Maximum Likelihood Training for Score-Based Diffusion ODEs by High-Order Denoising Score Matching

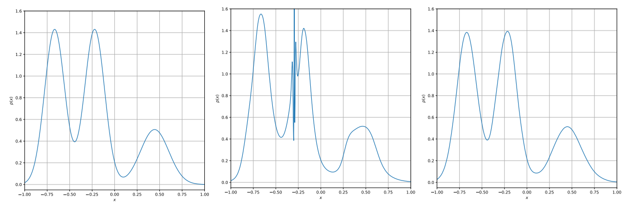
Cheng Lu [1] **Kaiwen Zheng** [1] **Fan Bao** [1] **Jianfei Chen** [1] **Chongxuan Li** [2] **Jun Zhu** [1]

Abstract

Score-based generative models have excellent performance in terms of generation quality and likelihood. They model the data distribution by matching a parameterized score network with first-order data score functions. The score network can be used to define an ODE ("score-based diffusion ODE") for exact likelihood evaluation. However, the relationship between the likelihood of the ODE and the score matching objective is unclear. In this work, we prove that matching the first-order score is not sufficient to maximize the likelihood of the ODE, by showing a gap between the maximum likelihood and score matching objectives. To fill up this gap, we show that the negative likelihood of the ODE can be bounded by controlling the first, second, and third-order score matching errors; and we further present a novel high-order denoising score matching method to enable maximum likelihood training of score-based diffusion ODEs. Our algorithm guarantees that the higher-order matching error is bounded by the training error and the lower-order errors. We empirically observe that by high-order score matching, score-based diffusion ODEs achieve better likelihood on both synthetic data and CIFAR-10, while retaining the high generation quality.

1. Introduction

Score-based generative models (SGMs) have shown promising performance in both sample quality and likelihood evaluation (Song et al., 2020; Vahdat et al., 2021; Dockhorn et al., 2021; Kingma et al., 2021), with applications on many tasks, such as image generation (Dhariwal & Nichol, 2021; Meng et al., 2021b), speech synthesis (Chen et al., 2020; Kong et al., 2020), shape generation (Luo & Hu, 2021; Zhou et al., 2021) and lossless compression (Kingma et al., 2021). SGMs define a forward diffusion process by a stochastic differential equation (SDE) that gradually perturbs the data distribution to a simple noise distribution. The forward diffusion process has an equivalent reverse-time process with analytical form (Anderson, 1982), which depends on the *score function* (the gradient of the log probability density) of the forward process at each time step. SGMs learn a parameterized neural network (called "score model") (Song & Ermon, 2019; 2020; Song et al., 2020) to estimate the score functions, and further define probabilistic models by the score model.



(a) Data (b) ScoreODE (first-order SM) (c) ScoreODE (third-order SM)

Figure 1. An 1-D mixture-of-Gaussians data distribution (a) and ScoreODE's density (b,c). (b): ScoreODE trained by minimizing the first-order score matching (SM) objective in (Song et al., 2021), which fails to estimate the density around $x = -0.25$. (c): ScoreODE trained by minimizing our proposed third-order SM objective, which approximates the data density well.

Two types of probabilistic models are proposed with the learned score model. One is the score-based diffusion SDE (Song et al. (2020), "ScoreSDE" for short), which is defined by approximately reversing the diffusion process from the noise distribution by the score model and it can generate high-quality samples. The other is the score-based diffusion ordinary differential equation (Song et al. (2020; 2021), "ScoreODE" for short), which is defined by approximating the probability flow ODE (Song et al., 2020) of the forward process whose marginal distribution is the same as the forward process at each time. ScoreODEs can be viewed as continuous normalizing flows (Chen et al., 2018). Thus,

[1]Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua-Bosch Joint Center for ML, BNRist Center, State Key Lab for Intell. Tech. & Sys., Tsinghua University; Peng Cheng Laboratory [2]Gaoling School of AI, Renmin University of China; Beijing Key Lab of Big Data Management and Analysis Methods , Beijing, China. Correspondence to: Jianfei Chen <jianfeic@tsinghua.edu.cn>, Jun Zhu <dcszj@tsinghua.edu.cn>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

unlikely SDEs, they can compute exact likelihood by ODE solvers (Grathwohl et al., 2018).

In practice, the score models are trained by minimizing an objective of a time-weighted combination of score matching losses (Hyvärinen & Dayan, 2005; Vincent, 2011). It is proven that for a specific weighting function, minimizing the score matching objective is equivalent to maximizing the likelihood of the ScoreSDE (Song et al., 2021). However, minimizing the score matching objective does not necessarily maximize the likelihood of the ScoreODE. This is problematic since ScoreODEs are used for likelihood estimation. In fact, ScoreODEs trained with the score matching objective may even fail to estimate the likelihood of very simple data distributions, as shown in Fig. 1.

In this paper, we systematically analyze the relationship between the score matching objective and the maximum likelihood objective of ScoreODEs, and present a new algorithm to learn ScoreODEs via maximizing likelihood. Theoretically, we derive an equivalent form of the KL divergence[1] between the data distribution and the ScoreODE distribution, which explicitly reveals the gap between the KL divergence and the original first-order score matching objective. Computationally, as directly optimizing the equivalent form requires an ODE solver, which is time-consuming (Chen et al., 2018), we derive an upper bound of the KL divergence by controlling the approximation errors of the first, second, and third-order score matching. The optimal solution for this score model by minimizing the upper bound is still the data score function, so the learned score model can still be used for the sample methods in SGMs (such as PC samplers in (Song et al., 2020)) to generate high-quality samples.

Based on the analyses, we propose a novel high-order denoising score matching algorithm to train the score models, which theoretically guarantees bounded approximation errors of high-order score matching. We further propose some scale-up techniques for practice. Our experimental results empirically show that the proposed training method can improve the likelihood of ScoreODEs on both synthetic data and CIFAR-10, while retaining the high generation quality.

2. Score-Based Generative Models

We first review the dynamics and probability distributions defined by SGMs. The relationship between these dynamics is illustrated in Fig. 2(a).

2.1. ScoreSDEs and Maximum Likelihood Training

Let $q_0(\boldsymbol{x}_0)$ denote an unknown d-dimensional data distribution. SGMs define an SDE from time 0 to T (called "forward process") with $\boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0)$ and

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

where $\boldsymbol{x}_t$ is the solution at time t, $\boldsymbol{f}(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ and $g(t) \in \mathbb{R}$ are fixed functions such that the marginal distribution of $\boldsymbol{x}_T$ is approximately $\mathcal{N}(\boldsymbol{x}_T|\boldsymbol{0}, \sigma_T^2\boldsymbol{I})$ with $\sigma_T > 0$, and $\boldsymbol{w}_t \in \mathbb{R}^d$ is the standard Wiener process. Let $q_t(\boldsymbol{x}_t)$ denote the marginal distribution of $\boldsymbol{x}_t$. Under some regularity conditions, the forward process in Eqn. (1) has an equivalent reverse process (Anderson, 1982) from time T to 0 with $\boldsymbol{x}_T \sim q_T(\boldsymbol{x}_T)$:

$$\mathrm{d}\boldsymbol{x}_t = [\boldsymbol{f}(\boldsymbol{x}_t, t) - g(t)^2\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\boldsymbol{w}}_t, \tag{2}$$

where $\bar{\boldsymbol{w}}_t$ is a standard Wiener process in the reverse-time. The marginal distribution of $\boldsymbol{x}_t$ in the reverse process in Eqn. (2) is also $q_t(\boldsymbol{x}_t)$. The only unknown term in Eqn. (2) is the time-dependent score function $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$. SGMs use a neural network $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ parameterized by θ to estimate the score function for all $\boldsymbol{x}_t \in \mathbb{R}^d$ and $t \in [0, T]$. The neural network is trained by matching $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ and $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$ with the *score matching objective*:

$$\begin{aligned}&\mathcal{J}_{\mathrm{SM}}(\theta; \lambda(\cdot))\\ &:= \frac{1}{2}\int_0^T \lambda(t)\mathbb{E}_{q_t(\boldsymbol{x}_t)}\Big[\|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)\|_2^2\Big]\mathrm{d}t,\end{aligned}$$

where $\lambda(\cdot) > 0$ is a weighting function. By approximating the score function with $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$, SGMs define a parameterized reverse SDE from time T to time 0 with probability at each time t denoted as $p_t^{\mathrm{SDE}}(\boldsymbol{x}_t)$ (omitting the subscript θ). In particular, suppose $\boldsymbol{x}_T \sim p_T^{\mathrm{SDE}}(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_T|\boldsymbol{0}, \sigma_T^2\boldsymbol{I})$, the parameterized reverse SDE is defined by

$$\mathrm{d}\boldsymbol{x}_t = [\boldsymbol{f}(\boldsymbol{x}_t, t) - g(t)^2\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\boldsymbol{w}}_t, \tag{3}$$

where the marginal distribution of $\boldsymbol{x}_t$ is $p_t^{\mathrm{SDE}}(\boldsymbol{x}_t)$. By solving the parameterized reverse SDE, SGMs achieve excellent sample quality in many tasks (Song et al., 2020; Dhariwal & Nichol, 2021). Moreover, the KL divergence between $q_0(\boldsymbol{x}_0)$ and $p_0^{\mathrm{SDE}}(\boldsymbol{x}_0)$ can be bounded by the score matching error with the weight $g(\cdot)^2$ (Song et al., 2021):

$$D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{SDE}}) \leq D_{\mathrm{KL}}(q_T \parallel p_T^{\mathrm{SDE}}) + \mathcal{J}_{\mathrm{SM}}(\theta; g(\cdot)^2). \tag{4}$$

Therefore, minimizing $\mathcal{J}_{\mathrm{SM}}(\theta; g(\cdot)^2)$ is equivalent to maximum likelihood training of p_0^{SDE}. In the rest of this work, we mainly consider the maximum likelihood perspective, namely $\lambda(\cdot) = g(\cdot)^2$, and denote $\mathcal{J}_{\mathrm{SM}}(\theta) := \mathcal{J}_{\mathrm{SM}}(\theta; g(\cdot)^2)$.

2.2. ScoreODEs and Exact Likelihood Evaluation

The parameterized ScoreSDE cannot be used for exact likelihood evaluation, because we cannot evaluate $p_0^{\mathrm{SDE}}(\boldsymbol{x}_0)$ exactly for a given point $\boldsymbol{x}_0 \in \mathbb{R}^d$. However, every SDE

[1] Maximum likelihood estimation is equivalent to minimizing a KL divergence.

has an associated *probability flow ODE* (Song et al., 2020) whose marginal distribution at each time t is the same as that of the SDE. Unlike SDEs, the log-density of the ODE can be exactly computed by the "Instantaneous Change of Variables" (Chen et al., 2018). Particularly, the probability flow ODE of the forward process in Eqn. (1) is

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{h}_q(\boldsymbol{x}_t, t) \coloneqq \boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{1}{2}g(t)^2\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t). \quad (5)$$

If we start from $\boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0)$, the distribution of each $\boldsymbol{x}_t$ during the ODE trajectory is also $q_t(\boldsymbol{x}_t)$. Note that the only unknown term in Eqn. (5) is also the score function $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$. By approximating the score function with $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$, SGMs define a parameterized ODE called "score-based diffusion ODE" (ScoreODE) with probability $p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)$ (omitting the subscript θ) at each time t. In particular, suppose $\boldsymbol{x}_T \sim p_T^{\mathrm{ODE}}(\boldsymbol{x}_T) = p_T^{\mathrm{SDE}}(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_T|\boldsymbol{0}, \sigma_T^2\boldsymbol{I})$, the ScoreODE is defined by

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{h}_p(\boldsymbol{x}_t, t) \coloneqq \boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{1}{2}g(t)^2\boldsymbol{s}_\theta(\boldsymbol{x}_t, t), \quad (6)$$

where the marginal distribution of $\boldsymbol{x}_t$ is $p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)$. By the "Instantaneous Change of Variables" (Chen et al., 2018), $\log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)$ of $\boldsymbol{x}_t$ can be computed by integrating:

$$\frac{\mathrm{d}\log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)}{\mathrm{d}t} = -\operatorname{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{h}_p(\boldsymbol{x}_t, t)). \quad (7)$$

If $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \equiv \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t, t)$ for all $\boldsymbol{x}_t \in \mathbb{R}^d$ and $t \in [0, T]$, we have $\boldsymbol{h}_p(\boldsymbol{x}_t, t) = \boldsymbol{h}_q(\boldsymbol{x}_t, t)$. In this case, for all $\boldsymbol{x}_0 \in \mathbb{R}^d$, $\log p_0^{\mathrm{ODE}}(\boldsymbol{x}_0) - \log q_0(\boldsymbol{x}_0) = \log p_T^{\mathrm{ODE}}(\boldsymbol{x}_T) - \log q_T(\boldsymbol{x}_T) \approx 0$ (because $q_T \approx p_T^{\mathrm{ODE}}$), where $\boldsymbol{x}_T$ is the point at time T given by Eqn. (5) starting with $\boldsymbol{x}_0$ at time 0. Inspired by this, SGMs use the ScoreODE with the score model for exact likelihood evaluations.

However, if $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \neq \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t, t)$, there is no theoretical guarantee for bounding $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$ of ScoreODEs by the score matching objective $\mathcal{J}_{\mathrm{SM}}(\theta)$. As a result, after minimizing $\mathcal{J}_{\mathrm{SM}}(\theta)$, Song et al. (2021) find that the ScoreSDE of the "Variance Exploding" type (Song et al., 2020) achieves state-of-the-art sample quality, but the log-likelihood of the corresponding ScoreODE is much worse (e.g., ≈ 3.45 bits/dim on CIFAR-10 dataset) than that of other comparable models (e.g., ≈ 3.13 bits/dim of the "Variance Preserving" type (Song et al., 2020)). Such observations pose a serious concern on the behavior of the likelihood of p_0^{ODE}, which is not well understood. We aim to provide a systematical analysis in this work.

3. Relationship between Score Matching and KL Divergence of ScoreODEs

The likelihood evaluation needs the distribution p_0^{ODE} of ScoreODEs, which is usually different from the distribution p_0^{SDE} of ScoreSDEs (see Appendix B for detailed discussions). Although the score matching objective $\mathcal{J}_{\mathrm{SM}}(\theta)$ can upper bound the KL divergence of ScoreSDEs (up to constants) according to Eqn. (4), it is not enough to upper bound the KL divergence of ScoreODEs. To the best of our knowledge, there is no theoretical analysis of the relationship between the score matching objectives and the KL divergence of ScoreODEs.

In this section, we reveal the relationship between the first-order score matching objective $\mathcal{J}_{\mathrm{SM}}(\theta)$ and the KL divergence $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$ of ScoreODEs, and upper bound $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$ by involving high-order score matching. Fig. 2 demonstrates our theoretical analysis and the connection with previous works.

3.1. KL Divergence of ScoreODEs

In the following theorem, we propose a formulation of the KL divergence $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$ by integration, which shows that there is a non-zero gap between the KL divergence and the score matching objective $\mathcal{J}_{\mathrm{SM}}(\theta)$.

Theorem 3.1 (Proof in Appendix E.1)**.** *Let q_0 be the data distribution and q_t be the marginal distribution at time t through the forward diffusion process defined in Eqn. (1), and p_t^{ODE} be the marginal distribution at time t through the ScoreODE defined in Eqn. (6). Under some regularity conditions in Appendix A, we have*

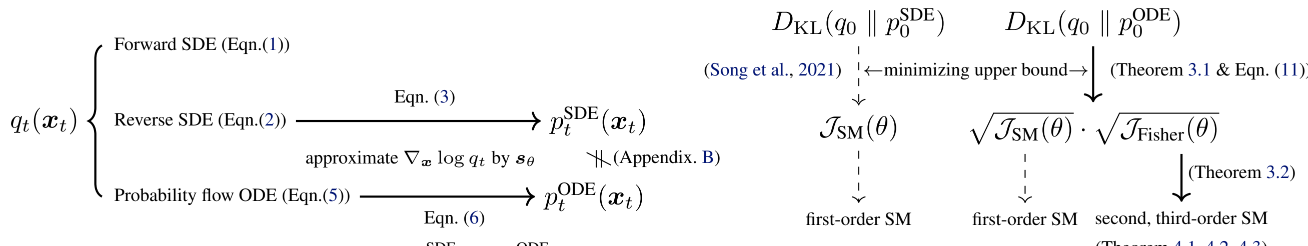
$$\begin{aligned} D_{\mathrm{KL}}(q_0 \parallel p_0^{ODE}) &= D_{\mathrm{KL}}(q_T \parallel p_T^{ODE}) + \mathcal{J}_{ODE}(\theta) \\ &= \underbrace{D_{\mathrm{KL}}(q_T \parallel p_T^{ODE}) + \mathcal{J}_{SM}(\theta)}_{\text{upper bound of } D_{\mathrm{KL}}(q_0 \parallel p_0^{SDE}) \text{ in Eqn. (4)}} + \mathcal{J}_{Diff}(\theta), \end{aligned}$$

where

$$\begin{aligned} \mathcal{J}_{ODE}(\theta) \coloneqq \frac{1}{2}\int_0^T g(t)^2 \mathbb{E}_{q_t(\boldsymbol{x}_t)}\Big[&(\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t))^\top \\ &(\nabla_{\boldsymbol{x}} \log p_t^{ODE}(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t))\Big]\mathrm{d}t, \end{aligned} \quad (8)$$

$$\begin{aligned} \mathcal{J}_{Diff}(\theta) \coloneqq \frac{1}{2}\int_0^T g(t)^2 \mathbb{E}_{q_t(\boldsymbol{x}_t)}\Big[&(\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t))^\top \\ &(\nabla_{\boldsymbol{x}} \log p_t^{ODE}(\boldsymbol{x}_t) - \boldsymbol{s}_\theta(\boldsymbol{x}_t, t))\Big]\mathrm{d}t. \end{aligned} \quad (9)$$

As p_T^{ODE} is fixed, minimizing $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$ is equivalent to minimizing $\mathcal{J}_{\mathrm{ODE}}(\theta)$, which includes $\mathcal{J}_{\mathrm{SM}}(\theta)$ and $\mathcal{J}_{\mathrm{Diff}}(\theta)$. While minimizing $\mathcal{J}_{\mathrm{SM}}(\theta)$ reduces the difference between $\boldsymbol{s}_\theta(\cdot, t)$ and the data score $\nabla_{\boldsymbol{x}} \log q_t$, it cannot control the error of $\mathcal{J}_{\mathrm{Diff}}(\theta)$, which includes the difference between $\boldsymbol{s}_\theta(\cdot, t)$ and the model score $\nabla_{\boldsymbol{x}} \log p_t^{\mathrm{ODE}}$. As we show in Fig. 1, only minimizing $\mathcal{J}_{\mathrm{SM}}(\theta)$ may cause problematic model density of the ScoreODE.



(a) Relationship between q_t, p_t^{SDE} and p_t^{ODE}.

(b) Relationship between SM objectives and KL divergence.

Figure 2. An illustration of the theoretical analysis of score-based generative models and score matching (SM) objectives.

3.2. Bounding the KL Divergence of ScoreODEs by High-Order Score Matching Errors

The KL divergence in Theorem 3.1 includes the model score function $\nabla_{\boldsymbol{x}} \log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)$, which needs ODE solvers to compute (see Appendix. D for details) and thus is hard to scale up. Below we derive an upper bound of $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$ by involving high-order score matching, which can avoid the computation of $\nabla_{\boldsymbol{x}} \log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)$. Specifically, we show that by bounding the first, second and third-order score matching errors for $\boldsymbol{s}_\theta, \nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta$ and $\nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta)$ with the corresponding order score functions[2] of q_t, we can further upper bound $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$.

Let $D_{\mathrm{F}}(q \parallel p) := \mathbb{E}_{q(\boldsymbol{x})}\|\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})\|_2^2$ denote the Fisher divergence between two distributions q and p. Define

$$\mathcal{J}_{\mathrm{Fisher}}(\theta) := \frac{1}{2}\int_0^T g(t)^2 D_{\mathrm{F}}(q_t \parallel p_t^{\mathrm{ODE}})\mathrm{d}t. \tag{10}$$

By straightforwardly using Cauchy-Schwarz inequality for Eqn. (8), we have

$$\mathcal{J}_{\mathrm{ODE}}(\theta) \leq \sqrt{\mathcal{J}_{\mathrm{SM}}(\theta)} \cdot \sqrt{\mathcal{J}_{\mathrm{Fisher}}(\theta)}. \tag{11}$$

The upper bound shows that we can minimize the KL divergence $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$ of ScoreODE by minimizing $\mathcal{J}_{\mathrm{SM}}(\theta)$ and $\mathcal{J}_{\mathrm{Fisher}}(\theta)$ simultaneously. When $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \equiv \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$ for all $\boldsymbol{x}_t \in \mathbb{R}^d$ and $t \in [0, T]$, the equality holds and $\mathcal{J}_{\mathrm{ODE}}(\theta) = \mathcal{J}_{\mathrm{SM}}(\theta) = 0$.

We can regard $\mathcal{J}_{\mathrm{Fisher}}(\theta)$ as a weighted Fisher divergence between q_t and p_t^{ODE} during $t \in [0, T]$. Directly computing the Fisher divergence also needs ODE solvers (see Appendix.D for details) and is hard to scale up. To address this problem, we present the following theorem for bounding $D_{\mathrm{F}}(q_t \parallel p_t^{\mathrm{ODE}})$, which introduces high-order score matching and avoids the computation costs of ODE solvers.

Theorem 3.2 (Proof in Appendix E.2)**.** *Assume that there exists $C > 0$, such that for all $t \in [0, T]$ and $\boldsymbol{x}_t \in \mathbb{R}^d$, $\|\nabla_{\boldsymbol{x}}^2 \log p_t^{ODE}(\boldsymbol{x}_t)\|_2 \leq C$ (where ∇^2 denotes the second-order derivative (Hessian), and $\|\cdot\|_2$ is the spectral norm of a matrix). And assume that there exist $\delta_1, \delta_2, \delta_3 > 0$, such that for all $\boldsymbol{x}_t \in \mathbb{R}^d$, $t \in [0, T]$, $s_\theta(\boldsymbol{x}_t, t)$ satisfies*

$$\begin{aligned}
&\|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)\|_2 \leq \delta_1, \\
&\|\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\|_F \leq \delta_2, \\
&\|\nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)) - \nabla_{\boldsymbol{x}} \operatorname{tr}\big(\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\big)\|_2 \leq \delta_3,
\end{aligned}$$

where $\|\cdot\|_F$ is the Frobenius norm of matrices. Then there exists $U(t; \delta_1, \delta_2, \delta_3, C, q) \geq 0$ which depends on the forward process q while is independent of θ, such that $D_{\mathrm{F}}(q_t \parallel p_t^{ODE}) \leq U(t; \delta_1, \delta_2, \delta_3, C, q)$. Moreover, for all $t \in [0, T]$, if $g(t) \neq 0$, then $U(t; \delta_1, \delta_2, \delta_3, C, q)$ is a strictly increasing function of δ_1, δ_2 and δ_3, respectively.

