MAXIMUM LIKELIHOOD TRAINING OF PARAMETRIZED DIFFUSION MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Whereas the diverse variations of the diffusion model exist in image synthesis, the previous variations have not innovated the diffusing mechanism by maintaining the static linear diffusion. Meanwhile, it is intuitive that there would be a more promising diffusion pattern adapted to the data distribution. This paper introduces such adaptive and nonlinear diffusion method for the score-based diffusion models. Unlike the static and linear VE-or-VP SDEs of the previous diffusion models, our parameterized diffusion model (PDM) learns the optimal diffusion process by combining the normalizing flow and the diffusion process. Specifically, PDM utilizes the flow to transform non-linearly a data variable into a latent variable, and PDM applies the diffusion process to the transformed latent variable with the linear diffusing mechanism. Subsequently, PDM enjoys the nonlinear and learned diffusion from the perspective of the data variable. We train PDM with the variational proxy of the log-likelihood, and we prove that the variational gap between the variational bound and the log-likelihood becomes tight when the normalizing flow is optimal.

1 INTRODUCTION

The diffusion models have recently achieved the success on the sample generation, and some researches claim the state-of-the-art performance in sample generation over GAN structures (Karras et al., 2019). This particular success is highlighted in the community of the likelihood-based models, including the normalizing flows (Grcić et al., 2021), the autoregressive models (Parmar et al., 2018), and the variational autoencoders (Vahdat and Kautz, 2020). Moreover, this success is noteworthy because it is achieved merely using the linear diffusing mechanisms, such as VESDEs (Song and Ermon, 2020) and VPSDEs (Ho et al., 2020).

This paper expands these linear diffusing mechanisms of VE/VPSDE to a data-adaptive trainable nonlinear diffusion. To motivate the expansion, though there are structural similarities between the diffusion models and VAEs, the forward diffusion in the diffusion model has not been trained in existing literature while its counterpart, which is the encoder of VAE, is trainable. Rather, the current diffusion models assume the linear diffusing mechanism to be fixed throughout the training procedure. Because of this static nature of the diffusing mechanism, the variational gap (Cremer et al., 2018) between the log-likelihood and the ELBO remains to be strictly positive unless the score perfectly estimates the data score. This variational gap prevents the score training from being MLE. This gap is the fundamental motivation to develop the trainable encoder of the diffusion model, so that the gap can be tight.

As we tighten the variational gap by training a nonlinear forward diffusion, the biggest challenge comes from the intractable optimization loss. The denoising diffusion loss requires the transition probability to be closed-form in order to achieve the fast optimization, but the nonlinear diffusing mechanism, in general, has no closed-form perturbation probability. Hence, our innovation becomes designing a diffusion model with the tractable loss while the diffusion is learnable and flexible, which we achieve by merging the previous diffusion model and the tractable variable transformation.

Our innovation concentrates on the theoretic and the practical aspects of a new diffusion, which is tractable, learnable, and flexible. Theoretically, we observe that the nonlinear diffusing mechanism can be transformed to a linear diffusion under an invertible transformation, and vice versa. Practically, we implement Parametrized Diffusion Model (PDM) by merging the normalizing flow model and the

diffusion model, which constructs the tractable, learnable, and flexible nonlinear diffusing mechanism. From this construction, we prove that the variational gap can be further tightened by training the forward diffusion nonlinearly. Also, we demonstrate the state-of-the-art performance in CelebA on FID. We summarize our key contribution as below.

- PDM expands the scope of the forward diffusion process from the linear and fixed dynamics to the nonlinear and trainable dynamics.
- PDM minimizes the variational gap between the log-likelihood and ELBO, lower than the gap of the previous models because the forward diffusion parameter of PDM is optimized.
- PDM largely remedies the training instability of the diffusion model by adjusting the score scales in the latent space.

2 PRELIMINARY

The diffusion model is constructed with the bidirectional stochastic processes. The *forward* direction diffuses the input variable to a noise variable, and the *reverse* direction denoises the random noise to construct the realistic input data. The diffusion model learns this reverse direction (generative process) by estimating the data score. We provide the brief summary of the diffusion model, as below.

Forward Diffusion At the beginning of model build-up, the diffusion process requires the diffusing mechanism on the input variable. This paper assumes that the diffusion process is governed by the SDE of $\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\,\mathrm{d}t + \mathbf{G}(\mathbf{x}_t, t)\,\mathrm{d}\boldsymbol{\omega}_t$ with $\mathbf{x}_0 \sim p_r$, where p_r is the distribution from a real-world dataset and $\mathbf{x}_t$ is the solution of the SDE.

Reverse Diffusion The theory of the stochastic calculus guarantees that we could create the identical diffusion process, $\{\mathbf{x}_t\}_{t=0}^T$, by solving an associated reverse SDE backwards in time (Anderson, 1982). The associated reverse SDE is

$$\mathrm{d}\mathbf{x}_t = \big[\mathbf{f}(\mathbf{x}_t, t) - \mathbf{G}(\mathbf{x}_t, t)\mathbf{G}^T(\mathbf{x}_t, t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\big]\,\mathrm{d}t + \mathbf{G}(\mathbf{x}_t, t)\,\mathrm{d}\bar{\boldsymbol{\omega}}_t, \quad \mathbf{x}_T \sim p_T, \tag{1}$$

where $\boldsymbol{\omega}_t$ and $\bar{\boldsymbol{\omega}}_t$ are the standard Wiener processes with the time flows forward and backward, respectively; and p_t is the probability law of $\mathbf{x}_t$.

Generative Diffusion The diffusion model approximates the above SDE 1 to eventually yield the estimation on the data distribution of p_r. In SDE 1, the previous literature setup that the drift and the diffusion terms, $\mathbf{f}$ and $\mathbf{G}$, are determined a-priori in the forward diffusion. However, the data score, $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$, is intractable to compute, so we estimate this data score with the score network of $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ in order to mimic the reverse diffusion with our generative process. This score network approximates the reverse diffusion by plugging the estimated score in place of the data score with the below generative process:

$$\mathrm{d}\mathbf{x}_t^{\boldsymbol{\theta}} = \big[\mathbf{f}(\mathbf{x}_t^{\boldsymbol{\theta}}, t) - \mathbf{G}(\mathbf{x}_t^{\boldsymbol{\theta}}, t)\mathbf{G}^T(\mathbf{x}_t^{\boldsymbol{\theta}}, t)\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\boldsymbol{\theta}}, t)\big]\,\mathrm{d}t + \mathbf{G}(\mathbf{x}_t^{\boldsymbol{\theta}}, t)\,\mathrm{d}\bar{\boldsymbol{\omega}}_t, \quad \mathbf{x}_T^{\boldsymbol{\theta}} \sim \pi. \tag{2}$$

The generative process starts from the prior distribution (π), and it constructs the time-continuous random variables $\mathbf{x}_t^{\boldsymbol{\theta}}$ by solving the SDE 2 backwards in time. The generated stochastic process is denoted by $\{\mathbf{x}_t^{\boldsymbol{\theta}}\}_{t=0}^T$, and we omit $\boldsymbol{\theta}$ in the superscript if no confusion arises. With this generative process, we define the generative distribution, $\mathbf{x}_0^{\boldsymbol{\theta}} \sim p_{\boldsymbol{\theta}}$, as the probability density of the generated random variable at time $t = 0$.

Score Estimation We train the score network by the variational bound of the log-likelihood, given by

$$\begin{aligned}\mathbb{E}_{\mathbf{x}_0}\big[-\log p_{\boldsymbol{\theta}}(\mathbf{x}_0)\big] \leq& \mathcal{L}\big(\{\mathbf{x}_t\}_{t=0}^T, \lambda; \boldsymbol{\theta}\big) - \mathbb{E}_{\mathbf{x}_T}\big[\log \pi(\mathbf{x}_T)\big] \\ =& \int_0^T \lambda(t)\mathcal{L}_t\big(\{\mathbf{x}_t\}_{t=0}^T; \boldsymbol{\theta}\big)\,\mathrm{d}t - \mathbb{E}_{\mathbf{x}_T}\big[\log \pi(\mathbf{x}_T)\big],\end{aligned} \tag{3}$$

where $\mathcal{L}_t\big(\{\mathbf{x}_t\}_{t=0}^T; \boldsymbol{\theta}\big) = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t}\big[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_{0t}(\mathbf{x}_t|\mathbf{x}_0)\|_2^2\big]$ up to a constant.

3 MOTIVATION OF NONLINEAR DIFFUSING MECHANISM

Though it has long been theoretically and empirically grounded to train the encoder part in VAE, such solid ground is not accomplished in the diffusion models. This section analyzes the structural similarity between the VAE model and the diffusion model, which brings the foundation of the data-adaptive nonlinear diffusing mechanism.

Table 1: Comparison of variational gap between the VAE, the diffusion model, and PDM.

Models	Diffusion	Stochastic Process	Variational Gap (=NLL−NELBO)
VAE	No	$\{\mathbf{x}, \mathbf{z}\}$	$\mathbb{E}_{\mathbf{x}}\big[D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}\vert\mathbf{x})\Vert p_{\boldsymbol{\theta}}(\mathbf{z}\vert\mathbf{x}))\big]$
Diffusion Models	Data space	$\{\mathbf{x}_t\}_{t=0}^T$	$\mathbb{E}_{\mathbf{x}_0}\big[D_{KL}(\boldsymbol{\mu}(\{\mathbf{x}_t\}\vert\mathbf{x}_0)\Vert\boldsymbol{\nu}_{\boldsymbol{\theta}}(\{\mathbf{x}_t\}\vert\mathbf{x}_0))\big]$
PDM	Latent space	$\{\mathbf{x}_t\}_{t=0}^T/\{\mathbf{z}_t\}_{t=0}^T$	$\mathbb{E}_{\mathbf{z}_0}\big[D_{KL}\big(\boldsymbol{\mu}_{\boldsymbol{\phi}}^l(\{\mathbf{z}_t\}\vert\mathbf{z}_0)\Vert\boldsymbol{\nu}_{\boldsymbol{\theta}}^l(\{\mathbf{z}_t\}\vert\mathbf{z}_0)\big)\big]$

3.1 Variational Gap of VAE

Given the ELBO of VAE as

$$
\begin{aligned}
-\log p_{\boldsymbol{\theta}}(\mathbf{x}) \leq & \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\big[-\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\big] + D_{KL}\big(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big) && (4)\\
= & -\log p_{\boldsymbol{\theta}}(\mathbf{x}) + D_{KL}\big(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\big), && (5)
\end{aligned}
$$

we have a pair of interpretations on this ELBO. First, when we focus on Eq. 4, the ELBO 1) aids the data reconstruction by optimizing $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\big[-\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\big]$ and 2) regularizes the inference distribution of the encoder to a prior distribution by $D_{KL}\big(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\big)$. On the other hand, if we concentrate on Eq. 5, ELBO bounds the log-likelihood by approximating the intractable decoder posterior of $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ with the tractable encoder posterior of $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. The vanilla VAE (Kingma and Welling, 2013) assumes this approximate posterior to be the Gaussian distribution with the mean and the diagonal covariance through amortized inference: $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mathbf{z}; \mu_{\boldsymbol{\phi}}(\mathbf{x}), \sigma_{\boldsymbol{\phi}}^2(\mathbf{x})\mathbf{I}\big)$. By expanding the flexibility of this approximate posterior into the variational family of general distributions, for instance, Rezende and Mohamed (2015) resulted in the tighter ELBO, which eventually leads the optimization of VAE more close to MLE, and their choice of the generalizable model was the normalizing flow.

