relative tolerance 10^{-5} and absolute tolerance 10^{-5}). We generate 50k samples and report the FIDs on them. Utilizing high-quality sampling procedures such as PC sampler (Song et al., 2021c) or fast sampling algorithms such as DPM-Solver (Lu et al., 2022b) may improve the results, which are left for future work.

J. Additional samples



(a) VDM(Kingma et al., 2021) (b) Our pretrain (c) Our pretrain+finetune

Figure 6. Random samples for ablation by ODE sampler. Our pretraining and finetuning lead to a better likelihood with small visual quality degeneration.

Figure 7. Random samples by ODE sampler (CIFAR-10, VP).

Figure 8. Random samples by ODE sampler (CIFAR-10, SP).

Figure 9. Random samples by ODE sampler (ImageNet-32, VP).

Figure 10. Random samples by ODE sampler (ImageNet-32, SP).