Theorem 3.2 shows that by bounding the errors between $\boldsymbol{s}_\theta$, $\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta$ and $\nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta)$ with the first, second and third-order score functions of q_t, we can upper bound $D_{\mathrm{F}}(q_t \parallel p_t^{\mathrm{ODE}})$ and then upper bound $\mathcal{J}_{\mathrm{Fisher}}(\theta)$. As $\mathcal{J}_{\mathrm{SM}}(\theta)$ can also be upper bounded by the first-order score matching error of $\boldsymbol{s}_\theta$, we can further upper bound $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$ by the first, second and third-order score matching errors.

Note that the high-order score matching for $\boldsymbol{s}_\theta$ is compatible for the traditional SGMs, because the optimal solution for $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ in the non-parametric limit is still $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$. So after minimizing $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$ by high-order score matching, we can still use $\boldsymbol{s}_\theta$ for the sample methods in SGMs, while directly minimizing $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$ by exact likelihood ODE solvers cannot ensure that (see Appendix. E.3 for detailed reasons).

4. Error-Bounded High-Order Denoising Score Matching (DSM)

The analysis in Sec. 3.2 shows that we can upper bound the KL divergence $D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})$ of the ScoreODE by controlling the first, second and third-order score matching errors of the score model $\boldsymbol{s}_\theta$ (and its corresponding derivatives) at each time $t \in [0, T]$. Inspired by this, we generalize the traditional denoising score matching (DSM) (Vincent,

[2]In this work, we refer to $\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)$ as the second-order score function of q_t, and $\nabla_{\boldsymbol{x}} \operatorname{tr}\big(\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\big)$ as the third-order score function of q_t.

2011) to second and third orders to minimize the high-order score matching errors. In this section, we propose a novel high-order DSM method, such that the higher-order score matching error can be exactly bounded by the training error and the lower-order score matching error. All the proofs in this section are in Appendix F.

Without loss of generality, in this section, we focus on a fixed time $t \in (0, T]$, and propose the error-bounded high-order DSM method for matching the forward process distribution q_t at the fixed time t. The whole training objective for the entire process by $t \in [0, T]$ is detailed in Sec. 5. Moreover, for SGMs, there commonly exist $\alpha_t \in \mathbb{R}$ and $\sigma_t \in \mathbb{R}_{>0}$, such that the forward process satisfies $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\alpha_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I})$ for $\boldsymbol{x}_0, \boldsymbol{x}_t \sim q_{t0}(\boldsymbol{x}_t, \boldsymbol{x}_0)$ (Song et al., 2020). Therefore, we always assume $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\alpha_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I})$ in the rest of this work.

4.1. First-Order Denoising Score Matching

We present some preliminaries on first-order denoising score matching (Vincent, 2011; Song et al., 2020) in this section.

The first-order score matching objective $\mathcal{J}_{\text{SM}}(\theta)$ needs the data score functions $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$ for all $t \in [0, T]$, which are unknown in practice. To address this issue, SGMs leverage the denoising score matching method (Vincent, 2011) to train the score models. For a fixed time t, one can learn a first-order score model $\boldsymbol{s}_1(\cdot, t; \theta) : \mathbb{R}^d \to \mathbb{R}^d$ parameterized by θ which minimizes

$$\mathbb{E}_{q_t(\boldsymbol{x}_t)}\left[\left\|\boldsymbol{s}_1(\boldsymbol{x}_t, t; \theta) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)\right\|_2^2\right],$$

by optimizing the (first-order) DSM objective:

$$\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\frac{1}{\sigma_t^2}\left\|\sigma_t \boldsymbol{s}_1(\boldsymbol{x}_t, t; \theta) + \boldsymbol{\epsilon}\right\|_2^2\right], \quad (12)$$

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

4.2. High-Order Denoising Score Matching

In this section, we generalize the first-order DSM to second and third orders for matching the second-order and third-order score functions defined in Theorem 3.2. We firstly present the second-order method below.

Theorem 4.1. *(Error-Bounded Second-Order DSM) Suppose that $\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t)$ is an estimation for the first-order data score function $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$, then we can learn a second-order score model $\boldsymbol{s}_2(\cdot, t; \theta) : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ parameterized by θ which minimizes*

$$\mathbb{E}_{q_t(\boldsymbol{x}_t)}\left[\left\|\boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta) - \nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right\|_F^2\right],$$

by optimizing

$$\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\frac{1}{\sigma_t^4}\left\|\sigma_t^2 \boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta) + \boldsymbol{I} - \boldsymbol{\ell}_1\boldsymbol{\ell}_1^\top\right\|_F^2\right], \quad (13)$$

where

$$\boldsymbol{\ell}_1(\boldsymbol{\epsilon}, \boldsymbol{x}_0, t) \coloneqq \sigma_t\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t) + \boldsymbol{\epsilon}, \qquad \boldsymbol{x}_t = \alpha_t\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \quad (14)$$

Moreover, denote the first-order score matching error as $\delta_1(\boldsymbol{x}_t, t) \coloneqq \|\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)\|_2$, then $\forall \boldsymbol{x}_t, \theta$, the score matching error for $\boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta)$ can be bounded by

$$\begin{aligned}&\left\|\boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta) - \nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right\|_F \\ \le\ &\|\boldsymbol{s}_2(\boldsymbol{x}_t, t, \theta) - \boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta^*)\|_F + \delta_1^2(\boldsymbol{x}_t, t).\end{aligned}$$

Theorem 4.1 shows that the DSM objective in problem (13) is a valid surrogate for second-order score matching, because the difference between the model score $\boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta)$ and the true second-order score $\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)$ can be bounded by the training error $\|\boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta) - \boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta^*)\|_F$ and the first-order score matching error $\delta_1(\boldsymbol{x}_t, t)$. Note that previous second-order DSM method proposed in Meng et al. (2021a) does not have such error-bounded property, and we refer to Sec. 6 and Appendix F.4 for the detailed comparison.

The DSM objective in problem (13) requires learning a matrix-valued function $\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)$, but sometimes we only need the trace $\operatorname{tr}\left(\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right)$ of the second-order score function. Below we present a corollary for only matching the trace of the second-order score function.

Corollary 4.2. *(Error-Bounded Trace of Second-Order DSM) We can learn a second-order trace score model $\boldsymbol{s}_2^{\text{trace}}(\cdot, t; \theta) : \mathbb{R}^d \to \mathbb{R}$ which minimizes*

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

by optimizing

$$\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\frac{1}{\sigma_t^4}\left|\sigma_t^2 \boldsymbol{s}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta) + d - \|\boldsymbol{\ell}_1\|_2^2\right|^2\right]. \quad (15)$$

The estimation error for $\boldsymbol{s}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta)$ can be bounded by:

$$\begin{aligned}&\left|\boldsymbol{s}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta) - \operatorname{tr}\left(\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right)\right| \\ \le\ &|\boldsymbol{s}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta) - \boldsymbol{s}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta^*)| + \delta_1^2(\boldsymbol{x}_t, t).\end{aligned}$$

Finally, we present the third-order DSM method. The score matching error can also be bounded by the training error and the first, second-order score matching errors.

Theorem 4.3. *(Error-Bounded Third-Order DSM) Suppose that $\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t)$ is an estimation for $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$ and $\hat{\boldsymbol{s}}_2(\boldsymbol{x}_t, t)$ is an estimation for $\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)$, then we can learn a third-order score model $\boldsymbol{s}_3(\cdot, t; \theta) : \mathbb{R}^d \to \mathbb{R}^d$ which minimizes*

$$\mathbb{E}_{q_t(\boldsymbol{x}_t)}\left[\left\|\boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta) - \nabla_{\boldsymbol{x}} \operatorname{tr}\left(\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right)\right\|_2^2\right],$$

by optimizing

$$\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\frac{1}{\sigma_t^6}\left\|\sigma_t^3 \boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta) + \boldsymbol{\ell}_3\right\|_2^2\right] \quad (16)$$

where

$$\begin{aligned} \boldsymbol{\ell}_1(\boldsymbol{\epsilon}, \boldsymbol{x}_0, t) &\coloneqq \sigma_t \hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t) + \boldsymbol{\epsilon}, \\ \boldsymbol{\ell}_2(\boldsymbol{\epsilon}, \boldsymbol{x}_0, t) &\coloneqq \sigma_t^2 \hat{\boldsymbol{s}}_2(\boldsymbol{x}_t, t) + \boldsymbol{I}, \\ \boldsymbol{\ell}_3(\boldsymbol{\epsilon}, \boldsymbol{x}_0, t) &\coloneqq \left(\|\boldsymbol{\ell}_1\|_2^2 \boldsymbol{I} - \operatorname{tr}(\boldsymbol{\ell}_2)\boldsymbol{I} - 2\boldsymbol{\ell}_2\right)\boldsymbol{\ell}_1, \\ \boldsymbol{x}_t &= \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \end{aligned} \tag{17}$$

Denote the first-order score matching error as $\delta_1(\boldsymbol{x}_t, t) \coloneqq \|\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)\|_2$ *and the second-order score matching errors as* $\delta_2(\boldsymbol{x}_t, t) \coloneqq \|\hat{\boldsymbol{s}}_2(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\|_F$ *and* $\delta_{2,tr}(\boldsymbol{x}_t, t) \coloneqq |\operatorname{tr}(\hat{\boldsymbol{s}}_2(\boldsymbol{x}_t, t)) - \operatorname{tr}\left(\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right)|$. *Then* $\forall \boldsymbol{x}_t, \theta$, *the score matching error for* $\boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta)$ *can be bounded by:*

$$\begin{aligned} &\left\|\boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta) - \nabla_{\boldsymbol{x}} \operatorname{tr}\left(\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right)\right\|_2 \\ \leq &\ \|\boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta) - \boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta^*)\|_2 + \left(\delta_1^2 + \delta_{2,tr} + 2\delta_2\right)\delta_1^2 \end{aligned}$$

Theorem 4.3 shows that the third-order score matching needs first-order score matching, second-order score matching and trace of second-order score matching. The third-order score matching error can also be bounded by the training error and the lower-order score matching errors. We believe that our construction for the error-bounded high-order DSM can be extended to even higher orders by carefully designing the training objectives. In this paper, we only focus on the second and third-order methods, and leave this extension for future work.

5. Training Score Models by High-Order DSM

Building upon the high-order DSM for a specific time t (see Sec. 4), we present our algorithm to train ScoreODEs in the perspective of maximum likelihood by considering all timesteps $t \in [0, T]$. Practically, we further leverage the "noise-prediction" trick (Kingma et al., 2021; Ho et al., 2020) for variance reduction and the Skilling-Hutchinson trace estimator (Skilling, 1989; Hutchinson, 1989) to compute the involved high-order derivatives efficiently. The whole training algorithm is presented detailedly in Appendix. H.

5.1. Variance Reduction by Time-Reweighting

Theoretically, to train score models for all $t \in [0, T]$, we need to integrate the DSM objectives in Eqn. (12)(13)(15)(16) from $t = 0$ to $t = T$, which needs ODE solvers and is time-consuming. Instead, in practice, we follow the method in (Song et al., 2020; Ho et al., 2020) which uses Monte-Carlo method to unbiasedly estimate the objectives by sample $t \in [0, T]$ from a proposal distribution $p(t)$, avoiding ODE solvers. The main problem for the Monte-Carlo method is that sometimes the sample variance of the objectives may be large due to the different value of $\frac{1}{\sigma_t}$. To reduce the variance, we take the time-reweighted objectives by multiplying σ_t^2,σ_t^4,σ_t^6 with the corresponding first, second, third-order score matching objectives at each time t, which is known as the "noise-prediction" trick (Kingma et al., 2021; Ho et al., 2020) for the first-order DSM objective. Specifically, assume that $\boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0)$, the time proposal distribution $p(t) = \mathcal{U}[0, T]$ is the uniform distribution in $[0, T]$, the random noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ follows the standard Gaussian distribution, and let $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$. The training objective for the first-order DSM in Eqn. (12) through all $t \in [0, T]$ is

$$\mathcal{J}_{\text{DSM}}^{(1)}(\theta) \coloneqq \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\left\|\sigma_t \boldsymbol{s}_\theta(\boldsymbol{x}_t, t) + \boldsymbol{\epsilon}\right\|_2^2\right], \tag{18}$$

which empirically has low sample variance (Song et al., 2020; Ho et al., 2020).

As for the second and third-order DSM objectives, we take $\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t) \coloneqq \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ for the first-order score function estimation, and $\hat{\boldsymbol{s}}_2(\boldsymbol{x}_t, t) \coloneqq \nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ for the second-order score function estimation, and both disable the gradient computations for θ (which can be easily implemented by `stop_gradient` or `detach`). The reason for disabling gradients is because the high-order DSM only needs the estimation values of the lower-order score functions, as shown in Theorem. 4.1 and 4.3. The training objectives through all $t \in [0, T]$ for the second-order DSM in Eqn. (13), for the trace of second-order DSM in Eqn. (15) and for the third-order DSM in Eqn. (16) are

$$\begin{aligned} \mathcal{J}_{\text{DSM}}^{(2)}(\theta) &\coloneqq \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\left\|\sigma_t^2 \nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t) + \boldsymbol{I} - \boldsymbol{\ell}_1 \boldsymbol{\ell}_1^\top\right\|_F^2\right], \\ \mathcal{J}_{\text{DSM}}^{(2,\text{tr})}(\theta) &\coloneqq \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\left|\sigma_t^2 \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)) + d - \|\boldsymbol{\ell}_1\|_2^2\right|^2\right], \\ \mathcal{J}_{\text{DSM}}^{(3)}(\theta) &\coloneqq \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\left\|\sigma_t^3 \nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)) + \boldsymbol{\ell}_3\right\|_2^2\right], \end{aligned}$$

respectively, where $\boldsymbol{\ell}_1, \boldsymbol{\ell}_3$ are the same as that in Eqn. (17). Combining with the first-order objective in Eqn. (18), our final training objective is

$$\min_\theta \mathcal{J}_{\text{DSM}}^{(1)}(\theta) + \lambda_1 \left(\mathcal{J}_{\text{DSM}}^{(2)}(\theta) + \mathcal{J}_{\text{DSM}}^{(2,tr)}(\theta)\right) + \lambda_2 \mathcal{J}_{\text{DSM}}^{(3)}(\theta), \tag{19}$$

where $\lambda_1, \lambda_2 \geq 0$ are hyperparameters, and we discuss in Appendix. I.1 for details.

Remark. Although as shown in Eqn. (7), we can exactly compute $\log p_0^{\text{ODE}}(\boldsymbol{x}_0)$ for a given data point $\boldsymbol{x}_0$ via solving the ODE (Chen et al., 2018; Grathwohl et al., 2018), this method is hard to scale up for the large neural networks used in SGMs. For example, it takes $2 \sim 3$ minutes for evaluating $\log p_0^{\text{ODE}}$ for a single batch of the ScoreODE used in (Song et al., 2020). Instead, our proposed method leverages Monte-Carlo methods to avoid the ODE solvers.

5.2. Scalability and Numerical Stability

As the second and third-order DSM objectives need to compute the high-order derivatives of a neural network and are expensive for high-dimensional data, we use the Skilling-Hutchinson trace estimator (Skilling, 1989; Hutchinson, 1989) to unbiasedly estimate the trace of the Jacobian (Grathwohl et al., 2018) and the Frobenius norm of the Jacobian (Finlay et al., 2020). The detailed objectives for high-dimensional data can be found in Appendix. G.

In practice, we often face numerical instability problems for t near to 0. We follow Song et al. (2021) to choose a small starting time $\epsilon > 0$, and both of the training and the evaluation are performed for $t \in [\epsilon, T]$ instead of $[0, T]$. Note that the likelihood evaluation is still exact, because we use $p_\epsilon^{\text{ODE}}(\boldsymbol{x}_0)$ to compute the log-likelihood of $\boldsymbol{x}_0$, which is still a well-defined density model (Song et al., 2021).

6. Related Work

ScoreODEs are special formulations of Neural ODEs (NODEs) (Chen et al., 2018), and our proposed method can be viewed as maximum likelihood training of NODEs by high-order score matching. Traditional maximum likelihood training for NODEs aims to match the NODE distribution at $t = 0$ with the data distribution, and it cannot control the distribution between 0 and T. Finlay et al. (2020) show that training NODEs by simply maximizing likelihood could result in unnecessary complex dynamics, which is hard to solve. Instead, our high-order score matching objective is to match the distributions between the forward process distribution and the NODE distribution at each time t, and empirically the dynamics is kind of smooth. Moreover, our proposed algorithm uses the Monto-Carlo method to unbiasedly estimate the objectives, which does not need any black-box ODE solvers. Therefore, our algorithm is suitable for maximum likelihood training for large-scale NODEs.

Recently, Meng et al. (2021a) propose a high-order DSM method for estimating the second-order data score functions. However, the training objective of our proposed error-bounded high-order DSM is different from that of Meng et al. (2021a). Our proposed algorithm can guarantee a bounded error of the high-order score matching exactly by the lower-order estimation errors and the training error, while the score matching error raised by minimizing the objective in Meng et al. (2021a) may be unbounded, even when the lower-order score matching error is small and the training error of the high-order DSM is zero (see Appendix F.4 for detailed analysis). Moreover, our method can also be used to train a separate high-order score model in other applications, such as uncertainty quantification and Ozaki sampling presented in Meng et al. (2021a).

7. Experiments

In this section, we demonstrate that our proposed high-order DSM algorithm can improve the likelihood of ScoreODEs, while retaining the high sample quality of the corresponding ScoreSDEs. Particularly, we use the Variance Exploding (VE) (Song et al., 2020) type diffusion models, which empirically have shown high sample quality by the ScoreSDE but poor likelihood by the ScoreODE (Song et al., 2021) when trained by minimizing the first-order score matching objective $\mathcal{J}_{\text{SM}}(\theta)$. We implement our experiments by JAX (Bradbury et al., 2018), which is efficient for computing the derivatives of the score models. In all experiments, we choose the start time $\epsilon = 10^{-5}$, which follows the default settings in Song et al. (2020). In this section, we refer to "first-order score matching" as minimizing the objective in Eqn. (19) with $\lambda_1 = \lambda_2 = 0$, and "second-order score matching" as minimizing the one with $\lambda_2 = 0$. Please see Appendix. I.1 for detailed settings about λ_1 and λ_2. The released code can be found at `https://github.com/LuChengTHU/mle_score_ode`.

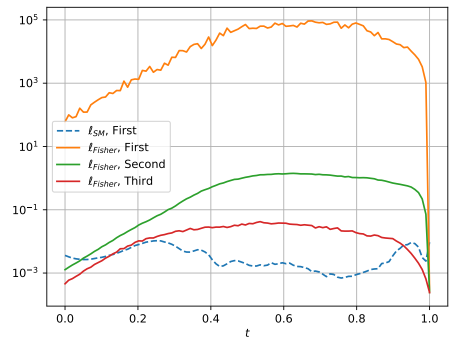


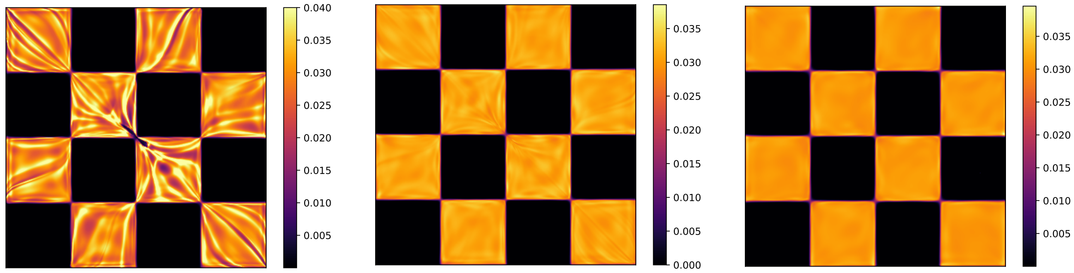
Figure 3. $\ell_{\text{Fisher}}(t)$ and $\ell_{\text{SM}}(t)$ of ScoreODEs (VE type) on 1-D mixture of Gaussians, trained by minimizing the first, second, third-order score matching objectives.

Remark. Song et al. (2021) use a proposal distribution $t \sim p(t)$ to adjust the weighting for different time t to minimize $\mathcal{J}_{\text{SM}}(\theta)$. In our experiments, we mainly focus on the VE type, whose proposal distribution $p(t) = \mathcal{U}[0, T]$ is the same as that of our final objective in Eqn. (19).

7.1. Example: 1-D Mixture of Gaussians

We take an example to demonstrate how the high-order score matching training impacts the weighted Fisher divergence $\mathcal{J}_{\text{Fisher}}(\theta)$ and the model density of ScoreODEs. Let q_0 be a 1-D Gaussian mixture distribution as shown in Fig. 1 (a). We train a score model of VE type by minimizing the first-order score matching objective $\mathcal{J}_{\text{SM}}(\theta)$, and use the corresponding ScoreODE to evaluate the model density, as shown in Fig. 1 (b). The model density achieved by only first-order score matching is quite different from the data distribution at some data points. However, by third-order score matching, the model density in Fig. 1 (c) is quite similar to the data distribution, showing the effectiveness of our proposed algorithm.

We further analyze the difference of $\mathcal{J}_{\text{Fisher}}(\theta)$ between the



(a) First-order score matching (b) Second-order score matching (c) Third-order score matching

Figure 4. Model density of ScoreODEs (VE type) on 2-D checkerboard data.

first, second, third-order score matching training. As q_0 is a Gaussian mixture distribution, we can prove that each q_t is also a Gaussian mixture distribution, and $\nabla_{\boldsymbol{x}} \log q_t$ can be analytically computed. Denote

$$\ell_{\text{Fisher}}(t) \coloneqq \frac{1}{2} g(t)^2 D_{\text{F}}(q_t \parallel p_t^{\text{ODE}}),$$
$$\ell_{\text{SM}}(t) \coloneqq \frac{1}{2} g(t)^2 \mathbb{E}_{q_t(\boldsymbol{x}_t)} \|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)\|_2^2,$$

which are the integrands of each time t for $\mathcal{J}_{\text{SM}}(\theta)$ and $\mathcal{J}_{\text{Fisher}}(\theta)$. We use the method in Appendix. D to compute $D_{\text{F}}(q_t \parallel p_t^{\text{ODE}})$, and evaluate $\ell_{\text{Fisher}}(t)$ and $\ell_{\text{SM}}(t)$ from $t = \epsilon$ to $t = T$ with 100 points of the ScoreODEs trained by first, second, third-order denoising score matching objectives. As shown in Fig. 3, although $\ell_{\text{SM}}(t)$ is small when minimizing first-order score matching objective, $\ell_{\text{Fisher}}(t)$ is rather large for most t, which means $\mathcal{J}_{\text{Fisher}}(\theta)$ is also large. However, the second and third score matching gradually reduce $\ell_{\text{Fisher}}(t)$, which shows that high-order score matching training can reduce $\ell_{\text{Fisher}}(t)$ and then reduce $\mathcal{J}_{\text{Fisher}}(\theta)$. Such results empirically validate that controlling the high-order score matching error can control the Fisher divergence, as proposed in Theorem 3.2.