To connect this VAE ELBO to the diffusion loss, we restate this ELBO in the language of a stochastic process. Having that VAE attains the stochastic process of bivariate random variables, $\{\mathbf{x}, \mathbf{z}\}$, the ELBO is reformulated to the KL divergence between the two joint distributions modeled in bidirectional ways:

$$
\begin{aligned}
D_{KL}(p_r\|p_{\boldsymbol{\theta}}) \leq & D_{KL}(p_r\|p_{\boldsymbol{\theta}}) + \mathbb{E}_{p_r(\mathbf{x})}\big[D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))\big]\\
= & \mathbb{E}_{p_r(\mathbf{x})}\left[\log\frac{p_r(\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{x})} + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right]\\
= & \mathbb{E}_{p_r(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\frac{p_r(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{x})p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\\
= & D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{x},\mathbf{z})\|p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})),
\end{aligned} \tag{6}
$$

The inequality 6, i.e., $D_{KL}(p_r\|p_{\boldsymbol{\theta}}) \leq D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{x},\mathbf{z})\|p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}))$, is well-known by itself in the field of information theory (Duchi, 2016) by the name of the data processing inequality. This restated bound interprets that VAE is essentially a bimodeling approach of the joint distribution on the bivariate stochastic process, $\{\mathbf{x}, \mathbf{z}\}$. On the forward direction ($\mathbf{x} \to \mathbf{z}$), the latent variable is conditioned on the data variable, and the joint distribution is modeled by $q_{\boldsymbol{\phi}}(\mathbf{x},\mathbf{z}) = p_r(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. On the reverse direction ($\mathbf{z} \to \mathbf{x}$), the generative data variable is conditioned on the latent variable, and the joint distribution is modeled by $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}) = p(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$. Under this equivalent framework, we present an analytic tool to measure the closeness of the Negative ELBO (NELBO) and the Negative Log-Likelihood (NLL) as in Table 1 by

$$
\begin{aligned}
\text{Gap}(q_{\boldsymbol{\phi}}, p_{\boldsymbol{\theta}}) = & \text{NELBO} - \text{NLL}\\
= & D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{x},\mathbf{z})\|p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})) - D_{KL}(p_r\|p_{\boldsymbol{\theta}})\\
= & \mathbb{E}_{p_r(\mathbf{x})}\big[D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))\big],
\end{aligned}
$$

and we denote this quantity as the variational gap (Cremer et al., 2018). Though it is one of the central tasks in VAE to minimize this variational gap (Kingma et al., 2016), only limited works (Burda et al., 2015; Neal, 2001) have quantitatively estimated this variational gap due to its computational burden. We introduce the variational gap in the next section of the diffusion model, where the gap is tractably computed.

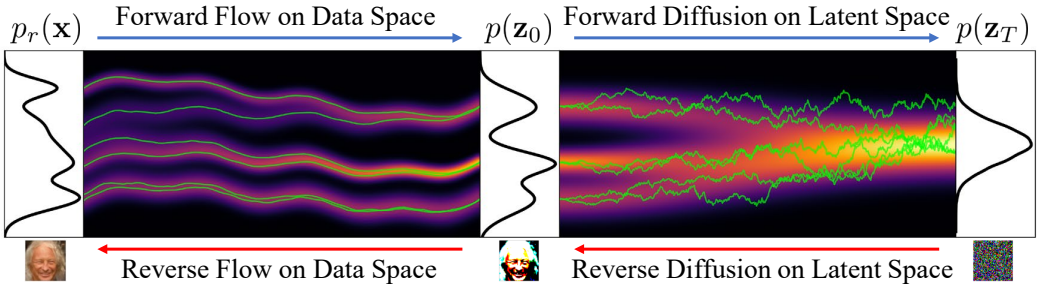


Figure 1: Schematic view of PDM. The normalizing flow transforms the data variable to the latent variable, and the linear diffusion such as VESDE or VPSDE diffuses the latent variable to a prior distribution. Data generation starts from the prior distribution, denoises the randomly generated noise to a latent, and transforms back the latent to the data space.

3.2 Variational Gap of Diffusion Models

This section constructs the analogy of the diffusion model to VAE. To begin with, we observe that the classical data processing inequality in VAE holds on the stochastic process with bivariate random variables on a finite-dimensional Euclidean space. On the other hand, the time horizon of the diffusion model is continuous, so it contains an infinite number of random variables in its stochastic process as $\{\mathbf{x}_t\}_{t=0}^T$. Fortunately, the finite-dimensional classical data processing inequality on the bivariate random variables is directly extendable to the stochastic process with a continuous index. To be concrete, the infinite-variate joint distribution is called the path measure (Oksendal, 2013), which is defined as the probability law on the space of continuous functions induced by $\{\mathbf{x}_t\}_{t=0}^T$. Therefore, the path measure is an alternative concept of the joint distribution for the stochastic process with a continuous index. The path measure for the forward diffusion in the diffusion model corresponds to the joint distribution for the encoder part in VAE, and similarly, the path measure for the generative diffusion corresponds to the decoder part in VAE. If $\boldsymbol{\mu}$ is the forward (inference) path measure and $\boldsymbol{\nu}_{\boldsymbol{\theta}}$ is the generative path measure; then the data processing inequality becomes

$$D_{KL}(p_r \| p_{\boldsymbol{\theta}}) \leq D_{KL}(\boldsymbol{\mu} \| \boldsymbol{\nu}_{\boldsymbol{\theta}}).$$

The real distribution of p_r is the marginal distribution of $\boldsymbol{\mu}$ at $t = 0$, and the generative distribution of $p_{\boldsymbol{\theta}}$ becomes the marginal distribution of $\boldsymbol{\nu}_{\boldsymbol{\theta}}$ at $t = 0$. Under this concept, the data processing inequality means that the distributional divergence of $D_{KL}(\boldsymbol{\mu} \| \boldsymbol{\nu}_{\boldsymbol{\theta}})$ between the path measures bounds the divergence of $D_{KL}(p_r \| p_{\boldsymbol{\theta}})$ between the *sliced* path measures at $t = 0$.

Previous works (Song et al., 2021; Huang et al., 2021) regard this variational bound being equivalent to the diffusion loss,

$$D_{KL}(\boldsymbol{\mu} \| \boldsymbol{\nu}_{\boldsymbol{\theta}}) = \mathcal{L}\big(\{\mathbf{x}_t\}_{t=0}^T, g^2; \boldsymbol{\theta}\big) + D_{KL}(p_T \| \pi),$$

where $\mathbf{G}(\mathbf{x}_t, t) = g(t)$ is a scaler function on t, and $\lambda(t)$ is the square of the diffusion term $g^2(t)$. Since the previous diffusion models optimize the diffusion loss, the KL divergence of $D_{KL}(\boldsymbol{\mu} \| \boldsymbol{\nu}_{\boldsymbol{\theta}})$ is an equivalent target of the optimization on the diffusion loss. Subsequently, we can credit the variational gap as the training performance to tighten the NELBO to the NLL by

$$\text{Gap}(\boldsymbol{\mu}, \boldsymbol{\nu}_{\boldsymbol{\theta}}) = D_{KL}(\boldsymbol{\mu} \| \boldsymbol{\nu}_{\boldsymbol{\theta}}) - D_{KL}(p_r \| p_{\boldsymbol{\theta}}).$$

This variational gap satisfies

$$\text{Gap}(\boldsymbol{\mu}, \boldsymbol{\nu}_{\boldsymbol{\theta}}) = \mathbb{E}_{p_r(\mathbf{x}_0)}\Big[D_{KL}\big(\boldsymbol{\mu}(\{\mathbf{x}_t\}|\mathbf{x}_0) \| \boldsymbol{\nu}_{\boldsymbol{\theta}}(\{\mathbf{x}_t\}|\mathbf{x}_0)\big)\Big],$$

so the variational gap is minimized if the inference posterior, $\boldsymbol{\mu}(\{\mathbf{x}_t\}|\mathbf{x}_0)$, estimates the intractable generative posterior, $\boldsymbol{\nu}_{\boldsymbol{\theta}}(\{\mathbf{x}_t\}|\mathbf{x}_0)$. However, unlike VAE, the inference posterior is not trainable, which restricts the variational gap to be strictly positive in previous diffusion models. We note that the diffusion model provides the tractable NLL using the Instantaneous Change of Variable (Song et al., 2020), and this tractable variational gap becomes useful in analyzing the diffusion model quality.

4 Parametrized Diffusion Model

4.1 No Closed-form Transition Probability on Nonlinear Diffusion

All the suggested SDEs for the diffusion models, including VESDE and VPSDE, are categorized as the linear SDEs because they have the linear drift terms of $\mathbf{f}(\mathbf{x}_t, t)$. This drift term determines the

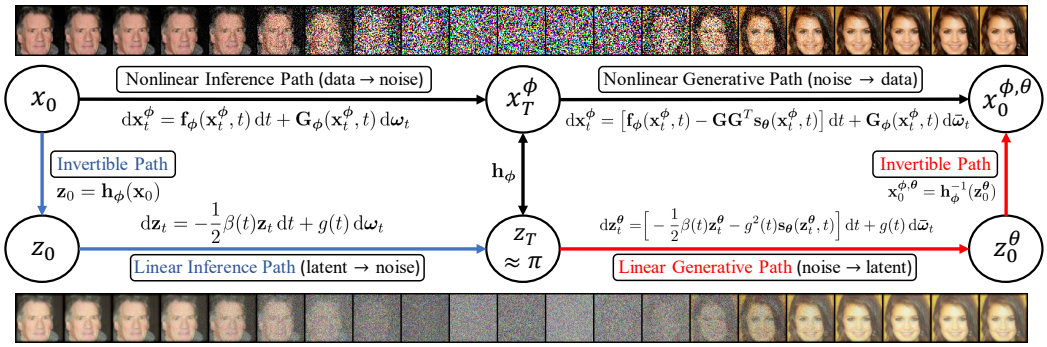


Figure 2: PDM attains the ladder structure between the data space and the latent space by the invertibility of the normalizing flow. The score network estimates the data score following the blue arrows, and the model generates the images following the red arrows.

dynamics of the diffusion, so a data-adaptive diffusion could be attained simply by parametrizing the drift and diffusion terms. However, the transition probability with the nonlinear coefficients for the SDE, $p_{0t}(\mathbf{x}_t|\mathbf{x}_0)$, is intractable in general, and the diffusion loss becomes intractable to optimize. In fact, it is worth to note that the closed-form of the transition probability exists only on certain types of SDEs like VESDE and VPSDE.

4.2 Parametrized Forward Diffusion with Normalizing Flow

To construct a tractable transition probability while keeping the nonlinearity, we observe that the nonlinear diffusion can be decomposed into the linear diffusion and the nonlinear invertible transformation, where the transformation provides the flexible nonlinearity on the diffusion process. Figure 1 and 2 provide the schematic view of our model. The normalizing flow transforms the data input to a latent variable, and the diffusion model linearly permeates the latent variable to the noise distribution.