7.2. Density Modeling on 2-D Checkerboard Data

We then train ScoreODEs (VE type) by the first, second, third-order score matchings on the *checkerboard* data whose density is multi-modal, as shown in Fig. 4. We use a simple MLP neural network with Swish activations (Ramachandran et al., 2017), and the detailed settings are in Appendix. I.3.

For the second and third-order score matchings, we do not use any pre-trained lower-order score models. Instead, we train the score model and its first, second-order derivatives from scratch, using the objective in Eqn. (19). The model density by the first-order score matching is rather poor, because it cannot control the value of $\nabla_{\boldsymbol{x}} \log p_t^{\text{ODE}}$ in $\mathcal{J}_{\text{Diff}}(\theta)$ in Eqn. (9). The second-order score matching can control part of $\nabla_{\boldsymbol{x}} \log p_t^{\text{ODE}}$ and improve the model density, while the third-order score matching can reduce the Fisher divergence $D_{\text{F}}(q_t \parallel p_t^{\text{ODE}})$ and achieve excellent model density. Such results show the effectiveness of our error-bounded high-order denoising score matching methods, and accord with our theoretical analysis in Theorem 3.1 and 3.2.

Table 1. Negative log-likelihood (NLL) in bits/dim (bpd) and sample quality (FID scores) on CIFAR-10 and ImageNet 32x32.

Model	CIFAR-10		ImageNet 32x32
	NLL ↓	FID ↓	NLL ↓
VE (Song et al., 2020)	3.66	2.42	4.21
VE (second) **(ours)**	3.44	**2.37**	4.06
VE (third) **(ours)**	**3.38**	2.95	**4.04**
VE (deep) (Song et al., 2020)	3.45	**2.19**	4.21
VE (deep, second) **(ours)**	3.35	2.43	4.05
VE (deep, third) **(ours)**	**3.27**	2.61	**4.03**

7.3. Density Modeling on Image Datasets

We also train ScoreODEs (VE type) on both the CIFAR-10 dataset (Krizhevsky, 2009) and the ImageNet 32x32 dataset (Deng et al., 2009), which are two of the most popular datasets for generative modeling and likelihood evaluation. We use the estimated training objectives by trace estimators, which are detailed in Appendix. G. We use the same neural networks and hyperparameters as the NCSN++ cont. model in (Song et al., 2020) (denoted as VE) and the NCSN++ cont. deep model in (Song et al., 2020) (denoted as VE (deep)), respectively (see detailed settings in Appendix. I.4 and Appendix. I.5). We evaluate the likelihood by the ScoreODE p_ϵ^{ODE}, and sample from the ScoreSDE by the PC sampler (Song et al., 2020). As shown in Table 1, our proposed high-order score matching can improve the likelihood performance of ScoreODEs, while retraining the high sample quality of the corresponding ScoreSDEs (see Appendix. J for samples). Moreover, the computation costs of the high-order DSM objectives are acceptable because of the trace estimators and the efficient "Jacobian-vector-product" computation in JAX. We list the detailed computation costs in Appendix. I.4.

We also compare the VE, VP and subVP types of ScoreODEs trained by the first-order DSM and the third-order DSM, and the detailed results are listed in Appendix. J.

8. Conclusion

We propose maximum likelihood training for ScoreODEs by novel high-order denoising score matching methods. We analyze the relationship between the score matching objectives and the KL divergence from the data distribution to the ScoreODE distribution. Based on it, we provide an upper bound of the KL divergence, which can be controlled by minimizing the first, second, and third-order score matching errors of score models. To minimize the high-order score matching errors, we further propose a high-order DSM algorithm, such that the higher-order score matching error can be bounded by exactly the training error and the lower-order score matching errors. The optimal solution for the score model is still the same as the original training objective of SGMs. Empirically, our method can greatly improve the model density of ScoreODEs of the Variance Exploding type on several density modeling benchmarks. Finally, we believe that our training method is also suitable for other SGMs, including the Variance Preserving (VP) type (Song et al., 2020), the latent space type (Vahdat et al., 2021) and the critically-damped Langevin diffusion type (Dockhorn et al., 2021). Such extensions are left for future work.

Acknowledgements

This work was supported by National Key Research and Development Project of China (No. 2021ZD0110502); NSF of China Projects (Nos. 62061136001, 61620106010, 62076145, U19B2034, U1811461, U19A2081, 6197222, 62106120); Beijing NSF Project (No. JQ19016); Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098; a grant from Tsinghua Institute for Guo Qiang; the NVIDIA NVAIL Program with GPU/DGX Acceleration; the High Performance Computing Center, Tsinghua University; and Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China.

References

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., and Zhang, Q. JAX: Composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366*, 2018.

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. Diffusion models beat GANs on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021.

Dockhorn, T., Vahdat, A., and Kreis, K. Score-based generative modeling with critically-damped langevin diffusion. *arXiv preprint arXiv:2112.07068*, 2021.

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. *arXiv preprint arXiv:1810.01367*, 2018.

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

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

Hutchinson, M. F. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

Hyvärinen, A. and Dayan, P. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

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

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Luo, S. and Hu, W. Diffusion probabilistic models for 3D point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2837–2845, 2021.

Meng, C., Song, Y., Li, W., and Ermon, S. Estimating high order gradients of the data distribution by denoising. *Advances in Neural Information Processing Systems*, 34, 2021a.

Meng, C., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. SDEdit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021b.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Skilling, J. The eigenvalues of mega-dimensional matrices. In *Maximum Entropy and Bayesian Methods*, pp. 455–466. Springer, 1989.

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

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

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

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

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

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

Zhou, L., Du, Y., and Wu, J. 3D shape generation and completion through point-voxel diffusion. *arXiv preprint arXiv:2104.03670*, 2021.

Table 2. Three distributions: $q_t(\boldsymbol{x}_t)$, $p_t^{\text{SDE}}(\boldsymbol{x}_t)$ and $p_t^{\text{ODE}}(\boldsymbol{x}_t)$ and their corresponding equivalent SDEs / ODEs.

Distribution	Dynamics type	Formulation
$q_t(\boldsymbol{x}_t)$	Forward SDE	$\mathrm{d}\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}_t$
	Reverse SDE	$\mathrm{d}\boldsymbol{x}_t = [\boldsymbol{f}(\boldsymbol{x}_t, t) - g(t)^2\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\boldsymbol{w}}_t$
	Probability flow ODE	$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{1}{2}g(t)^2\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$
$p_t^{\text{SDE}}(\boldsymbol{x}_t)$	Forward SDE	$\mathrm{d}\boldsymbol{x}_t = [\boldsymbol{f}(\boldsymbol{x}_t, t) + g(t)^2(\nabla \log p_t^{\text{SDE}}(\boldsymbol{x}_t) - \boldsymbol{s}_\theta(\boldsymbol{x}_t, t))]\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}_t$
	Reverse SDE	$\mathrm{d}\boldsymbol{x}_t = [\boldsymbol{f}(\boldsymbol{x}_t, t) - g(t)^2\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\boldsymbol{w}}_t$
	Probability flow ODE	$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}_t, t) - g(t)^2\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) + \frac{1}{2}g(t)^2\nabla_{\boldsymbol{x}} \log p_t^{\text{SDE}}(\boldsymbol{x}_t)$
$p_t^{\text{ODE}}(\boldsymbol{x}_t)$	Forward SDE	$\backslash$
	Reverse SDE	$\backslash$
	Probability flow ODE	$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{1}{2}g(t)^2\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$

A. Assumptions

We follow the regularity assumptions in (Song et al., 2021) to ensure the existence of reverse-time SDEs and probability flow ODEs and the correctness of the "integration by parts" tricks for the computation of KL divergence. And to ensure the existence of the third score function, we change the assumptions of differentiability. For completeness, we list all these assumptions in this section.

For simplicity, in the Appendix sections, we use $\nabla(\cdot)$ to denote $\nabla_{\boldsymbol{x}}(\cdot)$ and omit the subscript $\boldsymbol{x}$. And we denote $\nabla \cdot \boldsymbol{h}(\boldsymbol{x}) \coloneqq \mathrm{tr}(\nabla \boldsymbol{h}(\boldsymbol{x}))$ as the divergence of a function $\boldsymbol{h}(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}^d$.

Assumption A.1. We make the following assumptions, most of which are presented in (Song et al., 2021):

1. $q_0(\boldsymbol{x}) \in \mathcal{C}^3$ and $\mathbb{E}_{q_0(\boldsymbol{x})}[\|\boldsymbol{x}\|_2^2] < \infty$.
2. $\forall t \in [0, T] : \boldsymbol{f}(\cdot, t) \in \mathcal{C}^2$. And $\exists C > 0, \forall \boldsymbol{x} \in \mathbb{R}^d, t \in [0, T] : \|\boldsymbol{f}(\boldsymbol{x}, t)\|_2 \le C(1 + \|\boldsymbol{x}\|_2)$.
3. $\exists C > 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d : \|\boldsymbol{f}(\boldsymbol{x}, t) - \boldsymbol{f}(\boldsymbol{y}, t)\|_2 \le C\|\boldsymbol{x} - \boldsymbol{y}\|_2$.
4. $g \in \mathcal{C}$ and $\forall t \in [0, T], |g(t)| > 0$.
5. For any open bounded set $\mathcal{O}$, $\int_0^T \int_{\mathcal{O}} \|q_t(\boldsymbol{x})\|_2^2 + d \cdot g(t)^2 \|\nabla q_t(\boldsymbol{x})\|_2^2 \mathrm{d}\boldsymbol{x}\mathrm{d}t < \infty$.
6. $\exists C > 0, \forall \boldsymbol{x} \in \mathbb{R}^d, t \in [0, T] : \|\nabla q_t(\boldsymbol{x})\|_2^2 \le C(1 + \|\boldsymbol{x}\|_2)$.
7. $\exists C > 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d : \|\nabla \log q_t(\boldsymbol{x}) - \nabla \log q_t(\boldsymbol{y})\|_2 \le C\|\boldsymbol{x} - \boldsymbol{y}\|_2$.
8. $\exists C > 0, \forall \boldsymbol{x} \in \mathbb{R}^d, t \in [0, T] : \|\boldsymbol{s}_\theta(\boldsymbol{x}, t)\|_2 \le C(1 + \|\boldsymbol{x}\|_2)$.
9. $\exists C > 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d : \|\boldsymbol{s}_\theta(\boldsymbol{x}, t) - \boldsymbol{s}_\theta(\boldsymbol{y}, t)\|_2 \le C\|\boldsymbol{x} - \boldsymbol{y}\|_2$.
10. Novikov's condition: $\mathbb{E}\left[\exp\left(\frac{1}{2}\int_0^T \|\nabla \log q_t(\boldsymbol{x}) - \boldsymbol{s}_\theta(\boldsymbol{x}, t)\|_2^2 \mathrm{d}t\right)\right] < \infty$.
11. $\forall t \in [0, T], \exists k > 0 : q_t(\boldsymbol{x}) = O(e^{-\|\boldsymbol{x}\|_2^k})$, $p_t^{\text{SDE}}(\boldsymbol{x}) = O(e^{-\|\boldsymbol{x}\|_2^k})$, $p_t^{\text{ODE}}(\boldsymbol{x}) = O(e^{-\|\boldsymbol{x}\|_2^k})$ as $\|\boldsymbol{x}\|_2 \to \infty$.

B. Distribution gap between score-based diffusion SDEs and ODEs

We list the corresponding equivalent SDEs and ODEs of $q_t(\boldsymbol{x}_t)$, $p_t^{\text{SDE}}(\boldsymbol{x}_t)$ and $p_t^{\text{ODE}}(\boldsymbol{x}_t)$ in Table 2. In most cases, $q_t(\boldsymbol{x}_t)$, $p_t^{\text{SDE}}(\boldsymbol{x}_t)$ and $p_t^{\text{ODE}}(\boldsymbol{x}_t)$ are different distributions, which we will prove in this section.

According to the reverse SDE and probability flow ODE listed in the table, it is obvious that when $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \neq \nabla \log q_t(\boldsymbol{x}_t)$, $q_t(\boldsymbol{x}_t)$ and p_t^{SDE} are different, and $q_t(\boldsymbol{x}_t)$, p_t^{ODE} are different. Below we show that in most cases, p_t^{SDE} and p_t^{ODE} are also different.

Firstly, we should notice that the probability flow ODE of the ScoreSDE in Eqn. (3) is

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}_t, t) - g(t)^2 \boldsymbol{s}_\theta(\boldsymbol{x}_t, t) + \frac{1}{2} g(t)^2 \nabla_{\boldsymbol{x}} \log p_t^{\text{SDE}}(\boldsymbol{x}_t), \tag{20}$$

where the distribution of $\boldsymbol{x}_t$ during the trajectory is also $p_t^{\text{SDE}}(\boldsymbol{x}_t)$. Below we show that for the commonly-used SGMs, in most cases, the probability flow ODE in Eqn. (20) of the ScoreSDE is different from the ScoreODE in Eqn. (6).

Proposition B.1. *Assume $\boldsymbol{f}(\boldsymbol{x}_t, t) = \alpha(t)\boldsymbol{x}_t$ is a linear function of $\boldsymbol{x}_t$ for all $\boldsymbol{x}_t \in \mathbb{R}^d$ and $t \in [0, T]$, where $\alpha(t) \in \mathbb{R}$. If for all $\boldsymbol{x}_t \in \mathbb{R}^d$ and $t \in [0, T]$, $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \equiv \nabla_{\boldsymbol{x}} \log p_t^{SDE}(\boldsymbol{x}_t)$, then p_t^{SDE} is a Gaussian distribution for all $t \in [0, T]$.*

Proof. If $\boldsymbol{s}_\theta(\cdot, t) \equiv p_t^{\text{SDE}}$, then for any $\boldsymbol{x} \in \mathbb{R}^d$, by Fokker-Planck equation, we have

$$\frac{\partial p_t^{\text{SDE}}(\boldsymbol{x})}{\partial t} = \nabla_{\boldsymbol{x}} \cdot \left(\left(\boldsymbol{f}(\boldsymbol{x}, t) - \frac{1}{2} g(t)^2 \nabla_{\boldsymbol{x}} \log p_t^{\text{SDE}}(\boldsymbol{x}) \right) p_t^{\text{SDE}}(\boldsymbol{x}) \right). \tag{21}$$

On the other hand, if we start the forward process in Eqn. (1) with $\boldsymbol{x}_0 \sim p_0^{\text{SDE}}$, and the distribution of $\boldsymbol{x}_t$ at time t during its trajectory follows the same equation as Eqn. (21). Therefore, the distribution of $\boldsymbol{x}_t$ is also p_t^{SDE}.

As $\boldsymbol{f}(\boldsymbol{x}_t, t) = \alpha(t)\boldsymbol{x}_t$ is linear to $\boldsymbol{x}_t$ and $g(t)$ is independent of $\boldsymbol{x}_t$, there exists a function $\mu(t) : \mathbb{R} \to \mathbb{R}$ and a function $\tilde{\sigma}(t) : \mathbb{R} \to \mathbb{R}_{>0}$, such that $p^{\text{SDE}}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\mu(t)\boldsymbol{x}_0, \tilde{\sigma}(t)^2\boldsymbol{I})$. Therefore, for any $\boldsymbol{x}_T \sim p_T^{\text{SDE}}(\boldsymbol{x}_T)$, there exists a random variable $\boldsymbol{\epsilon}_T$ which is independent of $\boldsymbol{x}_0$, such that $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\boldsymbol{x}_T = \mu(t)\boldsymbol{x}_0 + \sigma(t)\boldsymbol{\epsilon}_T$. As $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \sigma_T^2\boldsymbol{I})$, by Cramér's decomposition theorem, because the random $\boldsymbol{x}_T$ is normally distributed and admits a decomposition as a sum of two independent random variables, we can conclude that $\mu(t)\boldsymbol{x}_0$ also follows a Gaussian distribution. As $\mu(t)$ is a scalar function independent of $\boldsymbol{x}_t$, we can further conclude that $\boldsymbol{x}_0$ follows a Gaussian distribution. As $p^{\text{SDE}}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\mu(t)\boldsymbol{x}_0, \tilde{\sigma}(t)^2\boldsymbol{I})$ is a Gaussian distribution whose mean is linear to $\boldsymbol{x}_0$ and covariance is independent of $\boldsymbol{x}_0$, we have $p_t^{\text{SDE}}(\boldsymbol{x}_t)$ is a Gaussian distribution for all t. □

The assumption for $\boldsymbol{f}(\boldsymbol{x}_t, t)$ is true for the common SGMs, because for VPSDE, VESDE and subVPSDE, $\boldsymbol{f}(\boldsymbol{x}_t, t)$ are all linear to $\boldsymbol{x}_t$, which enables fast sampling for denoising score matching (Song et al., 2020). In most cases, p_0^{SDE} is not a Gaussian distribution because we want to minimize $D_{\text{KL}}(q_0 \parallel p_0^{\text{SDE}})$. Therefore, there exists $t \in [0, T]$ such that $\boldsymbol{s}_\theta(\cdot, t) \neq \nabla_{\boldsymbol{x}} \log p_t^{\text{SDE}}$, and then the probability flow ODE in Eqn. (20) is different from the ScoreODE in Eqn. (6). In conclusion, in most cases, $p_0^{\text{SDE}} \neq p_0^{\text{ODE}}$, and minimizing $\mathcal{J}_{\text{SM}}(\theta)$ is minimizing $D_{\text{KL}}(q_0 \parallel p_0^{\text{SDE}})$, but is not necessarily minimizing $D_{\text{KL}}(q_0 \parallel p_0^{\text{ODE}})$.

Remark. The "variational gap" of ScoreSDEs in (Huang et al., 2021) is the gap between the "joint distribution" KL divergence of $p_{0:T}^{\text{SDE}}$ and "marginal distribution" KL divergence of p_0^{SDE}, which does not include p_t^{ODE}. We refer to Appendix C for further discussions.

Remark. When $\mathcal{J}_{\text{SM}}(\theta) = 0$, we have $\boldsymbol{s}_\theta(\cdot, t) \equiv \nabla_{\boldsymbol{x}} \log q_t$. In this case, the ScoreSDE in Eqn. (3) becomes the reverse diffusion SDE in Eqn. (2), and the ScoreODE in Eqn. (6) becomes the probability flow ODE in Eqn. (5). However, if $\boldsymbol{f}(\boldsymbol{x}_t, t)$ is linear to $\boldsymbol{x}_t$ and $q_0(\boldsymbol{x}_0)$ is not Gaussian, we have q_T is not Gaussian, so $q_T \neq p_T^{\text{SDE}}$ and $q_T \neq p_T^{\text{ODE}}$. Therefore, in this case, we still have $\boldsymbol{s}_\theta(\cdot, T) = \nabla_{\boldsymbol{x}} \log q_T \neq \nabla_{\boldsymbol{x}} \log p_T^{\text{SDE}}$, so the probability flow in Eqn. (20) of the ScoreSDE is still different from the ScoreODE in Eqn. (6), leading to the fact that $p_0^{\text{SDE}} \neq p_0^{\text{ODE}}$. (But the difference is extremely small, because they are both very similar to q_0).

C. KL divergence and variational gap of ScoreSDEs

(Song et al., 2021) give an upper bound of marginal KL divergence $D_{\text{KL}}(q_0 \parallel p_0^{\text{SDE}})$ of ScoreSDE by calculating the joint KL divergence $D_{\text{KL}}(q_{0:T} \parallel p_{0:T}^{\text{SDE}})$ (path measure) using Girsanov Theorem and Itô integrals. The upper bound is:

$$D_{\text{KL}}(q_0 \parallel p_0^{\text{SDE}}) \leq D_{\text{KL}}(q_{0:T} \parallel p_{0:T}^{\text{SDE}}) = D_{\text{KL}}(q_T \parallel p_T^{\text{SDE}}) + \frac{1}{2}\int_0^T g(t)^2 \mathbb{E}_{q_t(\boldsymbol{x}_t)} \left[\|\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)\|_2^2 \right] \mathrm{d}t \tag{22}$$

In this section, we derive the KL divergence $D_{\text{KL}}(q_0 \parallel p_0^{\text{SDE}})$ and the variational gap of the ScoreSDE. We first propose the KL divergence $D_{\text{KL}}(q_0 \parallel p_0^{\text{SDE}})$ in the following proposition.