Concretely, let us define $\mathbf{z}_0$ to be the transformed latent variable, $\mathbf{z}_0 = \mathbf{h}_\phi(\mathbf{x}_0)$, where ϕ is the flow model parameter. Then, the canonical diffusion with the linear dynamics of

$$\mathrm{d}\mathbf{z}_t = -\frac{1}{2}\beta(t)\mathbf{z}_t\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{\omega}_t, \quad \mathbf{z}_0 = \mathbf{h}_\phi(\mathbf{x}_0) \text{ with } \mathbf{x}_0 \sim p_r,$$

describes the diffusing mechanism of the latent variable. Since the transformation is invertible, if we define the inverse random variable by $\mathbf{x}_t^\phi = \mathbf{h}_\phi^{-1}(\mathbf{z}_t)$, the dynamics of the stochastic process of $\{\mathbf{x}_t^\phi\}_{t=0}^T$ is governed by the SDE of

$$\mathrm{d}\mathbf{x}_t^\phi = \mathbf{f}_\phi(\mathbf{x}_t^\phi, t)\,\mathrm{d}t + \mathbf{G}_\phi(\mathbf{x}_t^\phi, t)\,\mathrm{d}\boldsymbol{\omega}_t, \quad \mathbf{x}_0^\phi \sim p_r,$$

where the drift and the diffusion terms are derived by

$$\begin{cases} \mathbf{f}_\phi(\mathbf{x}_t, t) & = -\frac{1}{2}\beta\left[\nabla\mathbf{h}_\phi\right]^{-1}\mathbf{h}_\phi + \frac{1}{2}g^2\mathrm{tr}\left(\nabla^2\mathbf{h}_\phi^{-1}\right) \\ \mathbf{G}_\phi(\mathbf{x}_t, t) & = g\left[\nabla\mathbf{h}_\phi\right]^{-1}, \end{cases}$$

from the Ito's lemma (Oksendal, 2013). See full derivation and explanation in Appendix B. This diffusion process is an *induced diffusion* on the data space that is defined indirectly by transforming back from the latent space to the data space. The nonlinearity of this induced diffusion fully depends on the invertible transformation as illustrated in Figure 2.

It is not straightforward to directly construct the generative reverse process on the data space, $\mathbf{x}_t^{\phi,\theta}$. Figures 1 and 2 build the generative process by red arrows using the transformation. Starting from the prior distribution of π, we denoise the random variable from the prior to $\mathbf{z}_t^\theta$ according to the generative process of the linear diffusion on the latent space, and we transform the fully denoised latent of $\mathbf{z}_0^\theta$ back to the data space $\mathbf{x}_0^{\phi,\theta} = \mathbf{h}_\phi^{-1}(\mathbf{z}_0^\theta)$.

4.3 Tractable Variational Bound of Parametrized Diffusion Model

The joint modeling of the normalizing flow and the diffusion model enables the tractable training of the nonlinear dynamics. The log-likelihood on the data space reduces to the log-likelihood on the

latent space by adding the log determinant of Jacobian to the latent log-likelihood. This calculation of log-likelihoods provides Theorem 1 that bounds the log-likelihood with the tractable ELBO.

Theorem 1. *Suppose that $p_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}_0)$ is the log-likelihood of the generative random variable $\mathbf{x}_0^{\boldsymbol{\phi},\boldsymbol{\theta}}$. Then, the log-likelihood is bounded by*

$$\mathbb{E}_{p_r(\mathbf{x}_0)}\big[-\log p_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}_0)\big] \leq \mathcal{L}\big(\{\mathbf{x}_t\}_{t=0}^T, g^2; \{\boldsymbol{\phi},\boldsymbol{\theta}\}\big),$$

where

$$\mathcal{L}\big(\{\mathbf{x}_t\}_{t=0}^T, g^2; \{\boldsymbol{\phi},\boldsymbol{\theta}\}\big) = -\mathbb{E}_{p_r(\mathbf{x}_0)}\Big[\log\Big|\det\Big(\frac{\partial \mathbf{h}_{\boldsymbol{\phi}}}{\partial \mathbf{x}_0}\Big)\Big|\Big] + \mathcal{L}\big(\{\mathbf{z}_t\}_{t=0}^T, g^2; \boldsymbol{\theta}\big) - \mathbb{E}_{\mathbf{z}_T}\big[\log \pi(\mathbf{z}_T)\big], \tag{7}$$

with $\mathcal{L}\big(\{\mathbf{z}_t\}_{t=0}^T, g^2; \boldsymbol{\theta}\big) = \int_0^T g^2(t)\mathbb{E}_{\mathbf{z}_0,\mathbf{z}_t}\big[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t) - \nabla_{\mathbf{z}_t}\log p_{0t}(\mathbf{z}_t|\mathbf{z}_0)\|_2^2\big]\,\mathrm{d}t$, up to a constant.

Compared to the vanilla diffusion ELBO in Eq. 3, the PDM ELBO in Eq. 7 differs in three aspects. First, the normalizing flow loss, i.e., the log determinant of Jacobian, is added in the loss. This counts the burden of transforming the data into the latent space. Second, the remaining terms related to the diffusion model count the linear diffusion on the latent space with linear dynamics that is tractably optimizable, rather than the intractable data diffusion of nonlinear dynamics. Third, the optimization loss in Eq. 7 estimates the score on the latent space with the score network of $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)$, while the diffusion loss in Eq. 3 estimates the score on the data space with the score network of $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$.

4.4 Optimization Losses

Previous works (Ho et al., 2020; Song et al., 2021; Vahdat et al., 2021) have empirically demonstrated that training the diffusion model with the likelihood weighting, $\lambda(t) = g^2(t)$, could bring worse performances than applying other weighting functions. For instance, Ho et al. (2020); Song and Ermon (2020) introduce the variance weighting function, $\lambda(t) = \sigma^2(t)$, and Ho et al. (2020) show that this weighting outperforms the likelihood weighting in terms of the sample generation.

Within our framework, we also observe that the variance weighting function is advantageous over the likelihood weighting. Henceforth, we optimize the flow model and the diffusion model with different losses as follows:

$$\text{Flow:} \quad \min_{\boldsymbol{\phi}} \mathcal{L}(\{\mathbf{x}_t\}_{t=0}^T, g^2; \{\boldsymbol{\phi},\boldsymbol{\theta}\})$$

$$\text{Diffusion:} \quad \min_{\boldsymbol{\theta}} \mathcal{L}(\{\mathbf{z}_t\}_{t=0}^T, \{g^2 \text{ or } \sigma^2\}; \boldsymbol{\theta}),$$

Note that the first term of the log Jacobian and the third term of the cross entropy in Eq. 7 are independent of the score parameters, $\boldsymbol{\theta}$. Therefore, the diffusion parameters are not influenced by these terms, in which only the second term remains to update the diffusion parameters.

4.5 Tight Variational Gap of Parametrized Forward Diffusion

This section theoretically analyzes the variational gap of PDM. To construct the variational gap of PDM, we enumerate few nomenclatures for the path measures in Table 2. On the data space, $\boldsymbol{\mu}_{\boldsymbol{\phi}}^d$ is the path measure for the *inference* process and $\boldsymbol{\nu}_{\boldsymbol{\phi},\boldsymbol{\theta}}^d$ is the path measure for the *generative* process. On the latent space, $\boldsymbol{\mu}_{\boldsymbol{\phi}}^l$ is the path measure for the *inference* process and $\boldsymbol{\nu}_{\boldsymbol{\theta}}^l$ is the path measure for the *generative* process. Here, $\boldsymbol{\nu}_{\boldsymbol{\theta}}^l$ does not depend on $\boldsymbol{\phi}$ because the generative diffusion on the latent space starts from the $\boldsymbol{\phi}$-independent noise distribution (π) and the diffusion proceeds the denoising process by following the score estimation of $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)$ that depends only on $\boldsymbol{\theta}$.

Table 2: List of path measures.

Space (Diffusion)	Measure	Type
Data (Nonlinear)	$\boldsymbol{\mu}_{\boldsymbol{\phi}}^d$	Inference
	$\boldsymbol{\nu}_{\boldsymbol{\phi},\boldsymbol{\theta}}^d$	Generative
Latent (Linear)	$\boldsymbol{\mu}_{\boldsymbol{\phi}}^l$	Inference
	$\boldsymbol{\nu}_{\boldsymbol{\theta}}^l$	Generative

The variational gap on the data space becomes

$$\mathrm{Gap}_d(\boldsymbol{\mu}_{\boldsymbol{\phi}}^d, \boldsymbol{\nu}_{\boldsymbol{\phi},\boldsymbol{\theta}}^d) = \text{NELBO} - \text{NLL} \tag{8}$$

$$= D_{KL}(\boldsymbol{\mu}_{\boldsymbol{\phi}}^d \| \boldsymbol{\nu}_{\boldsymbol{\phi},\boldsymbol{\theta}}^d) - D_{KL}(p_r \| p_{\boldsymbol{\phi},\boldsymbol{\theta}}), \tag{9}$$

where the subscript in Gap_d represents the data space. Recall that training the encoder in VAE guarantees the tighter variational gap because the encoder distribution of $q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z})$ approaches toward the decoder distribution of $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ by minimizing the NELBO of $D_{KL}\big(q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}) \| p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\big)$. However, this analogy is not valid in the variational gap in Eq. 8 because both inference and generative path measures depend on $\boldsymbol{\phi}$. This means that the generative path measure could arbitrarily deviate from the inference path measure when we optimize $\boldsymbol{\phi}$.

Using the change of variable, however, we derive an equivalent formulation of the above variational gap by

$$\begin{aligned}\text{Gap}_d(\boldsymbol{\mu}^d_{\boldsymbol{\phi}}, \boldsymbol{\nu}^d_{\boldsymbol{\phi},\boldsymbol{\theta}}) =& D_{KL}(\boldsymbol{\mu}^d_{\boldsymbol{\phi}} \| \boldsymbol{\nu}^d_{\boldsymbol{\phi},\boldsymbol{\theta}}) - D_{KL}(p_r \| p_{\boldsymbol{\phi},\boldsymbol{\theta}}) \\ =& D_{KL}(\boldsymbol{\mu}^l_{\boldsymbol{\phi}} \| \boldsymbol{\nu}^l_{\boldsymbol{\theta}}) - D_{KL}(p_{\boldsymbol{\phi}} \| p_{\boldsymbol{\theta}}) \\ =& \text{Gap}_l(\boldsymbol{\mu}^l_{\boldsymbol{\phi}}, \boldsymbol{\nu}^l_{\boldsymbol{\theta}}).\end{aligned}$$

With this derivation, optimizing the invertible transformation guarantees the tighter variational gap because the generative measure of $\boldsymbol{\nu}^l_{\boldsymbol{\theta}}$ is now fixed on $\boldsymbol{\phi}$. Also, we note that this variational gap reduces to the KL divergence between the inference posterior and the generative posterior by

$$\text{Gap}_d(\boldsymbol{\mu}^d_{\boldsymbol{\phi}}, \boldsymbol{\nu}^d_{\boldsymbol{\phi},\boldsymbol{\theta}}) = \text{Gap}_l(\boldsymbol{\mu}^l_{\boldsymbol{\phi}}, \boldsymbol{\nu}^l_{\boldsymbol{\theta}}) = \mathbb{E}_{\mathbf{z}_0}\big[D_{KL}\big(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\{\mathbf{z}_t\}|\mathbf{z}_0) \| \boldsymbol{\nu}_{\boldsymbol{\theta}}(\{\mathbf{z}_t\}|\mathbf{z}_0)\big)\big], \tag{10}$$

which provides another analogy to VAE. We enumerate the variational gaps of the explained models in Table 1. In fact, we show in Theorem 2 that this variational gap converges to zero as we train the forward nonlinear diffusion on the data space. See Appendix A for the proofs.

Theorem 2. *Suppose the variational family is flexible enough to cover any path measure. For any $\boldsymbol{\theta}$, if $\nabla.\mathbf{s}_{\boldsymbol{\theta}}(\cdot, t)$ is symmetric for any $t \in [0, T]$ and $\nabla. \log \pi(\cdot) \equiv \mathbf{s}_{\boldsymbol{\theta}}(\cdot, T)$, then there exists $\boldsymbol{\phi}^*$ such that $\text{Gap}_d(\boldsymbol{\mu}^d_{\boldsymbol{\phi}^*}, \boldsymbol{\nu}^d_{\boldsymbol{\phi}^*,\boldsymbol{\theta}}) = 0$. Furthermore, we have*

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \text{Gap}_d(\boldsymbol{\mu}^d_{\boldsymbol{\phi}}, \boldsymbol{\nu}^d_{\boldsymbol{\phi},\boldsymbol{\theta}}) = \arg\min_{\boldsymbol{\phi}} D_{KL}(\boldsymbol{\mu}^d_{\boldsymbol{\phi}} \| \boldsymbol{\nu}^d_{\boldsymbol{\phi},\boldsymbol{\theta}}) = \arg\min_{\boldsymbol{\phi}} D_{KL}(p_r \| p_{\boldsymbol{\phi},\boldsymbol{\theta}}).$$

5 Related Work

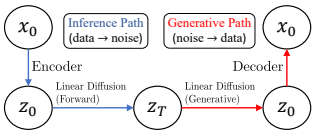


Figure 3: LSGM.

Recently, the diffusion model has been jointly modeled in conjunction with other deep generative models. There are two major previous works on this line of research. First, Vahdat et al. (2021) (LSGM) in Figure 3 applies the diffusion model to the latent space of VAE. Since VAE does not attain the invertible structure, LSGM is not suggesting the expansion of the diffusing mechanism. In other words, LSGM cannot provide the diffused data learning at the middle of the process because LSGM does not induce the diffusion process on the data space due to the lack of VAE's invertibility. Consequently, the analyzing tool for LSGM should be the variational gap of VAE, rather than that of the diffusion model.

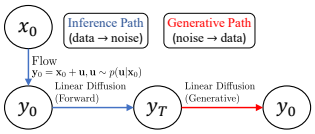


Figure 4: ScoreFlow.

Second, Song et al. (2021) (ScoreFlow) in Figure 4 simply concatenate the normalizing flow on top of the diffusion model to provide the dequantized data distribution only at the initial diffusion time. Despite of the structural similarity of ScoreFlow and PDM, the major difference of ScoreFlow and PDM lies on the differences between their density model construction: $p_{\boldsymbol{\theta}}$ in Scoreflow and $p_{\boldsymbol{\phi},\boldsymbol{\theta}}$ in PDM. ScoreFlow constructs $p_{\boldsymbol{\theta}}$ only with the diffusion model, so the flow and the diffusion networks are not jointly combined in the loss function of $\int p_r(\mathbf{x}) \int_{[0,1)^d} q_{\boldsymbol{\phi}}(\mathbf{u}|\mathbf{x}) \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}+\mathbf{u})}{q_{\boldsymbol{\phi}}(\mathbf{u}|\mathbf{x})} \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x}$, where $\mathbf{u}$ is the dequantization variable that lies in a unit cube, $[0, 1)^d$. In other words, ScoreFlow adopts the flow *outside* of the model density, so the flow is independent of the model density. In fact, ScoreFlow can be best understood as a dequantization technique (Ho et al., 2019), whose goal is estimating the optimal dequantizing region on $[0, 1)^d$ to minimize the dequantization gap (Hoogeboom et al., 2020). Therefore, the variational gap of ScoreFlow remains to be strictly positive because training the flow does not optimize the forward diffusing mechanism. On the other hand, PDM constructs the flow-embedded model density of $p_{\boldsymbol{\phi},\boldsymbol{\theta}}$, and PDM optimizes the loss of $\int p_r(\mathbf{x}) \log \int_{[0,1)^d} p_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x} + \mathbf{u}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x}$ that puts the normalizing flow *inside* of the model density. Although we train PDM with the uniform dequantization (Theis et al., 2015) by default, we could apply the variational dequantization to PDM as well.

Table 3: Performance comparison on CIFAR-10.

SDE	Name	Model	NLL	NELBO	Gap	FID
VE	NCSN++	Diffusion Baseline	3.45	4.43	0.98	**2.20**
	PDM (FID)	Diffusion & Flow	**3.04**	**3.09**	**0.05**	3.36
VP	DDPM++	Diffusion Baseline	3.13	3.29	0.16	3.16
	LSGM (FID)	Diffusion & VAE	-	3.43	-	**2.10**
	PDM (FID)	Diffusion & Flow	**3.04**	**3.09**	**0.05**	6.79
	PDM (FID, dec)	Diffusion & Flow-v2	3.11	3.17	0.06	4.43
	DDPM++ (IS)	Diffusion Baseline	**2.91**	**3.10**	0.19	**5.70**
	PDM (NLL, dec)	Diffusion & Flow-v2	3.13	3.15	**0.02**	5.75
	ResFlow	Flow Backbone	3.28	-	-	46.37
	NVAE	VAE Backbone	-	2.96	-	43.18

Table 4: Performance comparison on CelebA.

Model	NLL	NELBO	Gap	FID
NCSN++ (VE)	2.39	3.96	1.57	3.95
PDM (VE, FID)	**2.00**	**2.09**	**0.09**	**2.50**
DDPM++ (VP)	**1.79**	2.19	0.40	3.03
PDM (VP, FID)	2.04	**2.10**	**0.06**	**2.23**
CR-NVAE	-	1.86	-	-
DenseFlow-74-10	1.99	-	-	-
GDDIM	-	-	-	2.92
Styleformer	-	-	-	3.66

6 EXPERIMENTS

This section quantitatively and qualitatively analyzes the suggested PDM on CIFAR-10 (Krizhevsky et al., 2009) and CelebA (Liu et al., 2015) 64×64. We compute the NLL for the density estimation in bits per dimension scale by solving the probability flow (Song et al., 2020) backwards in time, and we compare the baseline performances of Song et al. (2020; 2021) computed on the uniform dequantization. For the sample generation, we use the Predictor-Corrector algorithm (Song et al., 2020) for PDM on VESDE, and we use the numerical ODE sampler of the probability flow (Song et al., 2020) with RK45 method on VPSDE. Throughout the experimental results, we use NCSN++ (VE) and DDPM++ (VP) (Song et al., 2020) as the backbones of the diffusion model, and a pair of ResNet-based flow models (Chen et al., 2019; Ma et al., 2020) as the backbones of the flow model. We give a full description on models for our experiments in Appendix C. We name PDM (FID) and PDM (NLL) for the experiments that use weighting functions as σ^2 and g^2, respectively.

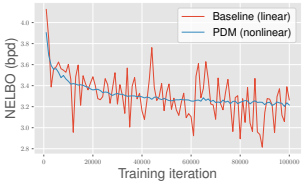
6.1 QUANTITATIVE RESULTS

We compare PDM to the baselines in Table 3 on VESDE and VPSDE as the linear diffusing mechanisms. Although minor performance degradation on the sample performance, Table 3 empirically demonstrates the reduced variational gap. In all experimental settings, the variational gaps significantly drop less than 0.1. On the other hand, the variational gaps of the baseline diffusion on VPSDE for the FID model and the NLL model are 0.16 and 0.19, respectively. We conclude that PDM effectively decreases the variational gap by training the parametrized inference path measure.

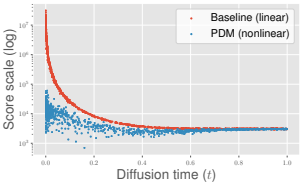
Table 4 compares the experiments on CelebA to the current state-of-the-art baseline models in various disciplines: the diffusion model of GDDIM (Nachmani et al., 2021), the VAE model of CR-NVAE (Sinha and Dieng, 2021), the flow model of DenseFlow (Grcić et al., 2021), and the GAN model of Styleformer (Park and Kim, 2021). We observe a pair of implications. First, training the diffusing mechanism with PDM largely reduces the variational gap. In particular, PDM reduces this inference gap from 1.57 to 0.09 on the NCSN++ backbone with VESDE and from 0.40 to 0.05 on the DDPM++ backbone with VPSDE. Also, we emphasize that the NELBO of PDM largely outperforms the NELBO of the baseline models. Second, our PDM reports the state-of-the-art performance with 2.23 FID score in the CelebA sample generation. This opposite trend on FID (compared to the results in CIFAR-10) indicates that the nonlinear diffusion is more effective on datasets of higher dimensions. In fact, this is a quite intuitive result because as the data dimension increases, the family of feasible diffusion expands exponentially, and the data-optimal diffusing mechanism is more likely to deviate from linear diffusions.

6.2 QUALITATIVE RESULTS

Test NELBO To figure out the effect of the nonlinear diffusing mechanism on the training procedure, Figure 5 (a) shows the test NELBO by training iteration on the diffusion model with 35M parameters. Without any modification on the hyperparameters, we train PDM with the shallow ResFlow (Chen et al., 2019) of 3M parameters, and Figure 5 (a) shows that the training on PDM surprisingly stabilizes the test NELBO. Figure 5 (b) illustrates that such training stability can be attributed to the nearly identical score scales. As described in Figure 5 (b), the score scale on the data space is unbounded as the diffusion time goes to zero, and this extreme scale variation eventually leads to the training instability on the baseline diffusion model. On the other hand, adopting the flow transformation on the data space mitigates such extreme scale variations that stabilize the training.



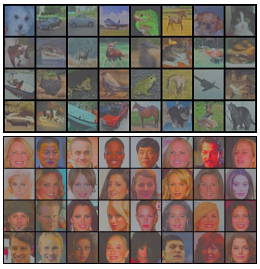
(a) Test NELBO



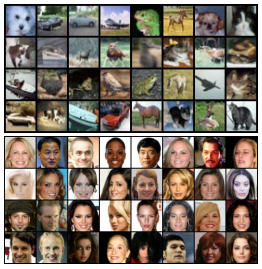
(b) Score scale

Figure 5: Comparison of the linear and nonlinear diffusing mechanisms on a shallow network.

Figure 6: Comparison of the linear diffusion model and PDM on a deep network.



(a) Samples from $\mathbf{z}_0$



(b) Samples from $\mathbf{x}_0$

Figure 7: Non cherry-picked samples from data and latent.

Figure 8: Ablation study for the flow temperature.

Figure 6 shows the test NELBO on the diffusion model with 107.6M number of parameters. To stabilize the diffusion model, we apply the AdamW optimizer (Loshchilov and Hutter, 2017) with 0.01 weight decay, and Exponential Moving Average (Song et al., 2020) of 0.9999. Using the ResFlow model with 10.6M parameters, Figure 6 indicates that PDM consistently outperforms the baseline diffusion in terms of NELBO with smaller variations.

Flow Depth The flow model architecture is the building block that fundamentally determines the family of the inference path measures. Table 5 empirically demonstrates that the inflexible variational family results in a large variational gap on CIFAR-10. Additionally, the absolute magnitude of the NLL performance keeps decreasing as we choose the deeper flow network.

Table 5: Ablation study on flow layers.

# layers	# params	NLL	NELBO	Gap
4	1.2M	3.19	3.30	0.11
8	2.3M	3.19	3.24	0.05
32	10.6M	**3.04**	**3.09**	**0.05**

Sample Generation Figure 7 presents the samples from (a) the latent space and (b) the data space. Since the sample generation on the latent space is unnormalized, we normalize the sampled latent variables to be bounded on the range of $[0, 255]$, so we can visualize the latent information in image. Figure 7 indicates that the diffusion model and the flow model have distinctive and complementary roles in sample generation: 1) the diffusion model constructs the global context and 2) the flow model enriches this global context by colorizing it into a realistic image.