Proposition C.1. *The KL divergence $D_{\mathrm{KL}}(q_0 \parallel p_0^{SDE})$ of the ScoreSDE is*

$$D_{\mathrm{KL}}(q_0 \parallel p_0^{SDE}) = D_{\mathrm{KL}}(q_T \parallel p_T^{SDE}) + \frac{1}{2}\int_0^T g(t)^2 \mathbb{E}_{q_t(\boldsymbol{x}_t)}\left[\|\boldsymbol{s}_\theta(\boldsymbol{x}_t,t) - \nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t)\|_2^2 - \|\boldsymbol{s}_\theta(\boldsymbol{x}_t,t) - \nabla_{\boldsymbol{x}}\log p_t^{SDE}(\boldsymbol{x}_t)\|_2^2\right]\mathrm{d}t \tag{23}$$

Proof. By Eqn. (20), we denote the probability flow ODE of ScoreSDE as

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{h}_{p^{\mathrm{SDE}}}(\boldsymbol{x}_t,t) \coloneqq \boldsymbol{f}(\boldsymbol{x}_t,t) - g(t)^2\boldsymbol{s}_\theta(\boldsymbol{x}_t,t) + \frac{1}{2}g(t)^2\nabla_{\boldsymbol{x}}\log p_t^{\mathrm{SDE}}(\boldsymbol{x}_t), \tag{24}$$

First we rewrite the KL divergence from q_0 to p_0^{SDE} in an integral form

$$\begin{aligned} D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{SDE}}) &= D_{\mathrm{KL}}(q_T \parallel p_T^{\mathrm{SDE}}) - D_{\mathrm{KL}}(q_T \parallel p_T^{\mathrm{SDE}}) + D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{SDE}}) \\ &= D_{\mathrm{KL}}(q_T \parallel p_T^{\mathrm{SDE}}) - \int_0^T \frac{\partial D_{\mathrm{KL}}(q_t \parallel p_t^{\mathrm{SDE}})}{\partial t}\mathrm{d}t \end{aligned} \tag{25}$$

Given a fixed $\boldsymbol{x}$, by the special case of Fokker-Planck equation with zero diffusion term, we can derive the time-evolution of ODE's associated probability density function by:

$$\frac{\partial q_t(\boldsymbol{x})}{\partial t} = -\nabla_{\boldsymbol{x}}\cdot(\boldsymbol{h}_q(\boldsymbol{x},t)q_t(\boldsymbol{x})), \quad \frac{\partial p_t^{\mathrm{SDE}}(\boldsymbol{x})}{\partial t} = -\nabla_{\boldsymbol{x}}\cdot(\boldsymbol{h}_{p^{\mathrm{SDE}}}(\boldsymbol{x},t)p_t^{\mathrm{SDE}}(\boldsymbol{x})) \tag{26}$$

Then we can expand the time-derivative of $D_{\mathrm{KL}}(q_t \parallel p_t^{\mathrm{SDE}})$ as

$$\begin{aligned} \frac{\partial D_{\mathrm{KL}}(q_t \parallel p_t^{\mathrm{SDE}})}{\partial t} &= \frac{\partial}{\partial t}\int q_t(\boldsymbol{x})\log\frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{SDE}}(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} \\ &= \int \frac{\partial q_t(\boldsymbol{x})}{\partial t}\log\frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{SDE}}(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} + \underbrace{\int\frac{\partial q_t(\boldsymbol{x})}{\partial t}\mathrm{d}\boldsymbol{x}}_{=0} - \int\frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{SDE}}(\boldsymbol{x})}\frac{\partial p_t^{\mathrm{SDE}}(\boldsymbol{x})}{\partial t}\mathrm{d}\boldsymbol{x} \\ &= -\int\nabla_{\boldsymbol{x}}\cdot(\boldsymbol{h}_q(\boldsymbol{x},t)q_t(\boldsymbol{x}))\log\frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{SDE}}(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} + \int\frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{SDE}}(\boldsymbol{x})}\nabla_{\boldsymbol{x}}\cdot(\boldsymbol{h}_{p^{\mathrm{SDE}}}(\boldsymbol{x},t)p_t^{\mathrm{SDE}}(\boldsymbol{x}))\mathrm{d}\boldsymbol{x} \\ &= \int(\boldsymbol{h}_q(\boldsymbol{x},t)q_t(\boldsymbol{x}))^\top\nabla_{\boldsymbol{x}}\log\frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{SDE}}(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} - \int(\boldsymbol{h}_{p^{\mathrm{SDE}}}(\boldsymbol{x},t)p_t^{\mathrm{SDE}}(\boldsymbol{x}))^\top\nabla_{\boldsymbol{x}}\frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{SDE}}(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} \\ &= \int q_t(\boldsymbol{x})\left[\boldsymbol{h}_q^\top(\boldsymbol{x},t) - \boldsymbol{h}_{p^{\mathrm{SDE}}}^\top(\boldsymbol{x},t)\right]\left[\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}) - \nabla_{\boldsymbol{x}}\log p_t^{\mathrm{SDE}}(\boldsymbol{x})\right]\mathrm{d}\boldsymbol{x} \\ &= -\frac{1}{2}g(t)^2\mathbb{E}_{q_t(\boldsymbol{x})}\left[\left(\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}\log p_t^{\mathrm{SDE}}(\boldsymbol{x}) - 2\boldsymbol{s}_\theta(\boldsymbol{x},t)\right)^\top\left(\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}) - \nabla_{\boldsymbol{x}}\log p_t^{\mathrm{SDE}}(\boldsymbol{x})\right)\right] \\ &= -\frac{1}{2}g(t)^2\mathbb{E}_{q_t(\boldsymbol{x})}\left[\|\boldsymbol{s}_\theta(\boldsymbol{x},t) - \nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x})\|_2^2 - \|\boldsymbol{s}_\theta(\boldsymbol{x},t) - \nabla_{\boldsymbol{x}}\log p_t^{\mathrm{SDE}}(\boldsymbol{x})\|_2^2\right] \end{aligned} \tag{27}$$

where Eqn. (27) is due to integration by parts under the Assumption. A.1(11), which shows that $\lim_{\boldsymbol{x}\to\infty}\boldsymbol{h}_q(\boldsymbol{x},t)q_t(\boldsymbol{x}) = 0$ and $\lim_{\boldsymbol{x}\to\infty}\boldsymbol{h}_{p^{\mathrm{SDE}}}(\boldsymbol{x},t)p_t^{\mathrm{SDE}}(\boldsymbol{x}) = 0$ for all $t\in[0,T]$. Combining with Eqn. (25), we can finish the proof. □

Thus by Eqn. (22) and Proposition C.1, the expectation of the variational gap for p_0^{SDE} under data distribution q_0 is

$$\begin{aligned} \mathrm{Gap}(q_0, p_0^{\mathrm{SDE}}) &\coloneqq D_{\mathrm{KL}}(q_{0:T} \parallel p_{0:T}^{\mathrm{SDE}}) - D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{SDE}}) \\ &= \frac{1}{2}\int_0^T g(t)^2\mathbb{E}_{q_t(\boldsymbol{x}_t)}\left[\|\boldsymbol{s}_\theta(\boldsymbol{x}_t,t) - \nabla_{\boldsymbol{x}}\log p_t^{\mathrm{SDE}}(\boldsymbol{x}_t)\|_2^2\right]\mathrm{d}t \end{aligned} \tag{28}$$

Note that this is equivalent to the expectation of the variational gap in (Huang et al., 2021). This expression tells us that

$$\mathrm{Gap}(q_0, p_0^{\mathrm{SDE}}) = 0 \Leftrightarrow \forall\boldsymbol{x}_t\in\mathbb{R}^d, t\in[0,T], \boldsymbol{s}_\theta(\boldsymbol{x}_t,t)\equiv\nabla_{\boldsymbol{x}}\log p_t^{\mathrm{SDE}}(\boldsymbol{x}_t). \tag{29}$$

Therefore, in practice, we always have $\mathrm{Gap}(q_0, p_0^{\mathrm{SDE}}) > 0$ since according to Proposition B.1, $\mathrm{Gap}(q_0, p_0^{\mathrm{SDE}}) = 0$ means that p_t^{SDE} is a Gaussian distribution for all $t\in[0,T]$, which doesn't match real data.

Besides, we should notice that the variational gap $\mathrm{Gap}(q_0, p_0^{\mathrm{SDE}})$ is not necessarily related to the KL divergence of ScoreODE, because in practice $p_0^{\mathrm{SDE}} \neq p_0^{\mathrm{ODE}}$.

D. Instantaneous change of score function of ODEs

Theorem D.1. *(Instantaneous Change of Score Function). Let $\boldsymbol{z}(t) \in \mathbb{R}^d$ be a finite continuous random variable with probability $p(\boldsymbol{z}(t), t)$ describing by an ordinary differential equation $\frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{z}(t), t)$. Assume that $\boldsymbol{f} \in C^3(\mathbb{R}^{d+1})$, then for each time t, the score function $\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}(t), t)$ follows a differential equation:*

$$\begin{aligned}\frac{\mathrm{d}\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}(t), t)}{\mathrm{d}t} =& -\nabla_{\boldsymbol{z}} \left(\operatorname{tr}(\nabla_{\boldsymbol{z}} \boldsymbol{f}(\boldsymbol{z}(t), t))\right) \\ &- \left(\nabla_{\boldsymbol{z}} \boldsymbol{f}(\boldsymbol{z}(t), t)\right)^\top \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}(t), t)\end{aligned} \tag{30}$$

Proof. Given a fixed $\boldsymbol{x}$, by the special case of Fokker-Planck equation with zero diffusion term, we have

$$\frac{\partial p(\boldsymbol{z}, t)}{\partial t} = -\nabla_{\boldsymbol{z}} \cdot (\boldsymbol{f}(\boldsymbol{z}, t) p(\boldsymbol{z}, t)) \tag{31}$$

So

$$\frac{\partial \log p(\boldsymbol{z}, t)}{\partial t} = \frac{1}{p(\boldsymbol{z}, t)} \frac{\partial p(\boldsymbol{z}, t)}{\partial t} \tag{32}$$

$$= -\nabla_{\boldsymbol{z}} \cdot \boldsymbol{f}(\boldsymbol{z}, t) - \boldsymbol{f}(\boldsymbol{z}, t)^\top \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}, t) \tag{33}$$

and

$$\frac{\partial \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}, t)}{\partial t} = \nabla_{\boldsymbol{z}} \frac{\partial \log p(\boldsymbol{z}, t)}{\partial t} \tag{34}$$

$$= -\nabla_{\boldsymbol{z}} \left(\nabla_{\boldsymbol{z}} \cdot \boldsymbol{f}(\boldsymbol{z}, t)\right) - \nabla_{\boldsymbol{z}} \left(\boldsymbol{f}(\boldsymbol{z}, t)^\top \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}, t)\right) \tag{35}$$

Then assume $\boldsymbol{z}(t)$ follows the trajectory of ODE $\frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{z}(t), t)$, the total derivative of score function $\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}(t), t)$ w.r.t. t is

$$\frac{\mathrm{d}\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}(t), t)}{\mathrm{d}t} = \frac{\partial \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}(t), t)}{\partial \boldsymbol{z}} \frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} + \frac{\partial \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}(t), t)}{\partial t} \tag{36}$$

$$= \nabla_{\boldsymbol{z}}^2 \log p(\boldsymbol{z}(t), t) \boldsymbol{f}(\boldsymbol{z}(t), t) - \nabla_{\boldsymbol{z}} \left(\nabla_{\boldsymbol{z}} \cdot \boldsymbol{f}(\boldsymbol{z}(t), t)\right) - \nabla_{\boldsymbol{z}} \Big(\boldsymbol{f}(\boldsymbol{z}(t), t)^\top \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}(t), t)\Big) \tag{37}$$

$$= -\nabla_{\boldsymbol{z}} \left(\nabla_{\boldsymbol{z}} \cdot \boldsymbol{f}(\boldsymbol{z}(t), t)\right) - \left(\nabla_{\boldsymbol{z}} \boldsymbol{f}(\boldsymbol{z}(t), t)\right)^\top \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}(t), t) \tag{38}$$

□

By Theorem D.1, we can compute $\nabla_{\boldsymbol{x}} \log p_0^{\mathrm{ODE}}(\boldsymbol{x}_0)$ by firstly solving the forward ODE $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{h}_p(\boldsymbol{x}_t, t)$ from 0 to T to get $\boldsymbol{x}_T$, then solving a reverse ODE of $(\boldsymbol{x}_t, \nabla_{\boldsymbol{x}} \log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t))$ with the initial value $(\boldsymbol{x}_T, \nabla_{\boldsymbol{x}} \log p_T^{\mathrm{ODE}}(\boldsymbol{x}_T))$ from T to 0.

E. KL divergence of ScoreODEs

In this section, we propose the proofs for Sec. 3.

E.1. Proof of Theorem 3.1

Proof. First we rewrite the KL divergence from q_0 to p_0^{ODE} in an integral form

$$D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}}) = D_{\mathrm{KL}}(q_T \parallel p_T^{\mathrm{ODE}}) - D_{\mathrm{KL}}(q_T \parallel p_T^{\mathrm{ODE}}) + D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}}) \tag{39}$$

$$= D_{\mathrm{KL}}(q_T \parallel p_T^{\mathrm{ODE}}) - \int_0^T \frac{\partial D_{\mathrm{KL}}(q_t \parallel p_t^{\mathrm{ODE}})}{\partial t} \mathrm{d}t. \tag{40}$$

Given a fixed $\boldsymbol{x}$, by the special case of Fokker-Planck equation with zero diffusion term, we can derive the time-evolution of ODE's associated probability density function by:

$$\frac{\partial q_t(\boldsymbol{x})}{\partial t} = -\nabla_{\boldsymbol{x}} \cdot (\boldsymbol{h}_q(\boldsymbol{x}, t) q_t(\boldsymbol{x})), \quad \frac{\partial p_t^{\mathrm{ODE}}(\boldsymbol{x})}{\partial t} = -\nabla_{\boldsymbol{x}} \cdot (\boldsymbol{h}_p(\boldsymbol{x}, t) p_t^{\mathrm{ODE}}(\boldsymbol{x})). \tag{41}$$

Then we can expand the time-derivative of $D_{\mathrm{KL}}(q_t \parallel p_t^{\mathrm{ODE}})$ as

$$\begin{aligned}
\frac{\partial D_{\mathrm{KL}}(q_t \parallel p_t^{\mathrm{ODE}})}{\partial t} &= \frac{\partial}{\partial t}\int q_t(\boldsymbol{x})\log\frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{ODE}}(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} && (42)\\
&= \int \frac{\partial q_t(\boldsymbol{x})}{\partial t}\log\frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{ODE}}(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} + \underbrace{\int \frac{\partial q_t(\boldsymbol{x})}{\partial t}\mathrm{d}\boldsymbol{x}}_{=0} - \int \frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{ODE}}(\boldsymbol{x})}\frac{\partial p_t^{\mathrm{ODE}}(\boldsymbol{x})}{\partial t}\mathrm{d}\boldsymbol{x} && (43)\\
&= -\int \nabla_{\boldsymbol{x}}\cdot(\boldsymbol{h}_q(\boldsymbol{x},t)q_t(\boldsymbol{x}))\log\frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{ODE}}(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} + \int \frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{ODE}}(\boldsymbol{x})}\nabla_{\boldsymbol{x}}\cdot(\boldsymbol{h}_p(\boldsymbol{x},t)p_t^{\mathrm{ODE}}(\boldsymbol{x}))\mathrm{d}\boldsymbol{x} && (44)\\
&= \int (\boldsymbol{h}_q(\boldsymbol{x},t)q_t(\boldsymbol{x}))^\top\nabla_{\boldsymbol{x}}\log\frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{ODE}}(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} - \int (\boldsymbol{h}_p(\boldsymbol{x},t)p_t^{\mathrm{ODE}}(\boldsymbol{x}))^\top\nabla_{\boldsymbol{x}}\frac{q_t(\boldsymbol{x})}{p_t^{\mathrm{ODE}}(\boldsymbol{x})}\mathrm{d}\boldsymbol{x} && (45)\\
&= \int q_t(\boldsymbol{x})\left[\boldsymbol{h}_q^\top(\boldsymbol{x},t) - \boldsymbol{h}_p^\top(\boldsymbol{x},t)\right]\left[\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}) - \nabla_{\boldsymbol{x}}\log p_t^{\mathrm{ODE}}(\boldsymbol{x})\right]\mathrm{d}\boldsymbol{x} && (46)\\
&= -\frac{1}{2}g(t)^2\mathbb{E}_{q_t(\boldsymbol{x})}\left[(\boldsymbol{s}_\theta(\boldsymbol{x},t) - \nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}))^\top\left(\nabla_{\boldsymbol{x}}\log p_t^{\mathrm{ODE}}(\boldsymbol{x}) - \nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x})\right)\right], && (47)
\end{aligned}$$

where Eqn. (45) is due to integration by parts under the Assumption. A.1(11), which shows that $\lim_{\boldsymbol{x}\to\infty}\boldsymbol{h}_q(\boldsymbol{x},t)q_t(\boldsymbol{x}) = 0$ and $\lim_{\boldsymbol{x}\to\infty}\boldsymbol{h}_p(\boldsymbol{x},t)p_t^{\mathrm{ODE}}(\boldsymbol{x}) = 0$ for all $t\in[0,T]$. Combining with Eqn. (40), we can conclude that

$$\begin{aligned}
&D_{\mathrm{KL}}(q_0 \parallel p_0^{\mathrm{ODE}})\\
&= D_{\mathrm{KL}}(q_T \parallel p_T^{\mathrm{ODE}}) + \frac{1}{2}\int_0^T g(t)^2\mathbb{E}_{q_t(\boldsymbol{x}_t)}\left[(\boldsymbol{s}_\theta(\boldsymbol{x}_t,t) - \nabla\log q_t(\boldsymbol{x}_t))^\top\left(\nabla\log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t) - \nabla\log q_t(\boldsymbol{x}_t)\right)\right]\mathrm{d}t\\
&= D_{\mathrm{KL}}(q_T \parallel p_T^{\mathrm{ODE}}) + \mathcal{J}_{\mathrm{ODE}}(\theta)\\
&= D_{\mathrm{KL}}(q_T \parallel p_T^{\mathrm{ODE}}) + \mathcal{J}_{\mathrm{SM}}(\theta) + \mathcal{J}_{\mathrm{Diff}}(\theta).
\end{aligned} \tag{48}$$

□

E.2. Proof of Theorem 3.2

Firstly, we propose a Lemma for computing $\nabla\log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t) - \nabla\log q_t(\boldsymbol{x}_t)$ with the trajectory of $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{h}_q(\boldsymbol{x}_t,t)$.

Lemma E.1. *Assume $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{h}_q(\boldsymbol{x}_t,t)$, p_t^{ODE} and q_t are defined as the same in Sec. 2. We have*

$$\begin{aligned}
\frac{\mathrm{d}(\nabla\log p_t^{ODE} - \nabla\log q_t)}{\mathrm{d}t} = &-\left(\nabla\operatorname{tr}(\nabla\boldsymbol{h}_p(\boldsymbol{x}_t,t)) - \nabla\operatorname{tr}(\nabla\boldsymbol{h}_q(\boldsymbol{x}_t,t))\right)\\
&-\left(\nabla\boldsymbol{h}_p(\boldsymbol{x}_t,t)^\top\nabla\log p_t^{ODE}(\boldsymbol{x}_t) - \nabla\boldsymbol{h}_q(\boldsymbol{x}_t,t)^\top\nabla\log q_t(\boldsymbol{x}_t)\right)\\
&-\nabla^2\log p_t^{ODE}(\boldsymbol{x}_t)\big(\boldsymbol{h}_p(\boldsymbol{x}_t,t) - \boldsymbol{h}_q(\boldsymbol{x}_t,t)\big)
\end{aligned} \tag{49}$$

Proof. (Proof of Lemma E.1)

Given a fixed $\boldsymbol{x}$, by the special case of Fokker-Planck equation with zero diffusion term, we have

$$\frac{\partial p_t^{\mathrm{ODE}}(\boldsymbol{x})}{\partial t} = -\nabla\cdot(\boldsymbol{h}_p(\boldsymbol{x}_t,t)p_t^{\mathrm{ODE}}(\boldsymbol{x})) \tag{50}$$

So

$$\begin{aligned}
\frac{\partial\log p_t^{\mathrm{ODE}}(\boldsymbol{x})}{\partial t} &= \frac{1}{p_t^{\mathrm{ODE}}(\boldsymbol{x})}\frac{\partial p_t^{\mathrm{ODE}}(\boldsymbol{x})}{\partial t} && (51)\\
&= -\nabla\cdot\boldsymbol{h}_p(\boldsymbol{x}_t,t) - \boldsymbol{h}_p(\boldsymbol{x}_t,t)^\top\nabla\log p_t^{\mathrm{ODE}}(\boldsymbol{x}) && (52)
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial\nabla\log p_t^{\mathrm{ODE}}(\boldsymbol{x})}{\partial t} &= \nabla\frac{\partial\log p_t^{\mathrm{ODE}}(\boldsymbol{x})}{\partial t} && (53)\\
&= -\nabla\left(\nabla\cdot\boldsymbol{h}_p(\boldsymbol{x}_t,t)\right) - \nabla\left(\boldsymbol{h}_p(\boldsymbol{x}_t,t)^\top\nabla\log p_t^{\mathrm{ODE}}(\boldsymbol{x})\right) && (54)
\end{aligned}$$

As $\boldsymbol{x}_t$ follows the trajectory of $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{h}_q(\boldsymbol{x}_t, t)$, the total derivative of score function $\nabla \log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)$ w.r.t. t is

$$\frac{\mathrm{d}\nabla \log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)}{\mathrm{d}t} = \frac{\partial \nabla \log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} + \frac{\partial \nabla \log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)}{\partial t} \tag{55}$$

$$= \nabla^2 \log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)\boldsymbol{h}_q(\boldsymbol{x}_t, t) - \nabla\left(\nabla \cdot \boldsymbol{h}_p(\boldsymbol{x}_t, t)\right) - \nabla\left(\boldsymbol{h}_p(\boldsymbol{x}_t, t)^\top \nabla \log p_t^{\mathrm{ODE}}(\boldsymbol{x})\right) \tag{56}$$

$$= -\nabla\left(\nabla \cdot \boldsymbol{h}_p(\boldsymbol{x}_t, t)\right) - \nabla \boldsymbol{h}_p(\boldsymbol{x}_t, t)^\top \nabla \log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t) - \nabla^2 \log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)\left(\boldsymbol{h}_p(\boldsymbol{x}_t, t) - \boldsymbol{h}_q(\boldsymbol{x}_t, t)\right) \tag{57}$$

Similarly, for $\nabla \log q_t(\boldsymbol{x}_t)$ we have

$$\frac{\mathrm{d}\nabla \log q_t(\boldsymbol{x}_t)}{\mathrm{d}t} = -\nabla\left(\nabla \cdot \boldsymbol{h}_q(\boldsymbol{x}_t, t)\right) - \nabla \boldsymbol{h}_q(\boldsymbol{x}_t, t)^\top \nabla \log q_t(\boldsymbol{x}_t) \tag{58}$$

Therefore, by combining $\frac{\mathrm{d}\nabla \log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t)}{\mathrm{d}t}$ and $\frac{\mathrm{d}\nabla \log q_t(\boldsymbol{x}_t)}{\mathrm{d}t}$, we can derive the conclusion. □

Then we prove Theorem 3.2 below.

Proof. (Proof of Theorem 3.2.)

Firstly, we have

$$\boldsymbol{h}_p(\boldsymbol{x}_t, t) - \boldsymbol{h}_q(\boldsymbol{x}_t, t) = -\frac{1}{2}g(t)^2\left(\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla \log q_t(\boldsymbol{x}_t)\right) \tag{59}$$

So we have

$$\|\boldsymbol{h}_p(\boldsymbol{x}_t, t) - \boldsymbol{h}_q(\boldsymbol{x}_t, t)\|_2 \leq \frac{1}{2}g(t)^2\delta_1, \tag{60}$$

$$\|\nabla \boldsymbol{h}_p(\boldsymbol{x}_t, t) - \nabla \boldsymbol{h}_q(\boldsymbol{x}_t, t)\|_2 \leq \|\nabla \boldsymbol{h}_p(\boldsymbol{x}_t, t) - \nabla \boldsymbol{h}_q(\boldsymbol{x}_t, t)\|_F \leq \frac{1}{2}g(t)^2\delta_2, \tag{61}$$

$$\|\nabla \operatorname{tr}(\nabla \boldsymbol{h}_p(\boldsymbol{x}_t, t)) - \nabla \operatorname{tr}(\nabla \boldsymbol{h}_q(\boldsymbol{x}_t, t))\|_2 \leq \frac{1}{2}g(t)^2\delta_3, \tag{62}$$

where $\|\cdot\|_2$ is the L_2-norm for vectors and the induced 2-norm for matrices.