Temperature As the flow colorizes the latent sample, we experiment the effect of temperature (Kingma and Dhariwal, 2018) on this colorization in Figure 8. With temperature τ, the normalizing flow puts its latent input scaled by τ to the flow network. In our PDM, the image color with a higher temperature tends to be brighter, and we find that the optimal temperature depends on the experimental settings. For instance, the optimal temperature for PDM with VESDE turns out to be 1.1 in terms of FID as in Table 6.

Table 6: Ablation study on temperature.

Temperature (τ)	FID-5k
0.6	12.13
0.7	9.95
0.8	8.75
0.9	8.13
1	7.85
1.1	**7.84**
1.2	8.11
1.3	8.61

7 Conclusion

This paper expands the linear diffusing mechanism to be nonlinear by combining the invertible transformation and the diffusion model. This nonlinear diffusing mechanism learns the forward diffusion process out of the variational family of inference path measures, and such optimization provides the tighter variational gap compared to the baseline model with linear diffusion.

8 ETHICS STATEMENT

As our work improve the sample quality, our methodology could be used in the adversarial attack by using fake images. Also, we acknowledge that the research in this direction could potentially destroy the evidence-based justice system by fooling the system with artificial generations.

9 REPRODUCIBILITY STATEMENT

Every proofs and derivations are explained in details on Appendix A and B. Also, for the reproducibility, we would release our code as soon as the discussion period opens through an anonymous repository to the reviewers. We note that most of our code is built based on the released code of Song et al. (2020) in order to compare with the baselines in a fair setting introduced in Song et al. (2020) and Song et al. (2021). Additionally, we prodive the details on training/evaluation/neural architecture in Appendix C.

REFERENCES

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

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

Chen, R. T., Behrmann, J., Duvenaud, D., and Jacobsen, J.-H. (2019). Residual flows for invertible generative modeling. *arXiv preprint arXiv:1906.02735*.

Cremer, C., Li, X., and Duvenaud, D. (2018). Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pages 1078–1086. PMLR.

Do Carmo, M. P. (2012). *Differential forms and applications*. Springer Science & Business Media.

Duchi, J. (2016). Lecture notes for statistics 311/electrical engineering 377. *URL: https://stanford.edu/class/stats311/Lectures/full_notes. pdf. Last visited on*, 2:23.

Grcić, M., Grubišić, I., and Šegvić, S. (2021). Densely connected normalizing flows. *arXiv preprint arXiv:2106.04627*.

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

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*.

Hoogeboom, E., Cohen, T. S., and Tomczak, J. M. (2020). Learning discrete distributions by dequantization. *arXiv preprint arXiv:2001.11235*.

Huang, C.-W., Lim, J. H., and Courville, A. (2021). A variational perspective on diffusion-based generative models and score matching. *arXiv preprint arXiv:2106.02808*.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410.

Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29:4743–4751.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

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

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ma, X., Kong, X., Zhang, S., and Hovy, E. (2020). Decoupling global and local representations via invertible generative flows. *arXiv preprint arXiv:2004.11820*.

Nachmani, E., Roman, R. S., and Wolf, L. (2021). Non gaussian denoising diffusion models. *arXiv preprint arXiv:2106.07582*.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and computing*, 11(2):125–139.

Nichol, A. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*.

Oksendal, B. (2013). *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media.

Park, J. and Kim, Y. (2021). Styleformer: Transformer based generative adversarial networks with style vector. *arXiv preprint arXiv:2106.07023*.

Parmar, G., Zhang, R., and Zhu, J.-Y. (2021). On buggy resizing libraries and surprising subtleties in fid calculation. *arXiv preprint arXiv:2104.11222*.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. (2018). Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Rudin, W. et al. (1964). *Principles of mathematical analysis*, volume 3. McGraw-hill New York.

Sinha, S. and Dieng, A. B. (2021). Consistency regularization for variational auto-encoders. *arXiv preprint arXiv:2105.14859*.

Song, Y., Durkan, C., Murray, I., and Ermon, S. (2021). Maximum likelihood training of score-based diffusion models. *arXiv e-prints*, pages arXiv–2101.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *arXiv preprint arXiv:1907.05600*.

Song, Y. and Ermon, S. (2020). Improved techniques for training score-based generative models. *arXiv preprint arXiv:2006.09011*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*.

Theis, L., Oord, A. v. d., and Bethge, M. (2015). A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*.

Vahdat, A. and Kautz, J. (2020). Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*.

Vahdat, A., Kreis, K., and Kautz, J. (2021). Score-based generative modeling in latent space. *arXiv preprint arXiv:2106.05931*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

A PROOFS OF THEOREM 1 AND THEOREM 2

Theorem 1. *Suppose that $p_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}_0)$ is the log-likelihood of the generative random variable $\mathbf{x}_0^{\boldsymbol{\phi},\boldsymbol{\theta}}$. Then, the log-likelihood is bounded by*

$$\mathbb{E}_{p_r(\mathbf{x}_0)}\big[-\log p_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}_0)\big] \leq \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}\big(\{\mathbf{x}_t\}_{t=0}^T; g^2\big),$$

where

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}\big(\{\mathbf{x}_t\}_{t=0}^T; g^2\big) = &-\mathbb{E}_{p_r(\mathbf{x}_0)}\Big[\log\Big|\det\Big(\frac{\partial \mathbf{h}_{\boldsymbol{\phi}}}{\partial \mathbf{x}_0}\Big)\Big|\Big] + \mathcal{L}_{\boldsymbol{\theta}}\big(\{\mathbf{z}_t\}_{t=0}^T; g^2\big) - \mathbb{E}_{\mathbf{z}_T}\big[\log \pi(\mathbf{z}_T)\big] \\ &+ \frac{d}{2}\int_0^T \beta(t) - \frac{g^2(t)}{\sigma^2(t)}\,\mathrm{d}t,\end{aligned}$$

with $\sigma^2(t) = 1 - e^{-\int_0^t \beta(s)\,\mathrm{d}s} + \int_0^t [g^2(s) - \beta(s)]\,\mathrm{d}s$.

Proof of Theorem 1. From the change of variable, the transformation of $\mathbf{z}_0 = \mathbf{h}_{\boldsymbol{\phi}}(\mathbf{x}_0)$ induces

$$p_r(\mathbf{x}_0) = \frac{p_0(\mathbf{z}_0)}{\Big|\det\Big(\frac{\partial \mathbf{h}_{\boldsymbol{\phi}}}{\partial \mathbf{x}_0}\Big)\Big|^{-1}},$$

and thus the entropy of the data distribution becomes

$$\begin{aligned}\mathcal{H}(p_r) &= -\int p_r(\mathbf{x}_0)\log p_r(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0 \\ &= -\int p_0(\mathbf{z}_0)\log\frac{p_0(\mathbf{z}_0)}{\Big|\det\Big(\frac{\partial \mathbf{h}_{\boldsymbol{\phi}}}{\partial \mathbf{x}_0}\Big)\Big|^{-1}}\,\mathrm{d}\mathbf{z}_0 \\ &= -\int p_r(\mathbf{x}_0)\log\Big|\det\Big(\frac{\partial \mathbf{h}_{\boldsymbol{\phi}}}{\partial \mathbf{x}_0}\Big)\Big|\,\mathrm{d}\mathbf{x}_0 - \int p_0(\mathbf{z}_0)\log p_0(\mathbf{z}_0)\,\mathrm{d}\mathbf{z}_0 \\ &= -\mathbb{E}_{p_r(\mathbf{x}_0)}\Big[\log\Big|\det\Big(\frac{\partial \mathbf{h}_{\boldsymbol{\phi}}}{\partial \mathbf{x}_0}\Big)\Big|\Big] - \int p_0(\mathbf{z}_0)\log p_0(\mathbf{z}_0)\,\mathrm{d}\mathbf{z}_0 \\ &= -\mathbb{E}_{p_r(\mathbf{x}_0)}\Big[\log\Big|\det\Big(\frac{\partial \mathbf{h}_{\boldsymbol{\phi}}}{\partial \mathbf{x}_0}\Big)\Big|\Big] + \mathcal{H}(p_0).\end{aligned}$$

From Theorem 4 of Song et al. (2021), the entropy at $t = 0$ equals to

$$\mathcal{H}(p_0) = \mathcal{H}(p_T) - \frac{1}{2}\int_0^T \mathbb{E}_{p_t(\mathbf{z}_t)}\big[2\nabla_{\mathbf{z}_t}\cdot\mathbf{f}(\mathbf{z}_t,t) + g^2(t)\|\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\|_2^2\big]\,\mathrm{d}t,$$

where $\mathbf{f}(\mathbf{z}_t, t)$ is the drift term of the diffusion for $\mathbf{z}_t$ and p_t is the probability distribution of $\mathbf{z}_t$. Therefore, the negative log-likelihood becomes

$$\begin{aligned}-\mathbb{E}_{p_r(\mathbf{x}_0)}\big[\log p_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}_0)\big] &= D_{KL}(p_r\|p_{\boldsymbol{\phi},\boldsymbol{\theta}}) + \mathcal{H}(p_r) \\ &\leq D_{KL}(\boldsymbol{\mu}_{\boldsymbol{\phi}}^d\|\boldsymbol{\nu}_{\boldsymbol{\phi},\boldsymbol{\theta}}^d) + \mathcal{H}(p_r) \\ &= D_{KL}(\boldsymbol{\mu}_{\boldsymbol{\phi}}^d\|\boldsymbol{\nu}_{\boldsymbol{\phi},\boldsymbol{\theta}}^d) - \mathbb{E}_{p_r(\mathbf{x}_0)}\Big[\log\Big|\det\Big(\frac{\partial \mathbf{h}_{\boldsymbol{\phi}}}{\partial \mathbf{x}_0}\Big)\Big|\Big] + \mathcal{H}(p_0) \\ &= D_{KL}(\boldsymbol{\mu}_{\boldsymbol{\phi}}^l\|\boldsymbol{\nu}_{\boldsymbol{\theta}}^l) - \mathbb{E}_{p_r(\mathbf{x}_0)}\Big[\log\Big|\det\Big(\frac{\partial \mathbf{h}_{\boldsymbol{\phi}}}{\partial \mathbf{x}_0}\Big)\Big|\Big] + \mathcal{H}(p_T) \\ &\quad -\frac{1}{2}\int_0^T \mathbb{E}_{\mathbf{z}_t^{\boldsymbol{\phi}}}\big[-d\beta(t) + g^2(t)\|\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\|_2^2\big]\,\mathrm{d}t.\end{aligned}$$

Now, from Theorem 1 of Song et al. (2021), the KL-divergence between the path measures becomes

$$D_{KL}(\boldsymbol{\mu}_{\boldsymbol{\phi}}^l\|\boldsymbol{\nu}_{\boldsymbol{\theta}}^l) = D_{KL}(p_T\|\pi) + \frac{1}{2}\int_0^T g^2(t)\mathbb{E}_{p_t(\mathbf{z}_t)}\big[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t) - \nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\|_2^2\big]\,\mathrm{d}t,$$

so if we plug in this into the negative log-likelihood, we yield the following:

$$
\begin{aligned}
&-\mathbb{E}_{p_r(\mathbf{x}_0)}\big[\log p_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}_0)\big] \\
&\quad= -\mathbb{E}_{p_r(\mathbf{x}_0)}\Big[\log\Big|\det\Big(\frac{\partial \mathbf{h}_{\boldsymbol{\phi}}}{\partial \mathbf{x}_0}\Big)\Big|\Big] + \frac{1}{2}\int_0^T g^2(t)\mathbb{E}_{\mathbf{z}_t}\big[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t) - \nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\|_2^2\big]\,\mathrm{d}t \\
&\qquad + D_{KL}(p_T\|\pi) + \mathcal{H}(p_T) - \frac{1}{2}\int_0^T \mathbb{E}_{\mathbf{z}_t}\big[-d\beta(t) + g^2(t)\|\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\|_2^2\big]\,\mathrm{d}t \\
&\quad= -\mathbb{E}_{p_r(\mathbf{x}_0)}\Big[\log\Big|\det\Big(\frac{\partial \mathbf{h}_{\boldsymbol{\phi}}}{\partial \mathbf{x}_0}\Big)\Big|\Big] - \mathbb{E}_{\mathbf{z}_T}\big[\log\pi(\mathbf{z}_T)\big] + \frac{d}{2}\int_0^T \beta(t)\,\mathrm{d}t \\
&\qquad + \frac{1}{2}\int_0^T g^2(t)\mathbb{E}_{\mathbf{z}_t}\big[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t) - \nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\|_2^2 - \|\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\|_2^2\big]\,\mathrm{d}t
\end{aligned}
$$