Assume that $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{h}_q(\boldsymbol{x}_t, t)$, by Lemma E.1, we have

$$\nabla \log p_t^{\mathrm{ODE}}(\boldsymbol{x}_t) - \nabla \log q_t(\boldsymbol{x}_t) = \nabla \log p_T^{\mathrm{ODE}}(\boldsymbol{x}_T) - \nabla \log q_T(\boldsymbol{x}_T) \tag{63}$$

$$+ \int_t^T \left(\nabla \operatorname{tr}(\nabla \boldsymbol{h}_p(\boldsymbol{x}_s, s)) - \nabla \operatorname{tr}(\nabla \boldsymbol{h}_q(\boldsymbol{x}_s, s))\right)\mathrm{d}s \tag{64}$$

$$+ \int_t^T \left(\nabla \boldsymbol{h}_p(\boldsymbol{x}_s, s)^\top \nabla \log p_s^{\mathrm{ODE}}(\boldsymbol{x}_s) - \nabla \boldsymbol{h}_q(\boldsymbol{x}_s, s)^\top \nabla \log q_s(\boldsymbol{x}_s)\right)\mathrm{d}s \tag{65}$$

$$+ \int_t^T \nabla^2 \log p_s^{\mathrm{ODE}}(\boldsymbol{x}_s)\left(\boldsymbol{h}_p(\boldsymbol{x}_s, s) - \boldsymbol{h}_q(\boldsymbol{x}_s, s)\right)\mathrm{d}s \tag{66}$$

For simplicity, we denote $\boldsymbol{h}_p(s) \coloneqq \boldsymbol{h}_p(\boldsymbol{x}_s, s)$, $\boldsymbol{h}_q(s) \coloneqq \boldsymbol{h}_q(\boldsymbol{x}_s, s)$, $p_s \coloneqq p_s^{\mathrm{ODE}}(\boldsymbol{x}_s)$ and $q_s \coloneqq q_s(\boldsymbol{x}_s)$. We use $\|\cdot\|$ to denote the 2-norm for vectors and matrices. As

$$\nabla \boldsymbol{h}_p(s)^\top \nabla \log p_s - \nabla \boldsymbol{h}_q(s)^\top \nabla \log q_s = \left(\nabla \boldsymbol{h}_p(s) - \nabla \boldsymbol{h}_q(s)\right)^\top \left(\nabla \log p_s - \nabla \log q_s\right) \tag{67}$$

$$+ \nabla \boldsymbol{h}_q(s)^\top \left(\nabla \log p_s - \nabla \log q_s\right) + \left(\nabla \boldsymbol{h}_p(s) - \nabla \boldsymbol{h}_q(\boldsymbol{s})\right)^\top \nabla \log q_s \tag{68}$$

We have

$$\|\nabla \log p_t - \nabla \log q_t\| \leq \|\nabla \log p_T - \nabla \log q_T\| + \int_t^T \left\|\nabla \operatorname{tr}(\nabla \boldsymbol{h}_p(s)) - \nabla \operatorname{tr}(\nabla \boldsymbol{h}_q(s))\right\| \mathrm{d}s \tag{69}$$

$$+ \int_t^T \left(\left\|\nabla \boldsymbol{h}_p(s) - \nabla \boldsymbol{h}_q(s)\right\| + \left\|\nabla \boldsymbol{h}_q(s)\right\|\right) \cdot \left\|\nabla \log p_s - \nabla \log q_s\right\| \mathrm{d}s \tag{70}$$

$$+ \int_t^T \left\|\nabla \boldsymbol{h}_p(s) - \nabla \boldsymbol{h}_q(s)\right\| \cdot \left\|\nabla \log q_s\right\| \mathrm{d}s + \int_t^T \|\nabla^2 \log p_s\| \cdot \|\boldsymbol{h}_p(s) - \boldsymbol{h}_q(s)\| \,\mathrm{d}s \tag{71}$$

$$\leq \|\nabla \log p_T - \nabla \log q_T\| + \frac{1}{2}\int_t^T g(s)^2 \Big(\delta_3 + \delta_2 \|\nabla \log q_s\| + \delta_1 C\Big) \mathrm{d}s \tag{72}$$

$$+ \int_t^T \left(\frac{\delta_2}{2} g(s)^2 + \|\nabla \boldsymbol{h}_q(s)\|\right) \|\nabla \log p_s - \nabla \log q_s\| \mathrm{d}s \tag{73}$$

Denote

$$u(t) \coloneqq \|\nabla \log p_t - \nabla \log q_t\| \tag{74}$$

$$\alpha(t) \coloneqq \|\nabla \log p_T - \nabla \log q_T\| + \frac{1}{2}\int_t^T g(s)^2 \Big(\delta_3 + \delta_2 \|\nabla \log q_s\| + \delta_1 C\Big) \mathrm{d}s \tag{75}$$

$$\beta(t) \coloneqq \frac{\delta_2}{2} g(t)^2 + \|\nabla \boldsymbol{h}_q(t)\| \tag{76}$$

Then $\alpha(t) \geq 0, \beta(t) \geq 0$ are independent of θ, and we have

$$u(t) \leq \alpha(t) + \int_t^T \beta(s) u(s) \mathrm{d}s. \tag{77}$$

By Grönwall's inequality, we have

$$u(t) \leq \alpha(t) + \int_t^T \alpha(s)\beta(s) \exp\left(\int_t^s \beta(r)\mathrm{d}r\right) \mathrm{d}s, \tag{78}$$

and therefore,

$$D_{\mathrm{F}}(q_t \,\|\, p_t) = \mathbb{E}_{q_t}[u(t)^2] \leq \mathbb{E}_{q_t}\left[\left(\alpha(t) + \int_t^T \alpha(s)\beta(s)\exp\left(\int_t^s \beta(r)\mathrm{d}r\right)\mathrm{d}s\right)^2\right], \tag{79}$$

where the r.h.s. is only dependent on $\delta_1, \delta_2, \delta_3$ and C, with some constants that are only dependent on the forward process q. Furthermore, as r.h.s. is an increasing function of $\alpha(t) \geq 0$ and $\beta(t) \geq 0$, and $\alpha(t), \beta(t)$ are increasing functions of $\delta_1, \delta_2, \delta_3, C$, so we can minimize $\delta_1, \delta_2, \delta_3$ to minimize an upper bound of $D_{\mathrm{F}}(q_t \,\|\, p_t)$. □

E.3. Maximum likelihood training of ScoreODE by ODE solvers

By Eqn. (7), we have

$$\log p_0^{\mathrm{ODE}}(\boldsymbol{x}_0) = \log p_T^{\mathrm{ODE}}(\boldsymbol{x}_T) + \int_0^T \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{h}_p(\boldsymbol{x}_t, t))\mathrm{d}t, \tag{80}$$

where $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{h}_p(\boldsymbol{x}_t, t)$. So minimizing $D_{\mathrm{KL}}(q_0 \,\|\, p_0^{\mathrm{ODE}})$ is equivalent to maximizing

$$\mathbb{E}_{q_0(\boldsymbol{x}_0)}\left[\log p_0^{\mathrm{ODE}}(\boldsymbol{x}_0)\right] = \mathbb{E}_{q_0(\boldsymbol{x}_0)}\left[\log p_T^{\mathrm{ODE}}(\boldsymbol{x}_T) + \int_0^T \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{h}_p(\boldsymbol{x}_t, t))\mathrm{d}t\right]. \tag{81}$$

We can directly call ODE solvers to compute $\log p_0^{\mathrm{ODE}}(\boldsymbol{x}_0)$ for a given data point $\boldsymbol{x}_0$, and thus do maximum likelihood training (Chen et al., 2018; Grathwohl et al., 2018). However, this needs to be done at every optimization step, which is hard to scale up for the large neural networks used in SGMs. For example, it takes $2 \sim 3$ minutes for evaluating $\log p_0^{\mathrm{ODE}}$

for a single batch of the ScoreODE used in (Song et al., 2020). Besides, directly maximum likelihood training for p_0^{ODE} cannot make sure that the optimal solution for $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ is the data score function $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$, because the directly MLE for p_0^{ODE} cannot ensure that the model distribution p_t^{ODE} is also similar to the data distribution q_t at each time $t \in (0, T)$, thus cannot ensure the model ODE function $\boldsymbol{h}_p(\boldsymbol{x}_t, t)$ is similar to the data ODE function $\boldsymbol{h}_q(\boldsymbol{x}_t, t)$. In fact, ScoreODE is a special formulation of Neural ODEs (Chen et al., 2018). Empirically, Finlay et al. (2020) find that directly maximum likelihood training of Neural ODEs may cause rather complex dynamics ($\boldsymbol{h}_p(\boldsymbol{x}_t, t)$ here), which indicates that directly maximum likelihood training by ODE solvers cannot ensure $\boldsymbol{h}_p(\boldsymbol{x}_t, t)$ be similar to $\boldsymbol{h}_q(\boldsymbol{x}_t, t)$, and thus cannot ensure the score model $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ be similar to the data score function $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$.

F. Error-bounded High-Order Denoising Score Matching

In this section, we present all the lemmas and proofs for our proposed error-bounded high-order DSM algorithm.

Below we propose an expectation formulation of the first-order score function.

Lemma F.1. *Assume $(\boldsymbol{x}_t, \boldsymbol{x}_0) \sim q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, we have*

$$\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t) = \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[\nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) \right] \tag{82}$$

Proof. (Proof of Lemma F.1)

We use $\nabla(\cdot)$ to denote the derivative of $\boldsymbol{x}_t$, namely $\nabla_{\boldsymbol{x}_t}(\cdot)$.

$$\begin{aligned}
\nabla \log q_t(\boldsymbol{x}_t) &= \frac{\nabla q_t(\boldsymbol{x}_t)}{q_t(\boldsymbol{x}_t)} && (83)\\
&= \frac{\nabla \int q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) q_0(\boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0}{q_t(\boldsymbol{x}_t)} && (84)\\
&= \int \frac{q_0(\boldsymbol{x}_0) q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q_t(\boldsymbol{x}_t)} \frac{\nabla q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)} \mathrm{d}\boldsymbol{x}_0 && (85)\\
&= \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0 && (86)\\
&= \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) \right] && (87)
\end{aligned}$$

□

And below we propose an expectation formulation for the second-order score function.

Lemma F.2. *Assume $(\boldsymbol{x}_t, \boldsymbol{x}_0) \sim q(\boldsymbol{x}_t, \boldsymbol{x}_0)$, we have*

$$\begin{aligned}
&\nabla^2_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\\
&= \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \Big[\nabla^2_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) + \big(\nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\big)\big(\nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\big)^\top \Big]
\end{aligned} \tag{88}$$

and

$$\operatorname{tr}\big(\nabla^2_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\big) = \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \Big[\operatorname{tr}\big(\nabla^2_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\big) + \big\|\nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\big\|_2^2 \Big] \tag{89}$$

Proof. (Proof of Lemma F.2)

We use $\nabla(\cdot)$ to denote the derivative of $\boldsymbol{x}_t$, namely $\nabla_{\boldsymbol{x}_t}(\cdot)$. Firstly, the gradient of q_{t0} w.r.t. $\boldsymbol{x}_t$ can be calculated as

$$\begin{aligned}
\nabla q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) &= \nabla \frac{q_0(\boldsymbol{x}_0) q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q_t(\boldsymbol{x}_t)} && (90)\\
&= q_0(\boldsymbol{x}_0) \frac{q_t(\boldsymbol{x}_t) \nabla q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) \nabla q_t(\boldsymbol{x}_t)}{q_t(\boldsymbol{x}_t)^2} && (91)\\
&= \frac{q_0(\boldsymbol{x}_0) q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q_t(\boldsymbol{x}_t)} \left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t) \right) && (92)\\
&= q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t) \right) && (93)
\end{aligned}$$

Then using Lemma F.1 and the product rule, we have

$$\nabla^2 \log q_t(\boldsymbol{x}_t) = \nabla \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\right] \tag{94}$$

$$= \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)\nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) + \nabla q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)^\top \mathrm{d}\boldsymbol{x}_0 \tag{95}$$

$$= \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[\nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) + \left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\right) \nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)^\top\right] \tag{96}$$

From Lemma F.1, we also have $\mathbb{E}_{q_{t0}} \left[\nabla \log q_{0t} - \nabla \log q_t\right] = \nabla \log q_t - \nabla \log q_t = \mathbf{0}$, so

$$\mathbb{E}_{q_{t0}} \left[(\nabla \log q_{0t} - \nabla \log q_t)\nabla \log q_t^\top\right] = \mathbb{E}_{q_{t0}} \left[\nabla \log q_{0t} - \nabla \log q_t\right] \nabla \log q_t^\top \tag{97}$$

$$= \mathbf{0} \tag{98}$$

Thus Eqn. (96) can be further transformed by subtracting $\mathbf{0}$ as

$$\nabla^2 \log q_t(\boldsymbol{x}_t) \tag{99}$$

$$= \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[\nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) + \left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\right) \nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)^\top\right] \tag{100}$$

$$- \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t))\nabla \log q_t(\boldsymbol{x}_t)^\top\right] \tag{101}$$

$$= \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)} \left[\nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) + \left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\right) \left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\right)^\top\right] \tag{102}$$

which completes the proof of second-order score matrix and its trace. □

Below we propose a corollary of an expectation formulation of the sum of the first-order and the second-order score functions, which can be used to design the third-order denoising score matching method.

Corollary F.3. *Assume* $(\boldsymbol{x}_t, \boldsymbol{x}_0) \sim q(\boldsymbol{x}_t, \boldsymbol{x}_0)$*, we have*

$$\begin{aligned} &\mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\Big[\nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)^\top + \nabla^2_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\Big] \\ &= \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)^\top + \nabla^2_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t) \end{aligned} \tag{103}$$

and

$$\mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\Big[\big\|\nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\big\|_2^2 + \operatorname{tr}\big(\nabla^2_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\big)\Big] = \big\|\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\big\|_2^2 + \operatorname{tr}\big(\nabla^2_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\big) \tag{104}$$

Proof. (Proof of Corollary F.3)

This is the direct corollary of Eqn. (96) by adding $\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)^\top$ to both sides and using Lemma F.1. □

And below we propose an expectation formulation for the third-order score function.

Lemma F.4. *Assume* $(\boldsymbol{x}_t, \boldsymbol{x}_0) \sim q(\boldsymbol{x}_t, \boldsymbol{x}_0)$ *and* $\nabla^3_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathbf{0}$*, we have*

$$\nabla_{\boldsymbol{x}_t} \operatorname{tr}\big(\nabla^2_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\big) = \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\Big[\big\|\nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\big\|_2^2\big(\nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\big)\Big] \tag{105}$$

Proof. (Proof of Lemma F.4)

We use $\nabla(\cdot)$ to denote the derivative of $\boldsymbol{x}_t$, namely $\nabla_{\boldsymbol{x}_t}(\cdot)$. Firstly, according to the second-order score trace given by Lemma F.2, we have

$$\begin{aligned} &\nabla \operatorname{tr}\big(\nabla^2 \log q_t(\boldsymbol{x}_t)\big) \\ &= \underbrace{\nabla \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \operatorname{tr}\left(\nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\right) \mathrm{d}\boldsymbol{x}_0}_{(1)} + \underbrace{\nabla \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \|\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\|_2^2 \mathrm{d}\boldsymbol{x}_0}_{(2)} \end{aligned} \tag{106}$$

Before simplifying them, we should notice two simple tricks. Firstly, due to the assumption $\nabla^3 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathbf{0}$, we know that $\nabla^2 \log q_{0t}$ and $\operatorname{tr}\left(\nabla^2 \log q_{0t}\right)$ are constants, so they can be drag out of the integral w.r.t $\boldsymbol{x}_0$ and the derivative w.r.t. $\boldsymbol{x}_t$. Secondly, using Lemma F.1, we have $\mathbb{E}_{q_{t0}}\left[\nabla \log q_{0t} - \nabla \log q_t\right] = \mathbf{0}$. Also, $\nabla q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)$ can be represented by Eqn. (93). Thus

$$\begin{aligned}(1) &= \operatorname{tr}\left(\nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\right) \nabla \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \mathrm{d}\boldsymbol{x}_0 && (107)\\ &= \operatorname{tr}\left(\nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)\right) \nabla 1 && (108)\\ &= \mathbf{0} && (109)\end{aligned}$$

$$(2) = \underbrace{\int \nabla q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \|\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\|_2^2 \mathrm{d}\boldsymbol{x}_0}_{(3)} + \underbrace{\int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \nabla \|\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\|_2^2 \mathrm{d}\boldsymbol{x}_0}_{(4)} \tag{110}$$

where

$$\begin{aligned}(4) &= 2 \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \left(\nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla^2 \log q_t(\boldsymbol{x}_t)\right) \left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\right) \mathrm{d}\boldsymbol{x}_0 && (111)\\ &= 2\left(\nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla^2 \log q_t(\boldsymbol{x}_t)\right) \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\right) \mathrm{d}\boldsymbol{x}_0 && (112)\\ &= \mathbf{0} && (113)\end{aligned}$$

Combining equations above, we have

$$\begin{aligned}\nabla \operatorname{tr}\left(\nabla^2 \log q_t(\boldsymbol{x}_t)\right) &= (3) && (114)\\ &= \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \|\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\|_2^2 \left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\right) \mathrm{d}\boldsymbol{x}_0 && (115)\\ &= \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\|\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\|_2^2 \left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\right)\right] && (116)\end{aligned}$$

□

Lemma F.5. *Assume $(\boldsymbol{x}_t, \boldsymbol{x}_0) \sim q(\boldsymbol{x}_t, \boldsymbol{x}_0)$ and $\nabla^3_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathbf{0}$, we have*

$$\begin{aligned}&\nabla_{\boldsymbol{x}_t}\left(\boldsymbol{v}^\top \nabla^2_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t) \boldsymbol{v}\right)\\ &= \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\Big[\left(\left(\nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\right)^\top \boldsymbol{v}\right)^2 \left(\nabla_{\boldsymbol{x}_t} \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)\right)\Big]\end{aligned} \tag{117}$$

Proof. (Proof of Lemma F.5)

We use $\nabla(\cdot)$ to denote the derivative of $\boldsymbol{x}_t$, namely $\nabla_{\boldsymbol{x}_t}(\cdot)$. Firstly, according to the second-order score given by Lemma F.2, we have

$$\begin{aligned}\nabla\left(\boldsymbol{v}^\top \nabla^2 \log q_t(\boldsymbol{x}_t) \boldsymbol{v}\right) &= \underbrace{\nabla \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \left(\boldsymbol{v}^\top \nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) \boldsymbol{v}\right) \mathrm{d}\boldsymbol{x}_0}_{(1)} && (118)\\ &+ \underbrace{\nabla \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t) \left(\left(\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t)\right)^\top \boldsymbol{v}\right)^2 \mathrm{d}\boldsymbol{x}_0}_{(2)} && (119)\end{aligned}$$

Similar to the proof of Lemma F.4, we have

$$(1) = \left(\boldsymbol{v}^\top \nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) \boldsymbol{v}\right) \nabla \mathbf{1} = \mathbf{0}, \tag{120}$$

and

$$(2) = \underbrace{\int \nabla q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)\left((\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t))^\top \boldsymbol{v}\right)^2 \mathrm{d}\boldsymbol{x}_0}_{(3)} \tag{121}$$

$$+ \underbrace{\int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)\nabla\left((\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t))^\top \boldsymbol{v}\right)^2 \mathrm{d}\boldsymbol{x}_0}_{(4)} \tag{122}$$

Also similar to the proof of Lemma F.4, we have

$$(4) = 2\int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)\left(\left((\nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla^2 \log q_t(\boldsymbol{x}_t)\right)^\top \boldsymbol{v}\right)\left((\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t))^\top \boldsymbol{v}\right)\mathrm{d}\boldsymbol{x}_0 \tag{123}$$

$$= 2\left(\left((\nabla^2 \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla^2 \log q_t(\boldsymbol{x}_t)\right)^\top \boldsymbol{v}\right)\int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)\left((\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t))^\top \boldsymbol{v}\right)\mathrm{d}\boldsymbol{x}_0 \tag{124}$$

$$= \boldsymbol{0}, \tag{125}$$

so

$$\nabla\left(\boldsymbol{v}^\top \nabla^2 \log q_t(\boldsymbol{x}_t)\boldsymbol{v}\right) \tag{126}$$

$$= (3) \tag{127}$$

$$= \int q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)\left((\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t))^\top \boldsymbol{v}\right)^2 (\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t))\,\mathrm{d}\boldsymbol{x}_0 \tag{128}$$

$$= \mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\left[\left((\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t))^\top \boldsymbol{v}\right)^2 (\nabla \log q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0) - \nabla \log q_t(\boldsymbol{x}_t))\right] \tag{129}$$

□

F.1. Proof of Theorem 4.1

Proof. For simplicity, we denote $q_{0t} \coloneqq q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)$, $q_{t0} \coloneqq q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)$, $q_t \coloneqq q_t(\boldsymbol{x}_t)$, $q_0 \coloneqq q_0(\boldsymbol{x}_0)$, $\hat{\boldsymbol{s}}_1 \coloneqq \hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t)$, $\boldsymbol{s}_2(\theta) \coloneqq \boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta)$.

As $\nabla \log q_{0t} = -\frac{\boldsymbol{\epsilon}}{\sigma_t}$ and $\nabla^2 \log q_{0t} = -\frac{1}{\sigma_t^2}\boldsymbol{I}$, by rewriting the objective in Eqn. (13), the optimization is equivalent to

$$\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{q_t}\mathbb{E}_{q_{t0}}\left[\left\|\boldsymbol{s}_2(\theta) - \nabla^2 \log q_{0t} - (\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1)(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1)^\top\right\|_F^2\right] \tag{130}$$

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

$$\boldsymbol{s}_2(\theta^*) = \mathbb{E}_{q_{t0}}\left[\nabla^2 \log q_{0t} + (\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1)(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1)^\top\right] \tag{131}$$

By Lemma F.1 and Lemma F.2, we have

$$\boldsymbol{s}_2(\theta^*) - \nabla^2 \log q_t \tag{132}$$

$$= \mathbb{E}_{q_{t0}}\left[\hat{\boldsymbol{s}}_1\hat{\boldsymbol{s}}_1^\top - \hat{\boldsymbol{s}}_1\nabla \log q_{0t}^\top - \nabla \log q_{0t}\hat{\boldsymbol{s}}_1^\top - \nabla \log q_t \nabla \log q_t^\top + \nabla \log q_{0t}\nabla \log q_t^\top + \nabla \log q_t \nabla \log q_{0t}^\top\right] \tag{133}$$

$$= (\hat{\boldsymbol{s}}_1 - \nabla \log q_t)(\hat{\boldsymbol{s}}_1 - \nabla \log q_t)^\top \tag{134}$$

Therefore, we have

$$\|\boldsymbol{s}_2(\theta) - \nabla^2 \log q_t\|_F \leq \|\boldsymbol{s}_2(\theta) - \boldsymbol{s}_2(\theta^*)\|_F + \|\boldsymbol{s}_2(\theta^*) - \nabla^2 \log q_t\|_F \tag{135}$$

$$= \|\boldsymbol{s}_2(\theta) - \boldsymbol{s}_2(\theta^*)\|_F + \|\hat{\boldsymbol{s}}_1 - \nabla \log q_t\|_2^2 \tag{136}$$

Moreover, by leveraging the property of minimum mean square error, we should notice that the training objective can be rewritten to

$$\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{q_t(\boldsymbol{x}_t)}\|\boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta) - \boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta^*)\|_F^2, \tag{137}$$

which shows that $\|\boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta) - \boldsymbol{s}_2(\boldsymbol{x}_t, t; \theta^*)\|_F$ can be viewed as the training error. □

F.2. Proof of Corollary 4.2

Proof. For simplicity, we denote $q_{0t} \coloneqq q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)$, $q_{t0} \coloneqq q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)$, $q_t \coloneqq q_t(\boldsymbol{x}_t)$, $q_0 \coloneqq q_0(\boldsymbol{x}_0)$, $\hat{\boldsymbol{s}}_1 \coloneqq \hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t)$, $\boldsymbol{s}_2^{\text{trace}}(\theta) \coloneqq \boldsymbol{s}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta)$.

As $\nabla \log q_{0t} = -\frac{\boldsymbol{\epsilon}}{\sigma_t}$ and $\nabla^2 \log q_{0t} = -\frac{1}{\sigma_t^2}\boldsymbol{I}$, by rewriting the objective in Eqn. (15), the optimization is equivalent to

$$\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{q_t}\mathbb{E}_{q_{t0}}\left[\left|\boldsymbol{s}_2^{\text{trace}}(\theta) - \operatorname{tr}\left(\nabla^2 \log q_{0t}\right) - \|\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1\|_2^2\right|^2\right] \tag{138}$$

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

$$\boldsymbol{s}_2^{\text{trace}}(\theta^*) = \mathbb{E}_{q_{t0}}\left[\operatorname{tr}\left(\nabla^2 \log q_{0t}\right) + \|\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1\|_2^2\right] \tag{139}$$

By Lemma F.1 and Lemma F.2, similarly we have

$$\boldsymbol{s}_2^{\text{trace}}(\theta^*) - \operatorname{tr}\left(\nabla^2 \log q_t\right) = \|\hat{\boldsymbol{s}}_1 - \nabla \log q_t\|_2^2 \tag{140}$$

Therefore, we have

$$\begin{aligned} |\boldsymbol{s}_2^{\text{trace}}(\theta) - \operatorname{tr}\left(\nabla^2 \log q_t\right)| &\le |\boldsymbol{s}_2^{\text{trace}}(\theta) - \boldsymbol{s}_2^{\text{trace}}(\theta^*)| + |\boldsymbol{s}_2^{\text{trace}}(\theta^*) - \operatorname{tr}\left(\nabla^2 \log q_t\right)| & (141)\\ &= |\boldsymbol{s}_2^{\text{trace}}(\theta) - \boldsymbol{s}_2^{\text{trace}}(\theta^*)| + \|\hat{\boldsymbol{s}}_1 - \nabla \log q_t\|_2^2 & (142)\end{aligned}$$

Moreover, by leveraging the property of minimum mean square error, we should notice that the training objective can be rewritten to

$$\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{q_t(\boldsymbol{x}_t)}|\boldsymbol{s}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta) - \boldsymbol{s}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta^*)|^2, \tag{143}$$

which shows that $|\boldsymbol{s}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta) - \boldsymbol{s}_2^{\text{trace}}(\boldsymbol{x}_t, t; \theta^*)|$ can be viewed as the training error. □

F.3. Proof of Theorem 4.3

Proof. For simplicity, we denote $q_{0t} \coloneqq q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)$, $q_{t0} \coloneqq q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)$, $q_t \coloneqq q_t(\boldsymbol{x}_t)$, $q_0 \coloneqq q_0(\boldsymbol{x}_0)$, $\hat{\boldsymbol{s}}_1 \coloneqq \hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t)$, $\hat{\boldsymbol{s}}_2 \coloneqq \hat{\boldsymbol{s}}_2(\boldsymbol{x}_t, t)$, $\boldsymbol{s}_3(\theta) \coloneqq \boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta)$.

As $\nabla \log q_{0t} = -\frac{\boldsymbol{\epsilon}}{\sigma_t}$ and $\nabla^2 \log q_{0t} = -\frac{1}{\sigma_t^2}\boldsymbol{I}$, by rewriting the objective in Eqn. (16), the optimization is equivalent to

$$\begin{aligned}\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{q_t(\boldsymbol{x}_t)}\mathbb{E}_{q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)}\Bigg[\Bigg\|&\boldsymbol{s}_3(\theta) - \|\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1\|_2^2\left(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1\right)\\ &+\Big(\left(\operatorname{tr}(\hat{\boldsymbol{s}}_2) - \operatorname{tr}\left(\nabla^2 \log q_{0t}\right)\right)\boldsymbol{I} + 2\left(\hat{\boldsymbol{s}}_2 - \nabla^2 \log q_{0t}\right)\Big)\Big(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1\Big)\Bigg\|_2^2\Bigg]\end{aligned} \tag{144}$$

For fixed t and $\boldsymbol{x}_t$, minimizing the inner expectation is a minimum mean square error problem for $\boldsymbol{s}_3(\theta)$, so the optimal θ^* satisfies

$$\begin{aligned}\boldsymbol{s}_3(\theta^*) &= \mathbb{E}_{q_{t0}}\left[\left\|\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1\right\|_2^2\left(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1\right)\right] & (145)\\ &- \mathbb{E}_{q_{t0}}\left[\Big(\left(\operatorname{tr}(\hat{\boldsymbol{s}}_2) - \operatorname{tr}\left(\nabla^2 \log q_{0t}\right)\right)\boldsymbol{I} + 2\left(\hat{\boldsymbol{s}}_2 - \nabla^2 \log q_{0t}\right)\Big)\Big(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1\Big)\right] & (146)\end{aligned}$$

As $\nabla^2 \log q_{0t} = -\frac{1}{\sigma_t^2}\boldsymbol{I}$ is constant w.r.t. $\boldsymbol{x}_0$, by Lemma F.1, we have

$$\begin{aligned}\boldsymbol{s}_3(\theta^*) &= \mathbb{E}_{q_{t0}}\left[\left\|\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1\right\|_2^2\left(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1\right)\right] & (147)\\ &- \Big(\left(\operatorname{tr}(\hat{\boldsymbol{s}}_2) - \operatorname{tr}\left(\nabla^2 \log q_{0t}\right)\right)\boldsymbol{I} + 2\left(\hat{\boldsymbol{s}}_2 - \nabla^2 \log q_{0t}\right)\Big)\Big(\nabla \log q_t - \hat{\boldsymbol{s}}_1\Big) & (148)\end{aligned}$$

By Lemma F.4, we have

$$\boldsymbol{s}_3(\theta^*) - \nabla \operatorname{tr}\left(\nabla^2 \log q_t\right) \tag{149}$$
$$= \mathbb{E}_{q_{t0}}\left[\left(2\nabla \log q_{0t} \nabla \log q_{0t}^\top + 2\nabla^2 \log q_{0t} - 2\hat{\boldsymbol{s}}_2\right)\left(\nabla \log q_t - \hat{\boldsymbol{s}}_1\right)\right] \tag{150}$$
$$+ \mathbb{E}_{q_{t0}}\left[\left(\|\nabla \log q_{0t}\|_2^2 + \operatorname{tr}\left(\nabla^2 \log q_{0t}\right) - \operatorname{tr}(\hat{\boldsymbol{s}}_2)\right)\left(\nabla \log q_t - \hat{\boldsymbol{s}}_1\right)\right] \tag{151}$$
$$+ (\|\hat{\boldsymbol{s}}_1\|_2^2 - \|\nabla \log q_t\|_2^2)\mathbb{E}_{q_{t0}}[\nabla \log q_{0t}] \tag{152}$$
$$+ 2\mathbb{E}_{q_{t0}}\left[(\nabla \log q_{0t}^\top \hat{\boldsymbol{s}}_1)\hat{\boldsymbol{s}}_1 - (\nabla \log q_{0t}^\top \nabla \log q_t)\nabla \log q_t\right] \tag{153}$$
$$+ \|\nabla \log q_t\|_2^2 \nabla \log q_t - \|\hat{\boldsymbol{s}}_1\|_2^2 \hat{\boldsymbol{s}}_1 \tag{154}$$

By Lemma F.1 and Corollary F.3, we have

$$\begin{aligned}
&\boldsymbol{s}_3(\theta^*) - \nabla \operatorname{tr}\left(\nabla^2 \log q_t\right) \\
&= 2\left(\nabla \log q_t \nabla \log q_t^\top + \nabla^2 \log q_t - \hat{\boldsymbol{s}}_2\right)\left(\nabla \log q_t - \hat{\boldsymbol{s}}_1\right) \\
&\quad + \left(\|\nabla \log q_t\|_2^2 + \operatorname{tr}\left(\nabla^2 \log q_t\right) - \operatorname{tr}(\hat{\boldsymbol{s}}_2)\right)\left(\nabla \log q_t - \hat{\boldsymbol{s}}_1\right) \\
&\quad + (\|\hat{\boldsymbol{s}}_1\|_2^2 - \|\nabla \log q_t\|_2^2)\nabla \log q_t \\
&\quad + 2\left((\nabla \log q_t^\top \hat{\boldsymbol{s}}_1)\hat{\boldsymbol{s}}_1 - \|\nabla \log q_t\|_2^2 \nabla \log q_t\right) \\
&\quad + \|\nabla \log q_t\|_2^2 \nabla \log q_t - \|\hat{\boldsymbol{s}}_1\|_2^2 \hat{\boldsymbol{s}}_1 \\
&= \left(2\left(\nabla^2 \log q_t - \hat{\boldsymbol{s}}_2\right) + \left(\operatorname{tr}\left(\nabla^2 \log q_t\right) - \operatorname{tr}(\hat{\boldsymbol{s}}_2)\right)\right)\left(\nabla \log q_t - \hat{\boldsymbol{s}}_1\right) \\
&\quad + \|\nabla \log q_t - \hat{\boldsymbol{s}}_1\|_2^2(\nabla \log q_t - \hat{\boldsymbol{s}}_1)
\end{aligned} \tag{155}$$

Therefore, we have

$$\|\boldsymbol{s}_3(\theta) - \nabla \operatorname{tr}\left(\nabla^2 \log q_t\right)\|_2 \leq \|\boldsymbol{s}_3(\theta) - \boldsymbol{s}_3(\theta^*)\|_2 + \|\boldsymbol{s}_3(\theta^*) - \nabla \operatorname{tr}\left(\nabla^2 \log q_t\right)\|_2 \tag{156}$$
$$\leq \|\boldsymbol{s}_3(\theta) - \boldsymbol{s}_3(\theta^*)\|_2 + \|\nabla \log q_t - \hat{\boldsymbol{s}}_1\|_2^3 \tag{157}$$
$$+ \left(2\left\|\nabla^2 \log q_t - \hat{\boldsymbol{s}}_2\right\|_F + \left|\operatorname{tr}\left(\nabla^2 \log q_t\right) - \operatorname{tr}(\hat{\boldsymbol{s}}_2)\right|\right)\|\nabla \log q_t - \hat{\boldsymbol{s}}_1\|_2 \tag{158}$$

Moreover, by leveraging the property of minimum mean square error, we should notice that the training objective can be rewritten to

$$\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{q_t(\boldsymbol{x}_t)}\|\boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta) - \boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta^*)\|_2^2, \tag{159}$$

which shows that $\|\boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta) - \boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta^*)\|_2$ can be viewed as the training error. □

F.4. Difference between error-bounded high-order denoising score matching and previous high-order score matching in Meng et al. (2021a)

In this section, we analyze the DSM objective in Meng et al. (2021a), and show that the objective in Meng et al. (2021a) has the unbounded-error property, which means even if the training error is zero and the first-order score matching error is any small, the second-order score matching error may be arbitrarily large.

(Meng et al., 2021a) proposed an objective for estimating second-order score

$$\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{q_t}\mathbb{E}_{q_{t0}}\left[\left\|\boldsymbol{s}_2(\theta) - \nabla^2 \log q_{0t} - \nabla \log q_{0t} \nabla \log q_{0t}^\top + \hat{\boldsymbol{s}}_1 \hat{\boldsymbol{s}}_1^\top\right\|_F^2\right] \tag{160}$$

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

$$\boldsymbol{s}_2(\theta^*) = \mathbb{E}_{q_{t0}}\left[\nabla^2 \log q_{0t} + \nabla \log q_{0t} \nabla \log q_{0t}^\top - \hat{\boldsymbol{s}}_1 \hat{\boldsymbol{s}}_1^\top\right] \tag{161}$$

By Lemma F.1 and Lemma F.2, we have

$$\boldsymbol{s}_2(\theta^*) - \nabla^2 \log q_t = \mathbb{E}_{q_{t0}} \left[-\hat{\boldsymbol{s}}_1 \hat{\boldsymbol{s}}_1^\top - \nabla \log q_t \nabla \log q_t^\top + \nabla \log q_{0t} \nabla \log q_t^\top + \nabla \log q_t \nabla \log q_{0t}^\top \right] \tag{162}$$
$$= \nabla \log q_t \nabla \log q_t^\top - \hat{\boldsymbol{s}}_1 \hat{\boldsymbol{s}}_1^\top \tag{163}$$

However, below we show that even if θ achieves its optimal solution θ^* and the first-order score matching error $\|\hat{\boldsymbol{s}}_1 - \nabla \log q_t\|$ is small, the second-order score matching error $\boldsymbol{s}_2(\theta^*) - \nabla^2 \log q_t$ may still be rather large.

We construct an example to demonstrate this point. Suppose the first-order score matching error $\hat{\boldsymbol{s}}_1 - \nabla \log q_t = \delta_1 \cdot \mathbf{1}$, where $\delta_1 > 0$ is small, then the first-order score matching error $\|\hat{\boldsymbol{s}}_1 - \nabla \log q_t\|_2 = \delta_1 \sqrt{d}$, where d is the dimension of $\boldsymbol{x}_t$. We have

$$\boldsymbol{s}_2(\theta^*) - \nabla^2 \log q_t = \nabla \log q_t \nabla \log q_t^\top - (\nabla \log q_t + \delta_1 \cdot \mathbf{1})(\nabla \log q_t + \delta_1 \cdot \mathbf{1})^\top \tag{164}$$
$$= -\delta_1 \nabla \log q_t \mathbf{1}^\top - \delta_1 \mathbf{1} \nabla \log q_t^\top - \delta_1^2 \mathbf{1}\mathbf{1}^\top \tag{165}$$

We consider the unbounded score case (Song et al., 2020), if the first-order score function $\nabla \log q_t$ is unbounded, i.e. for any $\delta_1 > 0$ and any $C > 0$, there exists $\boldsymbol{x}_t \in \mathbb{R}^d$ such that $\|\nabla \log q_t\|_2 > \frac{C + \delta_1^2 d}{2\delta_1}$, then we have

$$\|\boldsymbol{s}_2(\theta^*) - \nabla \log q_t\|_F \geq \delta_1 \|\nabla \log q_t \mathbf{1}^\top + \mathbf{1} \nabla \log q_t^\top\|_F - \delta_1^2 \|\mathbf{1}\mathbf{1}^\top\|_F \tag{166}$$
$$\geq \delta_1 \|\mathrm{diag}(\nabla \log q_t \mathbf{1}^\top + \mathbf{1} \nabla \log q_t^\top)\|_2 - \delta_1^2 \|\mathbf{1}\mathbf{1}^\top\|_F \tag{167}$$
$$= \delta_1 \|2 \nabla \log q_t\|_2 - \delta_1^2 d \tag{168}$$
$$> C, \tag{169}$$

where the second inequality is because $\|A\|_F \geq \|\mathrm{diag}(A)\|_2$, where $\mathrm{diag}(A)$ means the diagonal vector of the matrix A. Therefore, for any small δ_1, the second-order score estimation error may be arbitrarily large.

In practice, Meng et al. (2021a) do not stop gradients for $\hat{\boldsymbol{s}}_1$, which makes the optimization of the second-order score model $\boldsymbol{s}_2(\theta)$ affecting the first-order score model $\boldsymbol{s}_1(\theta)$, and thus cannot theoretically guarantee the convergence. Empirically, we implement the second-order DSM objective in Meng et al. (2021a), and we find that if we stop gradients for $\hat{\boldsymbol{s}}_1$, the model quickly diverges and cannot work. We argue that this is because of the unbounded error of their method, as shown above.

Instead, our proposed high-order DSM method has the error-bounded property, which shows that $\|\boldsymbol{s}_2(\theta^*) - \nabla \log q_t\|_F = \|\hat{\boldsymbol{s}}_1 - \nabla \log q_t\|_2^2$. So our method does not have the problem mentioned above.

G. Estimated objectives of high-order DSM for high-dimensional data

In this section, we propose the estimated high-order DSM objectives for high-dimensional data.

The second-order DSM objective in Eqn. (13) requires computing the Frobenius norm of the full Jacobian of the score model, i.e. $\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}, t)$, which typically has $\mathcal{O}(d^2)$ time complexity and is unacceptable for high dimensional real data. Moreover, the high-order DSM objectives in Eqn. (15) and Eqn. (16) include computing the divergence of score network $\nabla_{\boldsymbol{x}} \cdot \boldsymbol{s}_\theta(\boldsymbol{x}, t)$ i.e. the trace of Jacobian $\mathrm{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}, t))$, and it also has $\mathcal{O}(d^2)$ time complexity. Similar to Grathwohl et al. (2018) and Finlay et al. (2020), the cost can be reduced to $\mathcal{O}(d)$ using Hutchinson's trace estimator and automatic diffentiation provided by general deep learning frameworks, which needs one-time backpropagation only.

For a d-by-d matrix A, its trace can be unbiasedly estimated by (Hutchinson, 1989):

$$\mathrm{tr}(A) = \mathbb{E}_{p(\boldsymbol{v})} \left[\boldsymbol{v}^\top A \boldsymbol{v} \right] \tag{170}$$

where $p(\boldsymbol{v})$ is a d-dimensional distribution such that $\mathbb{E}[\boldsymbol{v}] = \mathbf{0}$ and $\mathrm{Cov}[\boldsymbol{v}] = \boldsymbol{I}$. Typical choices of $p(\boldsymbol{v})$ are a standard Gaussian or Rademacher distribution. Moreover, the Frobenius norm can also be unbiasedly estimated, since

$$\|A\|_F^2 = \mathrm{tr}(A^\top A) = \mathbb{E}_{p(\boldsymbol{v})} \left[\boldsymbol{v}^\top A^\top A \boldsymbol{v} \right] = \mathbb{E}_{p(\boldsymbol{v})} \left[\|A\boldsymbol{v}\|_2^2 \right] \tag{171}$$

Let $A = \nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}, t)$, The Jacobian-vector-product $\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}, t) \boldsymbol{v}$ can be efficiently computed by using once forward-mode automatic differentiation in JAX, making the evaluating of trace and Frobenius norm approximately the same cost as evaluating $\boldsymbol{s}_\theta(\boldsymbol{x}, t)$. And we show the time costs for the high-order DSM training in Appendix. I.4.

By leveraging the unbiased estimator, our final objectives for second-order and third-order DSM are:

$$\mathcal{J}^{(2)}_{\text{DSM,estimation}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\mathbb{E}_{p(\boldsymbol{v})}\left[\left\|\sigma_t^2 \boldsymbol{s}_{jvp} + \boldsymbol{v} - (\sigma_t\hat{\boldsymbol{s}}_1\cdot\boldsymbol{v} + \boldsymbol{\epsilon}\cdot\boldsymbol{v})(\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})\right\|_2^2\right], \tag{172}$$

$$\mathcal{J}^{(2,\text{tr})}_{\text{DSM,estimation}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\mathbb{E}_{p(\boldsymbol{v})}\left[\left|\sigma_t^2 \boldsymbol{v}^\top\boldsymbol{s}_{jvp} + \|\boldsymbol{v}\|_2^2 - |\sigma_t\hat{\boldsymbol{s}}_1\cdot\boldsymbol{v} + \boldsymbol{\epsilon}\cdot\boldsymbol{v}|^2\right|^2\right], \tag{173}$$

$$\begin{aligned}\mathcal{J}^{(3)}_{\text{DSM,estimation}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\mathbb{E}_{p(\boldsymbol{v})}\Bigg[\Big\|&\sigma_t^3 \boldsymbol{v}^\top\nabla_{\boldsymbol{x}}\boldsymbol{s}_{jvp} + |\sigma_t\hat{\boldsymbol{s}}_1\cdot\boldsymbol{v} + \boldsymbol{\epsilon}\cdot\boldsymbol{v}|^2(\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon}) - (\sigma_t^2\boldsymbol{v}^\top\hat{\boldsymbol{s}}_{jvp} + \|\boldsymbol{v}\|_2^2)(\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})\\ &- 2(\sigma_t\hat{\boldsymbol{s}}_1\cdot\boldsymbol{v} + \boldsymbol{\epsilon}\cdot\boldsymbol{v})(\sigma_t^2\hat{\boldsymbol{s}}_{jvp} + \boldsymbol{v})\Big\|_2^2\Bigg],\end{aligned} \tag{174}$$

where $\boldsymbol{s}_{jvp} = \nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x},t)\boldsymbol{v}$ is the Jacobian-vector-product as mentioned above, $\hat{\boldsymbol{s}}_1 = \texttt{stop\_gradient}(\boldsymbol{s}_\theta)$, $\hat{\boldsymbol{s}}_{jvp} = \texttt{stop\_gradient}(\boldsymbol{s}_{jvp})$, and $\boldsymbol{a}\cdot\boldsymbol{b}$ denotes the vector-inner-product for vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. We compute $\boldsymbol{v}^\top\nabla_{\boldsymbol{x}}\boldsymbol{s}_{jvp}$ by using the auto-gradient functions in JAX (which can compute the "vector-Jacobian-product" by one-time backpropagation").