Also, we have

$$
\begin{aligned}
\mathbb{E}_{\mathbf{z}_t}\big[\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t)\cdot\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\big] &= \int p_t(\mathbf{z}_t)\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t)\cdot\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\,\mathrm{d}\mathbf{z}_t \\
&= \int \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t)\cdot\nabla_{\mathbf{z}_t} p_t(\mathbf{z}_t)\,\mathrm{d}\mathbf{z}_t \\
&= \int \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t)\cdot\int p_0(\mathbf{z}_0)\nabla_{\mathbf{z}_t} p_{0t}(\mathbf{z}_t|\mathbf{z}_0)\,\mathrm{d}\mathbf{z}_0\,\mathrm{d}\mathbf{z}_t \\
&= \mathbb{E}_{\mathbf{z}_0}\mathbb{E}_{\mathbf{z}_t|\mathbf{z}_0}\big[\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t)\cdot\nabla_{\mathbf{z}_t}\log p_{0t}(\mathbf{z}_t|\mathbf{z}_0)\big]
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\frac{1}{2}\int_0^T g^2(t)\mathbb{E}_{\mathbf{z}_t}\big[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t) - \nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\|_2^2 - \|\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\|_2^2\big] \\
&\quad= \int_0^T g^2(t)\mathbb{E}_{\mathbf{z}_t}\Big[\frac{1}{2}\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t)\|_2^2 - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t)\cdot\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)\Big] \\
&\quad= \int_0^T g^2(t)\mathbb{E}_{\mathbf{z}_0}\mathbb{E}_{\mathbf{z}_t|\mathbf{z}_0}\Big[\frac{1}{2}\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t)\|_2^2 - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t)\cdot\nabla_{\mathbf{z}_t}\log p_{0t}(\mathbf{z}_t|\mathbf{z}_0)\Big] \\
&\quad= \frac{1}{2}\int_0^T g^2(t)\mathbb{E}_{\mathbf{z}_0}\mathbb{E}_{\mathbf{z}_t|\mathbf{z}_0}\Big[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t) - \nabla_{\mathbf{z}_t}\log p_{0t}(\mathbf{z}_t|\mathbf{z}_0)\|_2^2 - \|\nabla_{\mathbf{z}_t}\log p_{0t}(\mathbf{z}_t|\mathbf{z}_0)\|_2^2\Big].
\end{aligned}
$$

Now, since $p_{0t}(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t;\mu(t)\mathbf{z}_t,\sigma^2(t)\mathbf{I})$ for $\mu(t)$ and $\sigma^2(t)$ determined by $\beta(t)$ and $g(t)$, we have

$$
\mathbb{E}_{\mathbf{z}_t|\mathbf{z}_0}\big[\|\nabla_{\mathbf{z}_t}\log p_{0t}(\mathbf{z}_t|\mathbf{z}_0)\big] = \mathbb{E}_{\mathbf{z}_t|\mathbf{z}_0}\Big[\Big\|\frac{\mathbf{z}_t - \mu(t)\mathbf{z}_0}{\sigma^2(t)}\Big\|_2^2\Big] = \mathbb{E}_{\mathcal{N}(\mathbf{z};0,\mathbf{I})}\Big[\frac{\|\mathbf{z}\|_2^2}{\sigma^2(t)}\Big] = \frac{d}{\sigma^2(t)},
$$

and we have the desired result. □

Theorem 2. *Suppose the variational family is flexible enough to cover any path measure. For any $\boldsymbol{\theta}$, if $\nabla.\mathbf{s}_{\boldsymbol{\theta}}(\cdot,t)$ is symmetric for any $t\in[0,T]$ and $\nabla.\log\pi(\cdot)\equiv\mathbf{s}_{\boldsymbol{\theta}}(\cdot,T)$, then there exists $\boldsymbol{\phi}^*$ such that* $\mathrm{Gap}_d(\boldsymbol{\mu}^d_{\boldsymbol{\phi}^*},\boldsymbol{\nu}^d_{\boldsymbol{\phi}^*,\boldsymbol{\theta}}) = 0$. *Furthermore, we have*

$$
\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}}\mathrm{Gap}_d(\boldsymbol{\mu}^d_{\boldsymbol{\phi}},\boldsymbol{\nu}^d_{\boldsymbol{\phi},\boldsymbol{\theta}}) = \arg\min_{\boldsymbol{\phi}} D_{KL}(\boldsymbol{\mu}^d_{\boldsymbol{\phi}}\|\boldsymbol{\nu}^d_{\boldsymbol{\phi},\boldsymbol{\theta}}) = \arg\min_{\boldsymbol{\phi}} D_{KL}(p_r\|p_{\boldsymbol{\phi},\boldsymbol{\theta}}).
$$

Proof of Theorem 2. The symmetry of $\nabla.\mathbf{s}_{\boldsymbol{\theta}}(\cdot,t)$ indicates that $\mathbf{s}_{\boldsymbol{\theta}}(\cdot,t)$ is a closed 1-form. Therefore, from the Poincare's lemma (Do Carmo, 2012), $\mathbf{s}_{\boldsymbol{\theta}}(\cdot,t)$ is exact, and there exists a scalar field $f^{\boldsymbol{\theta}}_t$ that satisfies $\mathbf{s}_{\boldsymbol{\theta}}(\cdot,t) = \nabla.f^{\boldsymbol{\theta}}_t(\cdot)$ for any t. If we define by $q^{\boldsymbol{\theta}}_t(\mathbf{z}_t) = \frac{\exp(f^{\boldsymbol{\theta}}_t(\mathbf{z}_t))}{\int\exp(f^{\boldsymbol{\theta}}_t(\mathbf{z}))\,\mathrm{d}\mathbf{z}}$, then we have $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t,t) = \nabla_{\mathbf{z}_t}\log q^{\boldsymbol{\theta}}_t(\mathbf{z}_t,t)$ for any t. With this construction, the reverse SDE on the transformed latent space becomes

$$
\mathrm{d}\mathbf{z}^{\boldsymbol{\theta}}_t = \Big[-\frac{1}{2}\beta(t)\mathbf{z}^{\boldsymbol{\theta}}_t - g^2(t)\nabla_{\mathbf{z}^{\boldsymbol{\theta}}_t}\log q^{\boldsymbol{\theta}}_t(\mathbf{z}^{\boldsymbol{\theta}}_t)\Big]\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{\boldsymbol{\omega}}_t,\quad \mathbf{z}^{\boldsymbol{\theta}}_T\sim\pi = q^{\boldsymbol{\theta}}_T.
$$

If $\boldsymbol{\rho_\theta}$ is the path measure of the solution of the above SDE, then the marginal distribution of $\boldsymbol{\rho_\theta}$ at t would be $q_t^{\boldsymbol{\theta}}$. Therefore, if $\mathbf{h}_{\boldsymbol{\phi}^*}^{-1}$ maps p_r to $q_0^{\boldsymbol{\theta}}$, then the forward process on the latent space $\{\mathbf{z}_t^{\boldsymbol{\phi}^*}\}_{t=0}^T$ becomes the generative process on the latent space $\{\mathbf{z}_t^{\boldsymbol{\theta}}\}_{t=0}^T$. Thus, we have $\boldsymbol{\mu}_{\boldsymbol{\phi}^*}^l = \boldsymbol{\nu}_{\boldsymbol{\theta}}^l$ in distribution, which leads $D_{KL}(\boldsymbol{\mu}_{\boldsymbol{\phi}^*}^l \| \boldsymbol{\nu}_{\boldsymbol{\theta}}^l) = 0$. We get the desired result by the change of variable formula: $D_{KL}(\boldsymbol{\mu}_{\boldsymbol{\phi}^*}^d \| \boldsymbol{\nu}_{\boldsymbol{\phi}^*,\boldsymbol{\theta}}^d) = D_{KL}(\boldsymbol{\mu}_{\boldsymbol{\phi}^*}^l \| \boldsymbol{\nu}_{\boldsymbol{\theta}}^l) = 0$. □

B Derivation of Nonlinear Data Diffusion Process

Throughout this section, we omit $\boldsymbol{\phi}$ for the notational simplicity. Beginning with the linear diffusion on the latent space given by

$$\mathrm{d}\mathbf{z}_t = -\frac{1}{2}\beta(t)\mathbf{z}_t\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{\omega}_t, \quad \mathbf{z}_0 = \mathbf{h}(\mathbf{x}_0) \text{ with } \mathbf{x}_0 \sim p_r, \tag{11}$$

we have defined $\mathbf{x}_t = \mathbf{h}^{-1}(\mathbf{z}_t)$ as the induced random variables. Then by the multivariate Ito's Lemma (Oksendal, 2013), the k-th component of the induced variable satisfies

$$\mathrm{d}\mathrm{x}_{t,k} = \frac{\partial \mathrm{h}_k^{-1}}{\partial t}\,\mathrm{d}t + \left[\nabla_{\mathbf{z}_t}\mathrm{h}_k^{-1}(\mathbf{z}_t)\right]^T \mathrm{d}\mathbf{z}_t + \frac{1}{2}\mathrm{tr}\left(\nabla_{\mathbf{z}_t}^2 \mathrm{h}_k^{-1}(\mathbf{z}_t)\,\mathrm{d}\mathbf{z}_t\,\mathrm{d}\mathbf{z}_t^T\right) \tag{12}$$

Plugging the linear diffusion SDE 11 in the SDE of Eq. 12, Eq. 12 is derived by

$$\begin{aligned}\mathrm{d}\mathrm{x}_{t,k} &= \left[\nabla_{\mathbf{z}_t}\mathrm{h}_k^{-1}(\mathbf{z}_t)\right]^T \left\{-\frac{1}{2}\beta(t)\mathbf{z}_t\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{\omega}_t\right\} + \frac{1}{2}g^2(t)\mathrm{tr}\left(\nabla_{\mathbf{z}_t}^2 \mathrm{h}_k^{-1}(\mathbf{z}_t)\right)\mathrm{d}t \\ &= \left\{-\frac{1}{2}\beta(t)\left[\nabla_{\mathbf{z}_t}\mathrm{h}_k^{-1}(\mathbf{z}_t)\right]^T\mathbf{z}_t + \frac{1}{2}g^2(t)\mathrm{tr}\left(\nabla_{\mathbf{z}_t}^2 \mathrm{h}_k^{-1}(\mathbf{z}_t)\right)\right\}\mathrm{d}t + g(t)\left[\nabla_{\mathbf{z}_t}\mathrm{h}_k^{-1}(\mathbf{z}_t)\right]^T \mathrm{d}\boldsymbol{\omega}_t,\end{aligned} \tag{13}$$

because $\frac{\partial \mathrm{h}_k^{-1}}{\partial t} = 0$. Then, Eq. 13 in vector form becomes