G.1. Relationship between the estimated objectives and the original objectives for high-order DSM

In this section, we show that the proposed estimated objectives are actually equivalent to or can upper bound the original objectives in Sec. 5.1 for high-order DSM. Specifically, we have

$$\mathcal{J}^{(2)}_{\text{DSM}}(\theta) = \mathcal{J}^{(2)}_{\text{DSM,estimation}}(\theta), \tag{175}$$

$$\mathcal{J}^{(2,\text{tr})}_{\text{DSM}}(\theta) \le \mathcal{J}^{(2,\text{tr})}_{\text{DSM,estimation}}(\theta), \tag{176}$$

$$\mathcal{J}^{(3)}_{\text{DSM}}(\theta) \le \mathcal{J}^{(3)}_{\text{DSM,estimation}}(\theta). \tag{177}$$

Firstly, the estimated second-order DSM objective in Eqn. (172) is equivalent to the original objective in Sec. 5.1, because

$$\begin{aligned}\mathcal{J}^{(2)}_{\text{DSM}}(\theta) &= \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\left\|\sigma_t^2\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t,t) + \boldsymbol{I} - (\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})(\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})^\top\right\|_F^2\right] &(178)\\ &= \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\mathbb{E}_{p(\boldsymbol{v})}\left[\left\|\left(\sigma_t^2\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t,t) + \boldsymbol{I} - (\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})(\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})^\top\right)\boldsymbol{v}\right\|_2^2\right] &(179)\\ &= \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\mathbb{E}_{p(\boldsymbol{v})}\left[\left\|\sigma_t^2\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t,t)\boldsymbol{v} + \boldsymbol{v} - (\sigma_t\hat{\boldsymbol{s}}_1\cdot\boldsymbol{v} + \boldsymbol{\epsilon}\cdot\boldsymbol{v})(\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})\right\|_2^2\right] &(180)\\ &= \mathcal{J}^{(2)}_{\text{DSM,estimation}}(\theta). &(181)\end{aligned}$$

And the estimated trace of second-order DSM objective in Eqn. (173) can upper bound the original objective in Sec. 5.1, because

$$\begin{aligned}\mathcal{J}^{(2,\text{tr})}_{\text{DSM}}(\theta) &= \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\left|\sigma_t^2\operatorname{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t,t)) + d - \|\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon}\|_2^2\right|^2\right] &(182)\\ &= \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\left|\operatorname{tr}\left(\sigma_t^2\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t,t) + \boldsymbol{I} - (\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})(\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})^\top\right)\right|^2\right] &(183)\\ &= \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\left|\mathbb{E}_{p(\boldsymbol{v})}\left[\boldsymbol{v}^\top\left(\sigma_t^2\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t,t) + \boldsymbol{I} - (\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})(\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})^\top\right)\boldsymbol{v}\right]\right|^2\right] &(184)\\ &\le \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\mathbb{E}_{p(\boldsymbol{v})}\left[\left|\boldsymbol{v}^\top\left(\sigma_t^2\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t,t) + \boldsymbol{I} - (\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})(\sigma_t\hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon})^\top\right)\boldsymbol{v}\right|^2\right] &(185)\\ &= \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\mathbb{E}_{p(\boldsymbol{v})}\left[\left|\sigma_t^2\boldsymbol{v}^\top\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t,t)\boldsymbol{v} + \|\boldsymbol{v}\|_2^2 - |\sigma_t\hat{\boldsymbol{s}}_1\cdot\boldsymbol{v} + \boldsymbol{\epsilon}\cdot\boldsymbol{v}|^2\right|^2\right] &(186)\\ &= \mathcal{J}^{(2,\text{tr})}_{\text{DSM,estimation}}(\theta). &(187)\end{aligned}$$

And the estimated third-order DSM objective in Eqn. (174) can also upper bound the original objective in Sec. 5.1. To prove that, we firstly propose the following theorem, which presents an equivalent form of the third-order DSM objective in Eqn. (16) for score models and the corresponding derivatives.

Theorem G.1. *(Error-Bounded Third-Order DSM by score models with trace estimators) Suppose that $\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t,t)$ is an estimation for $\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t)$, and its derivative $\hat{\boldsymbol{s}}_2(\boldsymbol{x}_t,t) \coloneqq \nabla_{\boldsymbol{x}}\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t,t)$ is an estimation for $\nabla^2_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t)$. Denote the*

Jacobian-vector-product of $\hat{\boldsymbol{s}}_1$ as $\hat{\boldsymbol{s}}_{jvp}(\boldsymbol{x}_t, t, \boldsymbol{v}) \coloneqq \nabla_{\boldsymbol{x}} \hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t)\boldsymbol{v}$. Assume we have a neural network $\boldsymbol{s}_\theta(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ parameterized by θ, then we can learn a third-order score model $\nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)) : \mathbb{R}^d \to \mathbb{R}^d$ which minimizes

$$\mathbb{E}_{q_t(\boldsymbol{x}_t)}\left[\left\|\nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)) - \nabla_{\boldsymbol{x}} \operatorname{tr}\left(\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right)\right\|_2^2\right],$$

by optimizing

$$\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\frac{1}{\sigma_t^6}\left\|\sigma_t^3 \nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)) + \mathbb{E}_{p(\boldsymbol{v})}\left[\boldsymbol{\ell}_3(\boldsymbol{x}_t, t, \boldsymbol{\epsilon}, \boldsymbol{v})\right]\right\|_2^2\right] \tag{188}$$

where

$$\begin{aligned}
&\boldsymbol{\ell}_3(\boldsymbol{x}_t, t, \boldsymbol{\epsilon}, \boldsymbol{v}) \coloneqq |\sigma_t \hat{\boldsymbol{s}}_1 \cdot \boldsymbol{v} + \boldsymbol{\epsilon} \cdot \boldsymbol{v}|^2 (\sigma_t \hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon}) - (\sigma_t^2 \boldsymbol{v}^\top \hat{\boldsymbol{s}}_{jvp} + \|\boldsymbol{v}\|_2^2)(\sigma_t \hat{\boldsymbol{s}}_1 + \boldsymbol{\epsilon}) - 2(\sigma_t \hat{\boldsymbol{s}}_1 \cdot \boldsymbol{v} + \boldsymbol{\epsilon} \cdot \boldsymbol{v})(\sigma_t^2 \hat{\boldsymbol{s}}_{jvp} + \boldsymbol{v}) \\
&\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \\
&\boldsymbol{v} \sim p(\boldsymbol{v}),\ \mathbb{E}_{p(\boldsymbol{v})}[\boldsymbol{v}] = \boldsymbol{0},\ \mathit{Cov}_{p(\boldsymbol{v})}[\boldsymbol{v}] = \boldsymbol{I}.
\end{aligned} \tag{189}$$

Denote the first-order score matching error as $\delta_1(\boldsymbol{x}_t, t) \coloneqq \|\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)\|_2$ and the second-order score matching errors as $\delta_2(\boldsymbol{x}_t, t) \coloneqq \|\nabla_{\boldsymbol{x}} \hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\|_F$ and $\delta_{2,tr}(\boldsymbol{x}_t, t) \coloneqq |\operatorname{tr}(\nabla_{\boldsymbol{x}} \hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t)) - \operatorname{tr}\left(\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right)|$. Then $\forall \boldsymbol{x}_t, \theta$, the score matching error for $\nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t))$ can be bounded by:

$$\begin{aligned}
&\left\|\nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)) - \nabla_{\boldsymbol{x}} \operatorname{tr}\left(\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right)\right\|_2 \\
&\leq \|\nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)) - \nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t, t))\|_2 + \left(\delta_1^2 + \delta_{2,tr} + 2\delta_2\right)\delta_1^2
\end{aligned}$$

Proof. For simplicity, we denote $q_{0t} \coloneqq q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)$, $q_{t0} \coloneqq q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)$, $q_t \coloneqq q_t(\boldsymbol{x}_t)$, $q_0 \coloneqq q_0(\boldsymbol{x}_0)$, $\hat{\boldsymbol{s}}_1 \coloneqq \hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t)$, $\hat{\boldsymbol{s}}_{jvp} \coloneqq \hat{\boldsymbol{s}}_{jvp}(\boldsymbol{x}_t, t)$, $\nabla \operatorname{tr}(\nabla \boldsymbol{s}(\theta)) \coloneqq \nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t))$.

As $\nabla \log q_{0t} = -\frac{\boldsymbol{\epsilon}}{\sigma_t}$ and $\nabla^2 \log q_{0t} = -\frac{1}{\sigma_t^2}\boldsymbol{I}$, by rewriting the objective in Eqn. (188), the optimization is equivalent to:

$$\begin{aligned}
\theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{q_t}\mathbb{E}_{q_{t0}}\Bigg[\Bigg\|&\nabla \operatorname{tr}(\nabla \boldsymbol{s}(\theta)) - \mathbb{E}_{p(\boldsymbol{v})}\Bigg[\left|(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1)^\top \boldsymbol{v}\right|^2 (\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1) \\
&- \left(\boldsymbol{v}^\top (\nabla \hat{\boldsymbol{s}}_1 - \nabla^2 \log q_{0t})\boldsymbol{v}\right)(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1) - 2\left((\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1)^\top \boldsymbol{v}\right)(\nabla \hat{\boldsymbol{s}}_1 - \nabla^2 \log q_{0t})\boldsymbol{v}\Bigg]\Bigg\|_2^2\Bigg]
\end{aligned} \tag{190}$$

For fixed t and $\boldsymbol{x}_t$, minimizing the inner expectation is a minimum mean square error problem for $\nabla \operatorname{tr}(\nabla \boldsymbol{s}(\theta))$, so the optimal θ^* satisfies

$$\nabla \operatorname{tr}(\nabla \boldsymbol{s}(\theta^*)) = \mathbb{E}_{q_{t0}}\mathbb{E}_{p(\boldsymbol{v})}\Big[\left|(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1)^\top \boldsymbol{v}\right|^2 (\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1) \tag{191}$$

$$- \left(\boldsymbol{v}^\top (\hat{\boldsymbol{s}}_2 - \nabla^2 \log q_{0t})\boldsymbol{v}\right)(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1) - 2\left((\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1)^\top \boldsymbol{v}\right)(\hat{\boldsymbol{s}}_2 - \nabla^2 \log q_{0t})\boldsymbol{v}\Big]. \tag{192}$$

Denote

$$\boldsymbol{s}_3(\boldsymbol{v}, \theta^*) \coloneqq \mathbb{E}_{q_{t0}}\Big[\left|(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1)^\top \boldsymbol{v}\right|^2 (\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1) \tag{193}$$

$$- \left(\boldsymbol{v}^\top (\hat{\boldsymbol{s}}_2 - \nabla^2 \log q_{0t})\boldsymbol{v}\right)(\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1) - 2\left((\nabla \log q_{0t} - \hat{\boldsymbol{s}}_1)^\top \boldsymbol{v}\right)(\hat{\boldsymbol{s}}_2 - \nabla^2 \log q_{0t})\boldsymbol{v}\Big], \tag{194}$$

then we have $\mathbb{E}_{p(\boldsymbol{v})}[\boldsymbol{s}_3(\boldsymbol{v}, \theta^*)] = \nabla \operatorname{tr}(\nabla \boldsymbol{s}(\theta^*))$. Similar to the proof in Appendix. F.3, combining with Lemma F.5, we have

$$
\begin{aligned}
\boldsymbol{s}_3(\boldsymbol{v},\theta^*) - \nabla(\boldsymbol{v}^\top\nabla^2\log q_t\boldsymbol{v}) &= \boldsymbol{s}_3(\boldsymbol{v},\theta^*) - \mathbb{E}_{q_{t0}}\left[|(\nabla\log q_{0t}-\nabla\log q_t)^\top\boldsymbol{v}|^2(\nabla\log q_{0t}-\nabla\log q_t)\right] \\
&= 2\left((\nabla\log q_t - \hat{\boldsymbol{s}}_1)^\top\boldsymbol{v}\right)(\nabla\log q_t\nabla\log q_t^\top + \nabla^2\log q_t - \hat{\boldsymbol{s}}_2)\boldsymbol{v} \\
&\quad + \boldsymbol{v}^\top(\nabla\log q_t\nabla\log q_t^\top + \nabla^2\log q_t - \hat{\boldsymbol{s}}_2)\boldsymbol{v}(\nabla\log q_t - \hat{\boldsymbol{s}}_1) \\
&\quad + \boldsymbol{v}^\top(\hat{\boldsymbol{s}}_1\hat{\boldsymbol{s}}_1^\top - \nabla\log q_t\nabla\log q_t^\top)\boldsymbol{v}\nabla\log q_t \\
&\quad + 2\left((\hat{\boldsymbol{s}}_1^\top\boldsymbol{v})(\nabla\log q_t^\top\boldsymbol{v})\hat{\boldsymbol{s}}_1 - (\nabla\log q_t^\top\boldsymbol{v})^2\nabla\log q_t\right) \\
&\quad + (\nabla\log q_t^\top\boldsymbol{v})^2\nabla\log q_t - (\hat{\boldsymbol{s}}_1^\top\boldsymbol{v})^2\hat{\boldsymbol{s}}_1 \\
&= \left(2(\nabla^2\log q_t - \hat{\boldsymbol{s}}_2)\boldsymbol{v}\boldsymbol{v}^\top + \boldsymbol{v}^\top(\nabla^2\log q_t - \hat{\boldsymbol{s}}_2)\boldsymbol{v}\right)(\nabla\log q_t - \hat{\boldsymbol{s}}_1) \\
&\quad + \left|(\nabla\log q_t - \hat{\boldsymbol{s}}_1)^\top\boldsymbol{v}\right|^2(\nabla\log q_t - \hat{\boldsymbol{s}}_1)
\end{aligned} \tag{195}
$$

As $\mathbb{E}_{p(\boldsymbol{v})}\left[\boldsymbol{v}^\top A\boldsymbol{v}\right] = \operatorname{tr}(A)$ and $\mathbb{E}_{p(\boldsymbol{v})}\left[\boldsymbol{v}\boldsymbol{v}^\top\right] = \mathbb{E}\left[\boldsymbol{v}\right]\mathbb{E}\left[\boldsymbol{v}\right]^\top + \mathrm{Cov}\left[\boldsymbol{v}\right] = \boldsymbol{I}$, we have

$$
\begin{aligned}
\nabla\operatorname{tr}(\nabla\boldsymbol{s}(\theta^*)) - \nabla\operatorname{tr}\left(\nabla^2\log q_t\right) &= \mathbb{E}_{p(\boldsymbol{v})}\left[\boldsymbol{s}_3(\boldsymbol{v},\theta^*) - \nabla(\boldsymbol{v}^\top\nabla^2\log q_t\boldsymbol{v})\right] && (196)\\
&= \left(2\left(\nabla^2\log q_t - \hat{\boldsymbol{s}}_2\right) + \left(\operatorname{tr}\left(\nabla^2\log q_t\right) - \operatorname{tr}(\hat{\boldsymbol{s}}_2)\right)\right)(\nabla\log q_t - \hat{\boldsymbol{s}}_1) \\
&\quad + \|\nabla\log q_t - \hat{\boldsymbol{s}}_1\|_2^2(\nabla\log q_t - \hat{\boldsymbol{s}}_1) && (197)
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
\|\nabla\operatorname{tr}(\nabla\boldsymbol{s}(\theta)) - \nabla\operatorname{tr}\left(\nabla^2\log q_t\right)\|_2 &\le \|\nabla\operatorname{tr}(\nabla\boldsymbol{s}(\theta)) - \nabla\operatorname{tr}(\nabla\boldsymbol{s}(\theta^*))\|_2 + \|\nabla\operatorname{tr}(\nabla\boldsymbol{s}(\theta^*)) - \nabla\operatorname{tr}\left(\nabla^2\log q_t\right)\|_2 && (198)\\
&\le \|\nabla\operatorname{tr}(\nabla\boldsymbol{s}(\theta)) - \nabla\operatorname{tr}(\nabla\boldsymbol{s}(\theta^*))\|_2 + \|\nabla\log q_t - \hat{\boldsymbol{s}}_1\|_2^3 && (199)\\
&\quad + \left(2\left\|\nabla^2\log q_t - \hat{\boldsymbol{s}}_2\right\|_F + \left|\operatorname{tr}\left(\nabla^2\log q_t\right) - \operatorname{tr}(\hat{\boldsymbol{s}}_2)\right|\right)\|\nabla\log q_t - \hat{\boldsymbol{s}}_1\|_2, && (200)
\end{aligned}
$$

which completes the proof. □

Below we show that the objective in Eqn. (16) in Theorem 4.3 is equivalent to the objective in Eqn. (188) in Theorem G.1.

Corollary G.2. *Suppose that $\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t,t)$ is an estimation for $\nabla_{\boldsymbol{x}}\log q_t(\boldsymbol{x}_t)$, and its derivative $\hat{\boldsymbol{s}}_2(\boldsymbol{x}_t,t) \coloneqq \nabla_{\boldsymbol{x}}\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t,t)$ is an estimation for $\nabla_{\boldsymbol{x}}^2\log q_t(\boldsymbol{x}_t)$. Denote the Jacobian-vector-product of $\hat{\boldsymbol{s}}_1$ as $\hat{\boldsymbol{s}}_{jvp}(\boldsymbol{x}_t,t,\boldsymbol{v}) \coloneqq \nabla_{\boldsymbol{x}}\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t,t)\boldsymbol{v}$. Assume we have a neural network $\boldsymbol{s}_\theta(\cdot,t):\mathbb{R}^d\to\mathbb{R}^d$ parameterized by θ, and we learn a third-order score model $\nabla_{\boldsymbol{x}}\operatorname{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t,t)):\mathbb{R}^d\to\mathbb{R}^d$ by the third-order DSM objectives. Then the objective in Eqn. (188) is equivalent to the objective in Eqn. (16) w.r.t. the optimization for θ.*

Proof. For simplicity, we denote $q_{0t}\coloneqq q_{0t}(\boldsymbol{x}_t|\boldsymbol{x}_0)$, $q_{t0}\coloneqq q_{t0}(\boldsymbol{x}_0|\boldsymbol{x}_t)$, $q_t\coloneqq q_t(\boldsymbol{x}_t)$, $q_0\coloneqq q_0(\boldsymbol{x}_0)$, $\hat{\boldsymbol{s}}_1\coloneqq\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t,t)$, $\hat{\boldsymbol{s}}_{jvp}\coloneqq\hat{\boldsymbol{s}}_{jvp}(\boldsymbol{x}_t,t)$, $\nabla\operatorname{tr}(\nabla\boldsymbol{s}(\theta))\coloneqq\nabla_{\boldsymbol{x}}\operatorname{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t,t))$.

On the one hand, by Eqn. (197), the optimal solution of the objective in Eqn. (188) is:

$$
\begin{aligned}
\nabla\operatorname{tr}(\nabla\boldsymbol{s}(\theta^*)) = \nabla\operatorname{tr}\left(\nabla^2\log q_t\right) &+ \left(2\left(\nabla^2\log q_t - \hat{\boldsymbol{s}}_2\right) + \left(\operatorname{tr}\left(\nabla^2\log q_t\right) - \operatorname{tr}(\hat{\boldsymbol{s}}_2)\right)\right)(\nabla\log q_t - \hat{\boldsymbol{s}}_1) \\
&+ \|\nabla\log q_t - \hat{\boldsymbol{s}}_1\|_2^2(\nabla\log q_t - \hat{\boldsymbol{s}}_1),
\end{aligned} \tag{201}
$$

so by the property of least mean square error, the objective in Eqn. (188) w.r.t. the optimization of θ is equivalent to

$$
\min_\theta \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\left\|\nabla_{\boldsymbol{x}}\operatorname{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t,t)) - \nabla_{\boldsymbol{x}}\operatorname{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t,t))\right\|_2^2\right]. \tag{202}
$$

On the other hand, by Eqn. (155), the optimal solution of the objective in Eqn. (16) is also:

$$
\begin{aligned}
\nabla\operatorname{tr}(\nabla\boldsymbol{s}(\theta^*)) = \nabla\operatorname{tr}\left(\nabla^2\log q_t\right) &+ \left(2\left(\nabla^2\log q_t - \hat{\boldsymbol{s}}_2\right) + \left(\operatorname{tr}\left(\nabla^2\log q_t\right) - \operatorname{tr}(\hat{\boldsymbol{s}}_2)\right)\right)(\nabla\log q_t - \hat{\boldsymbol{s}}_1) \\
&+ \|\nabla\log q_t - \hat{\boldsymbol{s}}_1\|_2^2(\nabla\log q_t - \hat{\boldsymbol{s}}_1),
\end{aligned} \tag{203}
$$

so by the property of least mean square error, the objective in Eqn. (16) w.r.t. the optimization of θ is also equivalent to

$$
\min_\theta \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\left\|\nabla_{\boldsymbol{x}}\operatorname{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x}_t,t)) - \nabla_{\boldsymbol{x}}\operatorname{tr}(\nabla_{\boldsymbol{x}}\boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t,t))\right\|_2^2\right]. \tag{204}
$$

Therefore, the two objectives in Eqn. (16) and Eqn. (188) are equivalent w.r.t. θ. □

Therefore, by Corollary G.2, we can derive an equivalent formulation of $\mathcal{J}_{\text{DSM}}^{(3)}(\theta)$ by Theorem. G.1:

$$\mathcal{J}_{\text{DSM}}^{(3)}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\left\|\sigma_t^3 \nabla_{\boldsymbol{x}} \text{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t,t)) + \mathbb{E}_{p(\boldsymbol{v})}\left[\boldsymbol{\ell}_3(\boldsymbol{x}_t,t,\boldsymbol{\epsilon},\boldsymbol{v})\right]\right\|_2^2\right], \tag{205}$$

where $\boldsymbol{\ell}_3(\boldsymbol{x}_t,t,\boldsymbol{\epsilon},\boldsymbol{v})$ is defined in Eqn. (189). Thus, we have

$$\begin{aligned}\mathcal{J}_{\text{DSM}}^{(3)}(\theta) &= \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\left\|\sigma_t^3 \mathbb{E}_{p(\boldsymbol{v})}\left[\nabla_{\boldsymbol{x}}\left(\boldsymbol{v}^\top \nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t,t)\boldsymbol{v}\right)\right] + \mathbb{E}_{p(\boldsymbol{v})}\left[\boldsymbol{\ell}_3(\boldsymbol{x}_t,t,\boldsymbol{\epsilon},\boldsymbol{v})\right]\right\|_2^2\right] \\ &\leq \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\mathbb{E}_{p(\boldsymbol{v})}\left[\left\|\sigma_t^3 \nabla_{\boldsymbol{x}}\left(\boldsymbol{v}^\top \nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t,t)\boldsymbol{v}\right) + \boldsymbol{\ell}_3(\boldsymbol{x}_t,t,\boldsymbol{\epsilon},\boldsymbol{v})\right\|_2^2\right] \\ &= \mathcal{J}_{\text{DSM,estimation}}^{(3)}(\theta)\end{aligned} \tag{206}$$

H. Training algorithm

In this section, we propose our training algorithm for high-dimensional data, based on the high-order DSM objectives in Appendix. G.

Let 'JVP' and 'VJP' denote forward-mode Jacobian-vector-product and reverse-mode vector-Jacobian-product with auxiliary data. More specifically, suppose $\boldsymbol{f}$ is a function $\mathbb{R}^d \to \mathbb{R}^d$, and fn($\boldsymbol{x}$) = ($\boldsymbol{f}(\boldsymbol{x})$, aux), where aux is auxiliary data, then JVP(fn, $\boldsymbol{x}_0$, $\boldsymbol{v}$) = ($\boldsymbol{f}(\boldsymbol{x}_0)$, $\nabla \boldsymbol{f}|_{\boldsymbol{x}=\boldsymbol{x}_0}\boldsymbol{v}$, aux) and VJP(fn, $\boldsymbol{x}_0$, $\boldsymbol{v}$) = ($\boldsymbol{f}(\boldsymbol{x}_0)$, $\boldsymbol{v}^\top \nabla \boldsymbol{f}|_{\boldsymbol{x}=\boldsymbol{x}_0}$, aux).

Algorithm 1 Training of high-order denoising score matching

Require: score network $\boldsymbol{s}_\theta$, hyperparameters λ_1, λ_2, smallest time ϵ, forward SDE, proposal distribution $p(t)$
Input: sample $\boldsymbol{x}_0$ from data distribution
Output: denoising score matching loss $\mathcal{J}_{\text{DSM}}$
  Sample $\boldsymbol{\epsilon}$ from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  Sample $\boldsymbol{v}$ from standard Gaussian or Rademacher distribution
  Sample t from proposal distribution $p(t)$ ($\mathcal{U}(\epsilon, 1)$ for VE SDEs)
  Get mean α_t and std σ_t at time t from the forward SDE.
  $\boldsymbol{x}_t \leftarrow \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$
  def grad_div_fn($\boldsymbol{x}_t$, t, $\boldsymbol{v}$):
    def score_jvp_fn($\boldsymbol{x}_t$):
      $\boldsymbol{s}_\theta \leftarrow$ lambda $\boldsymbol{x}$: $\boldsymbol{s}_\theta(\boldsymbol{x}, t)$
      $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t), \boldsymbol{s}_{jvp} \leftarrow$ JVP($\boldsymbol{s}_\theta$, $\boldsymbol{x}_t$, $\boldsymbol{v}$)
      return $\boldsymbol{s}_{jvp}$, $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$
    $\boldsymbol{s}_{jvp}$, $\boldsymbol{v}^\top \nabla \boldsymbol{s}_{jvp}$, $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \leftarrow$ VJP(score_jvp_fn, $\boldsymbol{x}_t$, $\boldsymbol{v}$)
    return $\boldsymbol{s}_{jvp}$, $\boldsymbol{v}^\top \nabla \boldsymbol{s}_{jvp}$, $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$
  $\boldsymbol{s}_{jvp}$, $\boldsymbol{v}^\top \nabla \boldsymbol{s}_{jvp}$, $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t) \leftarrow$ grad_div_fn($\boldsymbol{x}_t$, t, $\boldsymbol{v}$)
  $\hat{\boldsymbol{s}}_1 \leftarrow$ stop_gradient($\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$)
  $\hat{\boldsymbol{s}}_{jvp} \leftarrow$ stop_gradient($\boldsymbol{s}_{jvp}$)
  Calculate $\mathcal{J}_{\text{DSM}}^{(1)}(\theta), \mathcal{J}_{\text{DSM}}^{(2)}(\theta), \mathcal{J}_{\text{DSM}}^{(2,\text{tr})}(\theta), \mathcal{J}_{\text{DSM}}^{(3)}(\theta)$ from Eqn. (18) (172) (173) (174)
  return $\mathcal{J}_{\text{DSM}}^{(1)}(\theta) + \lambda_1\left(\mathcal{J}_{\text{DSM}}^{(2)}(\theta) + \mathcal{J}_{\text{DSM}}^{(2,\text{tr})}(\theta)\right) + \lambda_2 \mathcal{J}_{\text{DSM}}^{(3)}(\theta)$

I. Experiment details

I.1. Choosing of λ_1, λ_2

As we use Monto-Carlo method to unbiasedly estimate the expectations, we empirically find that we need to ensure the mean values of $\mathcal{J}_{\text{DSM}}^{(1)}(\theta)$, $\lambda_1\left(\mathcal{J}_{\text{DSM}}^{(2)}(\theta) + \mathcal{J}_{\text{DSM}}^{(2,\text{tr})}(\theta)\right)$ and $\lambda_2 \mathcal{J}_{\text{DSM}}^{(3)}(\theta)$ are in the same order of magnitude. Therefore, for synthesis data experiments, we choose $\lambda_1 = 0.5$ and $\lambda_2 = 0.1$ for our final objectives. And for CIFAR-10 experiments, we simply choose $\lambda_1 = \lambda_2 = 1$ with no further tuning.

Specifically, for synthesis data, we choose $\lambda_1 = 0.5$, $\lambda_2 = 0$ as the second-order score matching objective, and $\lambda_1 = 0.5$, $\lambda_2 = 0.1$ as the third-order score matching objective. For CIFAR-10, we simply choose $\lambda_1 = 1$, $\lambda_2 = 0$ as the second-order score matching objective, and $\lambda_1 = \lambda_2 = 1$ as the third-order score matching objective.

In practice, the implementation of the first-order DSM objective in (Song et al., 2020; 2021) divides the loss function by the data dimension (i.e. use $\frac{\|\cdot\|_2^2}{d}$ instead of $\|\cdot\|_2^2$). We also divide the second-order DSM and the third-order DSM objectives by the data dimension for the image data to balance the magnitudes of the first, second and third-order objectives. Please refer to the implementation in our released code for details.

I.2. 1-D mixture of Gaussians

We exactly compute the high-order score matching objectives in Sec. 5.1, and choose the starting time $\epsilon = 10^{-5}$.

The density function of the data distribution is

$$q_0(\boldsymbol{x}_0) = 0.4 * \mathcal{N}(\boldsymbol{x}_0| -\frac{2}{9}, \frac{1}{9^2}) + 0.4 * \mathcal{N}(\boldsymbol{x}_0| -\frac{2}{3}, \frac{1}{9^2}) + 0.2 * \mathcal{N}(\boldsymbol{x}_0|\frac{4}{9}, \frac{2}{9^2}) \tag{207}$$

We use the "noise-prediction" type model (Kingma et al., 2021), i.e. we use a neural network $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ to model $\sigma_t \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$. We use the time-embedding in Song et al. (2020). For the score model, we use a two-layer MLP to encode t, and a two-layer MLP to encode the input $\boldsymbol{x}_t$, then concatenate them together to another two-layer MLP network to output the predicted noise $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$.

We use Adam (Kingma & Ba, 2014) optimizer with the default settings in JAX. The batch size is 5000. We train the model for 50k iterations by one NVIDIA GeForece RTX 2080 Ti GPU card.

As our proposed high-order DSM algorithm has the property of bounded errors, we directly train our model from the default initialized neural network, without any pre-training for the lower-order models.

I.3. 2-D checkerboard data

We exactly compute the high-order score matching objectives in Sec. 5.1, and choose the starting time $\epsilon = 10^{-5}$.

We use the same neural network and hyperparameters in Appendix. I.2, and we train for 100k iterations for all experiments by one NVIDIA GeForece RTX 2080 Ti GPU card.

Similar to Appendix. I.2, we directly train our model from the default initialized neural network, without any pre-training for the lower-order models.

I.4. CIFAR-10 experiments

Our code of CIFAR-10 experiments are based on the released code of Song et al. (2020) and Song et al. (2021). We choose the start time $\epsilon = 10^{-5}$ for both training and evaluation.

We train the score model by the proposed high-order DSM objectives both from 0 iteration and from the pre-trained checkpoints in Song et al. (2020) to a fixed ending iteration, and we empirically find that the model performance by these two training procedure are nearly the same. Therefore, to save time, we simply use the pre-trained checkpoints to further train the second-order and third-order models by a few iteration steps and report the results of the fine-tuning models. Moreover, we find that further train the pre-trained checkpoints by first-order DSM cannot improve the likelihood of ScoreODE, so we simply use the pre-trained checkpoints to evaluate the first-order models.

Model architectures The model architectures are the same as (Song et al., 2020). For VE, we use NCSN++ cont. which has 4 residual blocks per resolution. For VE (deep), we use NCSN++ cont. deep which has 8 residual blocks per resolution. Also, our network is the "noise-prediction" type, same as the implementation in Song et al. (2020).

Training We follow the same training procedure and default settings for score-based models as (Song et al., 2020), and set the exponential moving average (EMA) rate to 0.999 as advised. Also as in (Song et al., 2020), the input images are pre-processed to be normalized to $[0, 1]$ for VESDEs. For all experiments, we set the "`n_jitted_steps`=1" in JAX code.

For the experiments of the VE model, we use 8 GPU cards of NVIDIA GeForece RTX 2080 Ti. We use the pre-trained checkpoints of 1200k iterations of Song et al. (2020), and further train 100k iterations (for about half a day) and the model quickly converges. We use a batchsize of 128 of the second-order training, and a batchsize of 48 of the third-order training.

For the experiments of the VE(deep) model, we use 8 GPU cards of Tesla P100-SXM2-16GB. We use the pre-trained

Table 3. Benchmark of computation time, iteration numbers and memory consumption on CIFAR-10.

Model	Time per iteration (s)	Memory in total (GiB)	Training iterations
VE (Song et al., 2020)	0.28	25.84	1200k
VE (second) (**ours**)	0.33	28.93	1200k(checkpoints) + 100k
VE (third) (**ours**)	0.44	49.12	1200k(checkpoints) + 100k

checkpoints of 600k iterations of Song et al. (2020), and further train 100k iterations (for about half a day) and the model quickly converges. We use a batchsize of 128 of the second-order training, and a batchsize of 48 of the third-order training.

Likelihood and sample quality We use the uniform dequantization for likelihood evaluation. For likelihood, we report the bpd on the test dataset with 5 times repeating (to reduce the variance of the trace estimator). For sampling, we find the ode sampler often produce low quality images and instead use the PC sampler (Song et al., 2020) discretized at 1000 time steps to generate 50k samples and report the FIDs on them. We use the released pre-trained checkpoints of the VESDE in Song et al. (2020) to evaluate the likelihood and sample quality of the first-order score matching models.

Computation time, iteration numbers and memory consumption We list the computation time and memory consumption of the VE model (shallow model) on 8 NVIDIA GeForce RTX 2080 Ti GPU cards with batch size 48 in Table 3. The `n_jitted_steps` is set to 1. When training by second-order DSM, we only use `score_jvp_fn` in the training algorithm and remove `grad_div_fn`. The computation time is averaged over 10k iterations. We use `jax.profiler` to trace the GPU memory usage during training, and report the peak total memory on 8 GPUs.

We find that the costs of the second-order DSM training are close to the first-order DSM training, and the third-order DSM training costs less than twice of the first-order training. Nevertheless, our method can scale up to high-dimensional data and improve the likelihood of ScoreODEs.

I.5. ImageNet 32x32 experiments

We adopt the same start time and model architecture for both VE and VE (deep) as CIFAR-10 experiments. Note that the released code of Song et al. (2020) and Song et al. (2021) provides no pretrained checkpoint of VE type for ImageNet 32x32 dataset, so we use their training of VP type as a reference, and train the first-order VE models from scratch. Specifically, we train the VE baseline for 1200k iterations, and the VE (deep) baseline for 950k iterations.

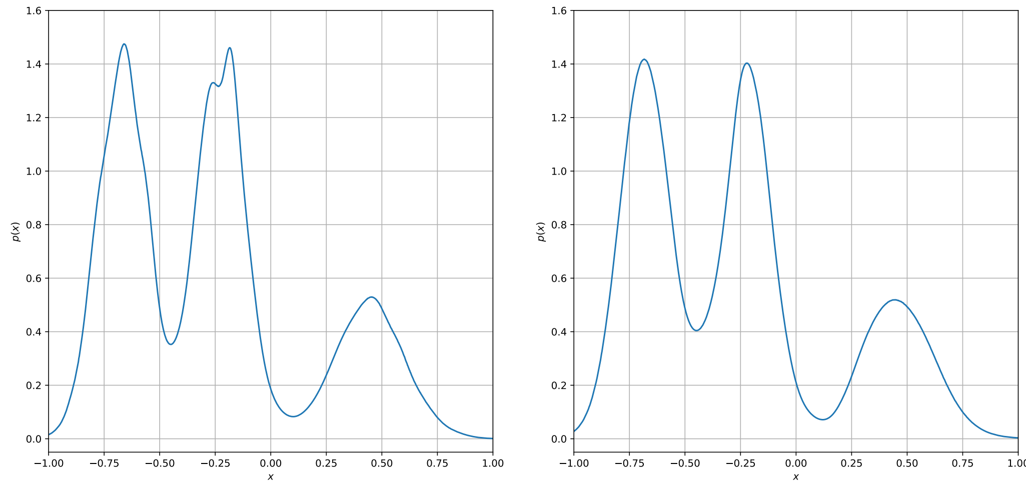
For the high-order experiments of both VE and VE (deep), we further train 100k iterations, using a batchsize of 128 for the second-order training and a batchsize of 48 for the third-order training. We use 8 GPU cards of NVIDIA GeForece RTX 2080 Ti for VE, and 8 GPU cards of NVIDIA A40 for VE (deep). We report the average bpd on the test dataset with 5 times repeating.

J. Additional results for VE, VP and subVP types

We also train VP and subVP types of ScoreODEs by the proposed high-order DSM method, with the maximum likelihood weighting function for ScoreSDE in (Song et al., 2021) (the weighting functions are detailed in (Song et al., 2021, Table 1)). Note that for the VE type, the likelihood weighting is exactly the DSM weighting used in our experiments.

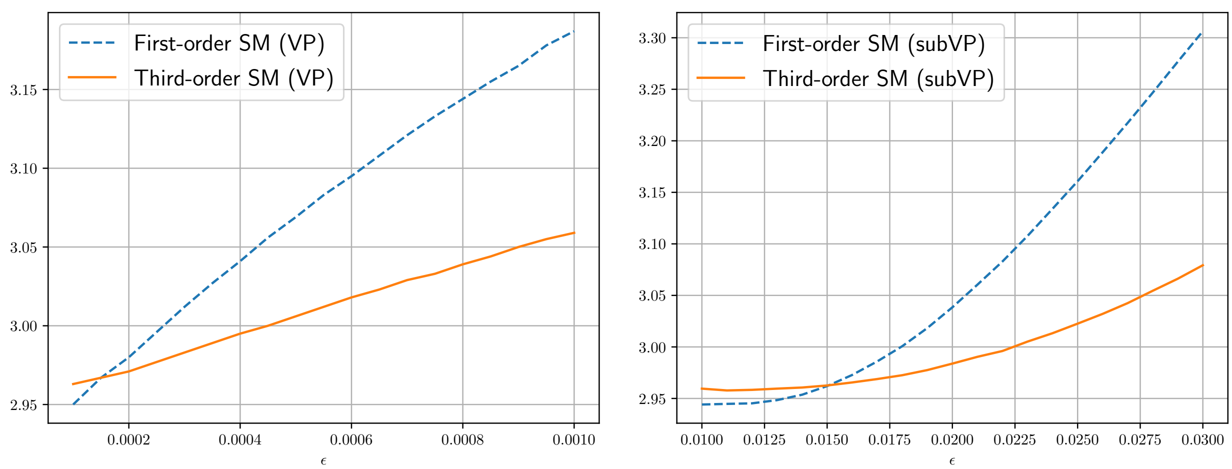
We firstly show that for the ScoreODE of VP type, even on the simple 1-D mixture-of-Gaussians, the model density trained by the first-order DSM is not good enough and can be improved by the third-order DSM, as shown in Fig. 5.

We then train ScoreODE of VP and subVP types on the CIFAR-10 dataset. We use the pretrained checkpoint by first-order DSM in (Song et al., 2021) (the checkpoint of "Baseline+LW+IS"), and further train 100k iterations. We use the same experiment settings as the VE experiments. We vary the start evaluation time ϵ and evaluate the model likelihood at each ϵ, as shown in Fig. 6. For the ScoreODE trained by the first-order DSM, the model likelihood is poor when ϵ is slightly large. Note that Song et al. (2020; 2021) uses $\epsilon = 10^{-3}$ for sampling of VP type, and the corresponding likelihood is poor. Moreover, our proposed method can improve the likelihood for ϵ larger than a certain value (e.g. our method can improve the likelihood for the VP type with $\epsilon = 10^{-3}$). However, for very small ϵ, our method cannot improve the likelihood for VP and subVP. We suspect that it is because of the "unbounded score" problem (Dockhorn et al., 2021). The first-order score



(e) First-order DSM, VP (f) Third-order DSM, VP

Figure 5. Model density of ScoreODEs (VP type) on 1-D mixture-of-Gaussians data.



(a) CIFAR-10 bpd (VP) (b) CIFAR-10 bpd (subVP)

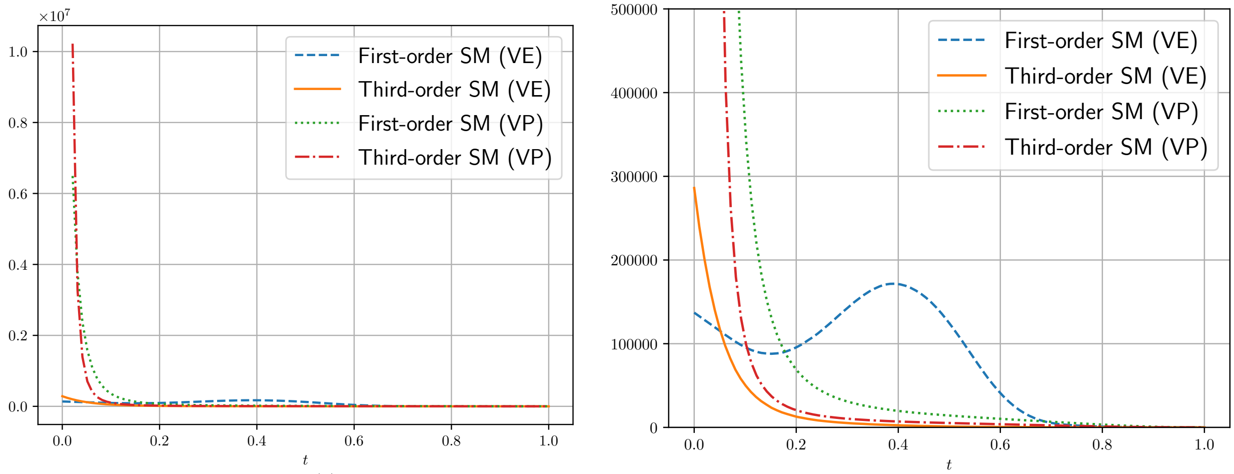
Figure 6. Model NLL (negative-log-likelihood) in bits/dim (bpd) of ScoreODEs for both VP type and subVP type trained by first-order DSM and our proposed third-order DSM on CIFAR-10 with the likelihood weighting functions and the importance sampling in Song et al. (2021), varying the start evaluation time ϵ.

function suffers numerical issues for t near 0 and the first-order SM error is so large that it cannot provide useful information for the higher-order SM.

As the data score function $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$ is unknown for CIFAR-10 experiments, we cannot evaluate $\mathcal{J}_{\text{Fisher}}(\theta)$. To show the effectiveness of our method, we evaluate the difference between $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ and $\nabla_{\boldsymbol{x}} \log p_t^{\text{ODE}}(\boldsymbol{x}_t)$ in the $\mathcal{J}_{\text{Diff}}(\theta)$. Denote

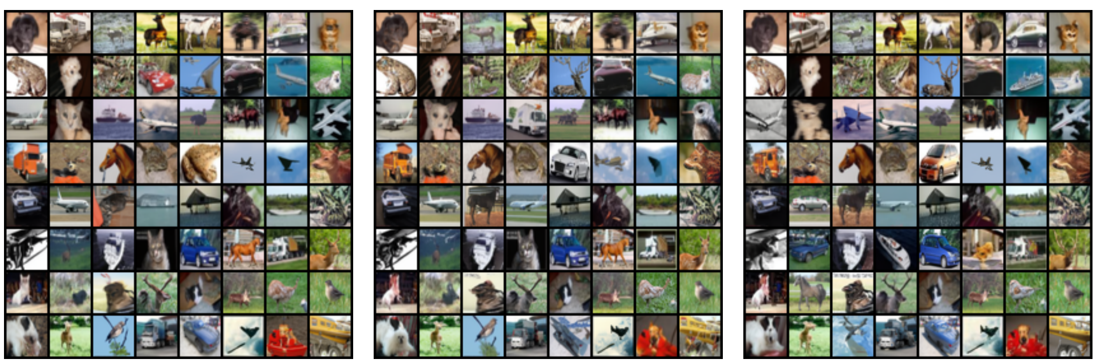
$$\ell_{\text{Diff}}(t) \coloneqq g(t)^2 \mathbb{E}_{q_t(\boldsymbol{x}_t)} \| \boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}} \log p_t^{\text{ODE}}(\boldsymbol{x}_t) \|_2^2, \tag{208}$$

we take 100 time steps between $\epsilon = 10^{-5}$ and $T = 1$ to evaluate $\ell_{\text{Diff}}(t)$, where the ODE score function $\nabla_{\boldsymbol{x}} \log p_t^{\text{ODE}}(\boldsymbol{x}_t)$ is computed by the method in Appendix. D, and the expectation w.r.t. $q_t(\boldsymbol{x}_t)$ are computed by the Monte-Carlo method, namely we use the test dataset for $q_0(\boldsymbol{x}_0)$ and then sample $\boldsymbol{x}_t$ from $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$. We evaluate the $\ell_{\text{Diff}}(t)$ for the first-order DSM training (baseline) and the third-order DSM training for both the VP and VE types, as shown in Fig. 7. We can find that our proposed high-order DSM method can reduce $\ell_{\text{Diff}}(t)$ for ScoreODEs for most $t \in [\epsilon, T]$. As $\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ is an estimation of the true data score function $\nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)$, the results indicates that our proposed method can reduce $\ell_{\text{Fisher}}(t)$ of ScoreODEs and then further reduce $\mathcal{J}_{\text{Fisher}}(\theta)$. Also, we can find that the "unbounded score" problem of VE is much milder than that of VP, which can explain why our method can greatly improve the likelihood of VE type even for very small ϵ.



(a) $\ell_{\text{Diff}}(t)$. (b) $\ell_{\text{Diff}}(t)$ (clipped by y-axis).

Figure 7. $\ell_{\text{Diff}}(t)$ of VP type and VE type on CIFAR-10, trained by first-order DSM with the likelihood weighting functions and importance sampling in Song et al. (2021) and our proposed third-order DSM.



(a) First-order score matching (b) Second-order score matching (c) Third-order score matching

Figure 8. Random samples of SGMs (VE type) by PC sampler, trained by different orders of DSM.

Moreover, we randomly select a batch of generated samples by the PC sampler (Song et al., 2020) of the same random seed by the VE model of first-order, second-order and third-order DSM training, as shown in Fig. 8. The samples are very close for human eyes, which shows that after our proposed training method, the score model can still be used for the sample methods of SGMs to generate high-quality samples.