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)\,\mathrm{d}t + \mathbf{G}(\mathbf{x}_t, t)\,\mathrm{d}\boldsymbol{\omega}_t,$$

where the vector-valued drift and diffusion terms are given by

$$\begin{cases} \mathbf{f}(\mathbf{x}_t, t) = -\frac{1}{2}\beta(t)\nabla_{\mathbf{z}_t}\mathbf{h}^{-1}(\mathbf{z}_t)\mathbf{z}_t + \frac{1}{2}g^2(t)\mathrm{tr}\left(\nabla_{\mathbf{z}_t}^2\mathbf{h}^{-1}(\mathbf{z}_t)\right) \\ \mathbf{G}(\mathbf{x}_t, t) = g(t)\nabla_{\mathbf{z}_t}\mathbf{h}^{-1}(\mathbf{z}_t), \end{cases} \tag{14}$$

where $\nabla_{\mathbf{z}_t}^2\mathbf{h}^{-1}(\mathbf{z}_t)$ is a 3-dimensional tensor with (i,j,k)-th element to be $\left(\nabla_{\mathbf{z}_t}^2 \mathrm{h}_k^{-1}(\mathbf{z}_t)\right)_{i,j}$, and the trace operator applied on this tensor results in the vector of $\left[\mathrm{tr}\left(\nabla_{\mathbf{z}_t}^2 \mathrm{h}_1^{-1}(\mathbf{z}_t)\right), ..., \mathrm{tr}\left(\nabla_{\mathbf{z}_t}^2 \mathrm{h}_d^{-1}(\mathbf{z}_t)\right)\right]^T$. From the inverse function theorem (Rudin et al., 1964), the Jacobian of the inverse function $\nabla_{\mathbf{z}_t}\mathbf{h}^{-1}(\mathbf{z}_t)$ equals to the inverse Jacobian $\left[\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)\right]^{-1}$. Therefore, Eq. 14 is transformed to

$$\begin{cases} \mathbf{f}(\mathbf{x}_t, t) = -\frac{1}{2}\beta(t)\left[\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)\right]^{-1}\mathbf{h}(\mathbf{x}_t) + \frac{1}{2}g^2(t)\mathrm{tr}\left(\nabla_{\mathbf{z}_t}^2\mathbf{h}^{-1}(\mathbf{z}_t)\right) \\ \mathbf{G}(\mathbf{x}_t, t) = g(t)\left[\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)\right]^{-1}. \end{cases} \tag{15}$$

Now, we derive the second term of $\mathbf{f}$ in terms of $\mathbf{x}_t$ as follows: observe that $\sum_k \frac{\partial \mathrm{h}_i^{-1}}{\partial \mathrm{z}_{t,k}}\frac{\partial \mathrm{h}_k}{\partial \mathrm{x}_{t,j}} = \delta_{i,j}$, where $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise. Differentiating both sides with respect to $\mathrm{x}_{t,l}$, we have

$$\sum_k \left\{\frac{\partial}{\partial \mathrm{x}_{t,l}}\left(\frac{\partial \mathrm{h}_i^{-1}}{\partial \mathrm{z}_{t,k}}\right)\right\}\frac{\partial \mathrm{h}_k}{\partial \mathrm{x}_{t,j}} + \frac{\partial \mathrm{h}_i^{-1}}{\partial \mathrm{z}_{t,k}}\left\{\frac{\partial}{\partial \mathrm{x}_{t,l}}\left(\frac{\partial \mathrm{h}_k}{\partial \mathrm{x}_{t,j}}\right)\right\} = 0,$$

where the first term is

$$\sum_{k,m}\frac{\partial \mathrm{h}_m}{\partial \mathrm{x}_{t,l}}\left\{\frac{\partial}{\partial \mathrm{z}_{t,m}}\left(\frac{\partial \mathrm{h}_i^{-1}}{\partial \mathrm{z}_{t,k}}\right)\right\}\frac{\partial \mathrm{h}_k}{\partial \mathrm{x}_{t,j}} = \sum_{k,m}\left(\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)\right)_{l,m}^T\left(\nabla_{\mathbf{z}_t}^2\mathbf{h}_i^{-1}(\mathbf{z}_t)\right)_{m,k}\left(\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)\right)_{k,j},$$

using the chain rule, and the second term becomes

$$\sum_k \frac{\partial \mathrm{h}_i^{-1}}{\partial \mathrm{z}_{t,k}}\left\{\frac{\partial}{\partial \mathrm{x}_{t,l}}\left(\frac{\partial \mathrm{h}_k}{\partial \mathrm{x}_{t,j}}\right)\right\} = \sum_k \left(\nabla_{\mathbf{z}_t}\mathbf{h}^{-1}(\mathbf{z}_t)\right)_{i,k}\left(\nabla_{\mathbf{x}_t}^2\mathbf{h}_k(\mathbf{x}_t)\right)_{l,j}.$$

From above, we derive the trace term of $\mathbf{f}$ in Eq. 15 as

$$\text{tr}\left(\nabla^2_{\mathbf{z}_t}\mathbf{h}^{-1}(\mathbf{z}_t)\right) = -\text{tr}\left(\left(\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)^{-1}\right)^T\left(\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)^{-1} * \nabla^2_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)\right)\left(\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)^{-1}\right)\right),$$

where $\nabla^2_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)$ is a 3-dimensional tensor with (i,j,k)-th element to be $\left(\nabla^2_{\mathbf{x}_t}\mathbf{h}_k(\mathbf{x}_t)\right)_{i,j}$. Also, we define $*$ operation as the element-wise matrix multiplication given by

$$\left(\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)^{-1} * \nabla^2_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)\right)_{i,j} := \nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)^{-1}\left(\nabla^2_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)\right)_{i,j}.$$

Combining all together, thus, we derive the nonlinear drift term in Eq. 15 as a function of $\mathbf{x}_t$ given by

$$\begin{aligned}\mathbf{f}(\mathbf{x}_t,t) = &-\frac{1}{2}\beta(t)\left[\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)\right]^{-1}\mathbf{h}(\mathbf{x}_t)\\ &-\frac{1}{2}g^2(t)\text{tr}\left(\left(\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)^{-1}\right)^T\left(\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)^{-1} * \nabla^2_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)\right)\left(\nabla_{\mathbf{x}_t}\mathbf{h}(\mathbf{x}_t)^{-1}\right)\right).\end{aligned}$$

C Implementation Details

C.1 Model Architecture

Diffusion Model We implement two diffusion models as backbone: NCSN++ (VE) (Song et al., 2020) and DDPM++ (VP) (Song et al., 2020), where two backbones are one of the best performers in CIFAR-10 dataset. In our setting, NCSN++ assumes the score network with parametrization of $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{z}_t, \log\sigma^2(t))$, where $\sigma^2(t) = \sigma^2_{min}(\frac{\sigma_{max}}{\sigma_{min}})^{2t}$ is the variance of the transition probability $p_{0t}(\mathbf{z}_t|\mathbf{z}_0)$ with VESDE. As introduced in Song et al. (2020), we use the Gaussian Fourier embeddings (Tancik et al., 2020) to model the high frequency details across the temporal embedding. DDPM++ models the score network with parametrization of $\epsilon_{\boldsymbol{\theta}}(\mathbf{z}_t, t)$, which targets to estimate $-\sigma(t)\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)$. We use the Transformer sinusoidal temporal embedding (Vaswani et al., 2017).

We use the U-Net (Ronneberger et al., 2015) architecture for the score networks on both NCSN++ and DDPM++ based on (Ho et al., 2020). We stack U-Net resblocks of up-and-down convolutions with skip connections that connect the identical-dimensional layers. Also, we follow Ho et al. (2020) by applying the global attention at the resolution of 16×16. We use four U-Net resblocks with four feature map resolutions (32×32 to 4×4). There are eight and four resblocks at each resolution on CIFAR-10 and CelebA, respectively.

Flow Model We implement two normalizing flow models as backbone: the ResFlow (Chen et al., 2019) and the decoupled ResFlow (Ma et al., 2020). Although Ma et al. (2020) makes use of the Glow (Kingma and Dhariwal, 2018) for their flow model, we replace this glow architecture with the ResFlow, mainly because the ResFlow transforms the variables from the identity function. We name the experiments on this decoupled ResFlow as PDM (FID, dec) and PDM (NLL, dec). Otherwise, we train the model with ResFlow as default. For the ResFlow, we drop three components from the original paper: 1) the activation normalization Kingma and Dhariwal (2018), 2) the batch normalization Ioffe and Szegedy (2015), and 3) the fully connected layers.

On the deep ResFlow, we use resolutions of 32×32 and 16×16 on both CIFAR-10 and CelebA. Otherwise, we use a single resolution of 32×32 on other experiments whose models have networks with layers less than 32. We apply the LipSwish activation (Chen et al., 2019) that provides the Lipschitz coefficients less than a unity. For the multi-GPU training, we use the Neumann log-determinant gradient estimator, instead of the memory efficient estimator (Chen et al., 2019).

C.2 Experimental details

Training All the models are trained with batch size of 128 and EMA rate of 0.999. We train each model for around 10 days on four V-100 GPUs with 128Gb GPU memory for all experiments. At the initial training phase, we train the diffusion model parameters with fixed flow parameters initialized by the identity until the sample generation performance saturates. After this pretraining phase, we start updating both diffusion and flow models. In this learning phase, we apply the learning rate scheduling after the sample performance saturation. For the diffusion model, we drop the learning rate from 2×10^{-4} to 10^{-5}. For the flow model, we drop the learning rate from 10^{-3} to 10^{-5}

for VPSDE and 5×10^{-5} to 10^{-5} for VESDE. We emphasize this learning rate scheduling takes additional performance boost.

VESDE and VPSDE have different training details. VESDE assumes $\sigma_{min} = 10^{-2}$ on all experiments. VESDE has $\sigma_{max} = 50$ on CIFAR-10 and $\sigma_{max} = 90$ on CelebA. On the other hand, VPSDE assumes $\beta(t) = \beta_{min} + (\beta_{max} - \beta_{min})t$ with $\beta_{min} = 0.1$ and $\beta_{max} = 20$. Both VESDE and VPSDE truncate the diffusion time on $[\epsilon, T]$ in order to stabilize the diffusion model, where $\epsilon = 10^{-5}$ and $T = 1$. On sample generation, however, DDPM++/PDM with VPSDE draws samples by proceeding the generative diffusion process up to $\epsilon = 10^{-3}$, which corresponds to the diffusion variance to be $\sigma_{min} = 10^{-2}$. On the sample generation, we apply the Predictor-Corrector (PC) sampler Song et al. (2020) for NCSN++/PDM with VESDE and the Probability Flow (PF) ODE sampler Song et al. (2020) for DDPM++/PDM. Throughout the experiments, we provide the identical diffusion model structures to compare the baseline model and the PDM model in a fair setting.

Evaluation While we use the EMA rates of 0.999 for both diffusion and flow models for training, we use this EMA only for the diffusion model on the evaluation mode because the flow model without the EMA on the evaluation gives better samples. One peculiarity occurs in the experiment on the decoupled ResFlow. We empirically find that turning off the EMA for both diffusion model and flow model on the evaluation mode provides better performances of 3.11 in test NLL and 3.17 in test NELBO as reported in Table 3 on PDM (FID, dec). With the EMA applied to the diffusion model, we have 3.15 for the test NLL and 3.22 for the test NELBO, which shows slight degradation on the density estimation performance.

We compute the FID score based on Song et al. (2020) for CIFAR-10 with the modified Inception V1 network[1] in order to compare PDM to the baselines (Song et al., 2020; 2021) in a fair setting. On the other hand, for the CelebA dataset, we compute the clean-FID (Parmar et al., 2021) that provides consistently antialiased results.

C.3 Variance Reduction

Flow Training When we train the flow network with $\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}\big(\{\mathbf{x}_t\}_{t=0}^T; g^2\big)$, this NELBO contains the integration of $\mathcal{L}_{\boldsymbol{\theta}}\big(\{\mathbf{z}_t\}_{t=0}^T; g^2\big)$. However, previous works on the diffusion models (Nichol and Dhariwal, 2021) show that the estimation variance is largely reduced with the importance sampling, which leads the performance variations (Song et al., 2021). Concretely, the importance sampling chooses an importance weight that is proportional to $\frac{g^2(t)}{\sigma^2(t)}$, and estimates the integration by $\mathcal{L}_{\boldsymbol{\theta}}\big(\{\mathbf{z}_t\}_{t=0}^T; g^2\big) = \int_0^T g^2(t)\mathcal{L}_t\big(\{\mathbf{z}_t\}_{t=0}^T; \boldsymbol{\theta}\big)\,\mathrm{d}t \approx \sum_{n=1}^N \sigma^2(t_n)\mathcal{L}_{t_n}\big(\{\mathbf{z}_t\}_{t=0}^T; \boldsymbol{\theta}\big)$, where t_n is sampled from the importance distribution.

For VESDE, it satisfies $\beta(t) = 0$ and $g(t) = \sigma_{min}(\frac{\sigma_{max}}{\sigma_{min}})^t\sqrt{2\log(\frac{\sigma_{max}}{\sigma_{min}})}$. Throughout the experiments, we select $\sigma_{min} = 0.01$, and, $\sigma_{max} = 50, 90$ for CIFAR-10 and CelebA, respectively. The transition probability becomes $p_{0t}(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; \mathbf{z}_0, \sigma^2(t))$, where $\sigma^2(t) = \int_0^t g^2(s)\,\mathrm{d}s = \sigma_{min}^2\big[(\frac{\sigma_{max}}{\sigma_{min}})^{2t} - 1\big]$. Song et al. (2020) approximates this variance to be $\sigma_{app}^2(t) = \sigma_{min}^2(\frac{\sigma_{max}}{\sigma_{min}})^{2t}$ so that the variance is proportional to $g^2(t)$. Therefore, the importance weight follows the uniform distribution, and the importance sampling is equivalent with choosing the uniform t.

On the other hand, VPSDE satisfies $\beta(t) = \beta_{min} + (\beta_{max} - \beta_{min})t$ with $g(t) = \sqrt{\beta(t)}$. Then, the transition probability becomes $p_{0t}(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; \mu(t)\mathbf{z}_t, \sigma^2(t)\mathbf{I})$, where $\mu(t) = e^{-\frac{1}{2}\int_0^t \beta(s)\,\mathrm{d}s}$ and $\sigma^2(t) = 1 - e^{-\int_0^t \beta(s)\,\mathrm{d}s}$. Thus, VPSDE has the importance weight of $\frac{g^2(t)}{\sigma^2(t)} = \frac{\beta(t)}{1-e^{-\int_0^t \beta(s)\,\mathrm{d}s}}$.

The Monte-Carlo sample from this importance weight is the solution of the inverse Cumulative Distribution Function (CDF) of the importance distribution as

$$t = F^{-1}(u) \tag{16}$$

$$\iff u = F(t) = \frac{1}{Z}\int_\epsilon^t \frac{g^2(s)}{\sigma^2(s)}\,\mathrm{d}s = \frac{1}{Z}\big(\mathcal{F}(t) - \mathcal{F}(\epsilon)\big), \tag{17}$$

[1] https://tfhub.dev/tensorflow/tfgan/eval/inception/1

Table 7: FID-5k of PDM (VE, FID) on CIFAR-10.

		SNR			
		0.14	0.145	0.15	0.155
Temperature	0.5	16.59	15.99	15.95	15.7
	0.6	12.61	12.13	12.17	11.99
	0.7	10.41	9.95	10.08	9.94
	0.8	9.2	8.75	8.94	8.84
	0.9	8.58	8.13	8.35	8.28
	1.0	8.31	7.85	8.07	8.06
	1.1	8.27	**7.84**	8.01	8.07
	1.2	8.49	8.11	8.21	8.31
	1.3	8.97	8.61	8.67	8.84
	1.4	9.76	9.44	9.45	9.73
	1.5	10.93	10.68	10.63	11.01

Table 8: FID-1k of PDM with VPSDE on CIFAR-10.

Temperature	FID-1k		
	PDM (FID)	PDM (FID, dec)	PDM (NLL, dec)
0.5	40.09	36.84	37.78
0.6	38.28	35.31	36.30
0.7	37.10	34.36	36.46
0.8	36.28	33.77	35.03
0.9	35.87	33.44	34.82
1.0	35.57	33.24	**34.80**
1.1	35.40	**33.21**	34.93
1.2	35.28	33.24	35.19
1.3	**35.28**	33.24	35.64
1.4	35.38	33.42	36.24
1.5	35.68	33.65	37.02

where u is a uniform sample from $[0, 1]$, $\mathcal{F}(t)$ is the antiderivative of the importance weight given by $\mathcal{F}(t) = \log\left(1 - e^{-0.5t^2(\beta_{max}-\beta_{min})-t\beta_{min}}\right) + 0.5t^2(\beta_{max} - \beta_{min}) + t\beta_{min}$, and Z is the normalizing constant given by

$$
\begin{aligned}
Z =& \int_{\epsilon}^{T} \frac{g^2(t)}{\sigma^2(t)} \, \mathrm{d}t \\
=& \Big[\log\left(1 - e^{-0.5t^2(\beta_{max}-\beta_{min})-t\beta_{min}}\right) + 0.5t^2(\beta_{max} - \beta_{min}) + t\beta_{min} \Big]_{\epsilon}^{T} \\
=& \log\left(1 - e^{-0.5T^2(\beta_{max}-\beta_{min})-T\beta_{min}}\right) - \log\left(1 - e^{-0.5\epsilon^2(\beta_{max}-\beta_{min})-\epsilon\beta_{min}}\right) \\
& + 0.5(T^2 - \epsilon^2)(\beta_{max} - \beta_{min}) + (T - \epsilon)\beta_{min} \\
=& 23.86
\end{aligned}
$$

for $T = 1$ and $\epsilon = 10^{-5}$. The solution for the inverse CDF in Eq. 16 becomes

$$
\begin{aligned}
& e^{\int_0^t \beta(s)\,\mathrm{d}s} = 1 + \exp\left(Zu + \mathcal{F}(\epsilon)\right) \\
\iff & \int_0^t \beta(s)\,\mathrm{d}s = \frac{1}{2}(\beta_{max} - \beta_{min})t^2 + \beta_{min}t = \log\left(1 + \exp\left(Zu + \mathcal{F}(\epsilon)\right)\right) \\
\iff & t = \frac{-\beta_{min} + \sqrt{\beta_{min}^2 + 2(\beta_{max} - \beta_{min})\log\left(1 + \exp\left(Zu + \mathcal{F}(\epsilon)\right)\right)}}{\beta_{max} - \beta_{min}}.
\end{aligned}
$$

The variation of the Monte-Carlo diffusion time depends on the uniform sample of u.

Diffusion Training Now, on the side of training the diffusion network, as described in the main paper, selecting the weighting function heavily influences the model performances. For instance, Song et al. (2021) compare the weighting functions of g^2 and σ^2, and Song et al. (2021) conclude that the weighting function of $\lambda(t) = \sigma^2(t)$ works best for the sample generation, and $\lambda(t) = g^2(t)$ yields the best NLL performance. We train the diffusion model with $\sigma^2(t)$ for PDM (VE/VP, FID) and PDM (VP, FID, dec) models, and with $g^2(t)$ for PDM (VP, NLL, dec) model. To obtain the better samples from PDM (VP, NLL, dec), we mix the weighting function by applying σ^2 on the initial phase that trains only the diffusion model, and subsequently by applying g^2 on the latter phase for both diffusion and flow training. In PDM (VP, NLL, dec), we get FID-1k of 59.76 with only g^2, whereas 34.80 with mixed weights. On the other hand, the density estimation for g^2 and mixed weights has the opposite trend by performing PDM with g^2 as 3.03 for the NLL and 3.05 for the NELBO compared to PDM with mixed weights as 3.13 for the NLL and 3.15 for the NELBO.

D Additional Experimental Results

D.1 Effect of Signal-to-Noise and Temperature

On sample generation, we draw latent samples from either the PC sampling or the PF ODE sampling. PC sampling assumes the Signal-to-Noise Ratio (SNR) that modifies the Corrector algorithm (Song

Table 9: FID-1k of PDM (VE, FID) on CelebA.

		SNR		
		0.16	0.17	0.18
Temperature	0.8	17.37	16.88	17.8
	0.9	15.37	15.04	16.02
	1.0	14.72	14.56	15.37
	1.05	14.72	**14.53**	15.24
	1.1	15.00	14.83	15.47
	1.15	15.52	15.36	15.94
	1.2	16.3	16.14	16.71
	1.3	19.09	18.81	19.40

Table 10: FID-1k of PDM (VP, FID) on CelebA.

Temperature	FID-1k
0.8	15.60
0.9	15.27
1.0	15.04
1.05	14.95
1.1	**14.90**
1.15	14.94
1.2	15.05
1.3	15.44

et al., 2020) of the annealed Langevin dynamics (Song and Ermon, 2019). From the sampled latent variable, we scale by multiplying the temperature τ (Kingma and Dhariwal, 2018) to the latent variable. Since the diffusion model learns the global image structure, the normalizing flow only paints the color of the image, and the temperature controls the brightness of the generated image. This is a contrastive result to the previous flow models (Kingma and Dhariwal, 2018; Chen et al., 2019) that the temperature in the previous models controls both global context (such as the smoothness) and color. Therefore, we can create images with different colors by manipulating this temperature.

CIFAR-10 With these hyperparameters of the SNR and the temperatures, we find optimal values in the below Tables. Table 7 computes the FID-5k score of all combinations for (SNR, τ) in PDM (VE, FID) on CIFAR-10. Unlike the SNR of 0.16 that is previously suggested in (Song et al., 2020) for the vanilla diffusion model, we find that SNR of 0.145 consistently outperforming other values for SNR in terms of FID. Similarly, Table 7 presents that the optimal temperature is 1.1.

For VPSDE, Table 8 shows that the temperature of 1.3 performs the best for sample generation in PDM (VP, FID). For PDM (VP, FID, dec), we find that the temperature of 1.1 performs the best, and for PDM (VP, NLL, dec), the vanilla temperature of 1.0 generates the best samples.

CelebA Table 9 shows the FID-1k performance of PDM (VE, FID) on CelebA. It shows that the SNR of 0.17 proposed in (Song et al., 2020) performs the best also in our PDM. In addition, it shows that the optimal temperature is either 1.0 or 1.05. Table 10 shows the FID-1k performance of PDM (VP) on CelebA. It is clear in this table that the sample quality is the best on the temperature range of 1.05 to 1.15.

D.2 Random samples

Figure 9 and 10 show the non cherry-picked random samples from PDM (VE, FID) on CIFAR-10 and PDM (VP, FID) on CelebA, respectively.

Figure 9: Non cherry-picked random samples from CIFAR-10 trained on PDM (VE, FID).

Figure 10: Non cherry-picked random samples from CelebA trained on PDM (VP, FID).