STABLE TARGET FIELD FOR REDUCED VARIANCE SCORE ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Score-based generative models (SGMs) generate samples by reversing a fixed forward diffusion process. Despite already providing impressive empirical results, these SGM algorithms can be further improved by reducing the variance of the training targets in their score-matching objective. We argue that the source of such variance lies in the handling of intermediate noise-variance scales, where multiple modes in the data affect the direction of reverse paths. We propose to remedy the problem by incorporating *a reference batch* which we use to calculate weighted conditional scores as more stable training targets. We show that the procedure indeed helps in the challenging intermediate regime by reducing (the trace of) the covariance of training targets. The new stable targets can be seen as trading bias for reduced variance, where the bias vanishes with increasing reference batch size. Empirically, we show that the new objective improves the image quality, stability, and training speed of various popular SGMs across datasets with both general ODE and SDE solvers. When used in combination with state-of-the-art SGMs (EDM (Karras et al., 2022)), our method yields a new SOTA FID of 1.90 with 35 network evaluations on the unconditional CIFAR-10 generation task.

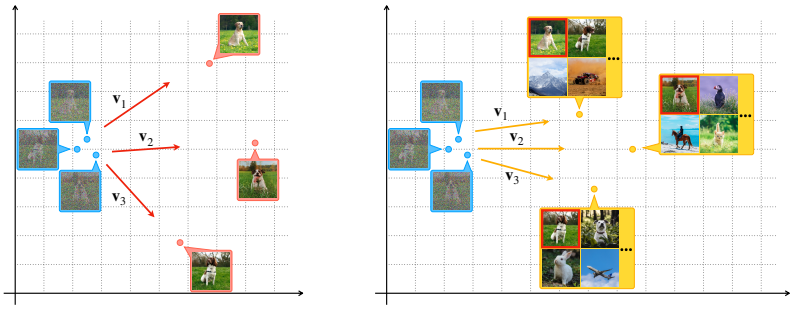
1 INTRODUCTION

Score-based generative models (SGMs) (Song & Ermon, 2019) and diffusion models (Ho et al., 2020; Sohl-Dickstein et al., 2015) have recently achieved impressive results on a wide spectrum of generative tasks, such as image generation (Nichol et al., 2022; Song et al., 2021b), 3D point cloud generation (Luo & Hu, 2021) and molecular conformer generation (Shi et al., 2021; Xu et al., 2022a). These models can be subsumed under a unified framework in the form of Itô stochastic differential equations (SDE) (Song et al., 2021b). The models learn time-dependent score fields via score-matching (Hyvärinen & Dayan, 2005), which then guides the reverse SDE during generative sampling. Popular instances of SGMs include variance-exploding (VE) and variance-preserving (VP) SDE (Song et al., 2021b). Building on these formulations, EDM (Karras et al., 2022) provides the best performance to date.

We argue that, despite achieving impressive empirical results, the current training scheme of SGMs can be further improved. In particular, the variance of training targets in the score-matching objective can be large and lead to suboptimal performance. To better understand the origin of this instability, we decompose the score field into three regimes. Our analysis shows that the phenomenon arises primarily in the intermediate regime, which is characterized by multiple modes or data points exhibiting comparable influences on the scores. In other words, in this regime, the sources of the noisy examples generated in the course of the forward process become ambiguous. We illustrate the problem in Figure 1(a), where each stochastic update of the score model is based on disparate targets.

We propose a generalized version of the vanilla score-matching objective, termed the *Stable Target Field* (**STF**) objective. The idea is to include an additional *reference batch* of examples that are used to calculate weighted conditional scores as targets. We apply self-normalized importance sampling to aggregate the contribution of each example in the reference batch. Although this process can substantially reduce the variance of training targets (Figure 1(b)), especially in the intermediate regime, it does introduce some bias. Nevertheless, we show that the bias together with the trace-of-covariance of the STF training targets shrinks to zero as we increase the size of the reference batch.

Experimentally, we show that our STF objective achieves new state-of-the-art performance on CIFAR-10 unconditional generation when incorporated into EDM (Karras et al., 2022). The resulting FID



(a) vanilla (b) STF

Figure 1: Illustration of differences between the vanilla score-matching objective and our proposed STF objective. The "destroyed" images (in blue box) are close to each other while their sources (in red box) are not. Although the true score in expectation is the weighted average of $\mathbf{v}_i$, the individual training updates of the vanilla objective have a high variance, which our STF objective reduces significantly by including a large reference batch (yellow box).

score (Heusel et al., 2017) is 1.90 with 35 network evaluations. STF also improves the FID/Inception scores for other variants of score-based models, *i.e.*, VE and VP SDEs (Song et al., 2021b), in most cases. In addition, it enhances the stability of converged score-based models on CIFAR-10 and CelebA 64^2 across random seeds, and helps avoid generating noisy images in VE. STF accelerates the training of score-based models ($3.6\times$ speed-up for VE on CIFAR-10) while obtaining comparable or better FID scores. To the best of our knowledge, STF is the first technique to accelerate the training process of SGMs. We further demonstrate the performance gain with increasing reference batch size, highlighting the negative effect of large variance.

Our contributions are summarized as follows: **(1)** We detail the instability of the current SGMs training objective in a principled and quantitative manner, characterizing a region in the forward process, termed *the intermediate phase*, where the score-learning targets are most variable (Section 3). **(2)** We propose a generalized score-matching objective, *stable target field*, which provides more stable training targets (Section 4). **(3)** We analyze the behavior of the new objective and prove that it is asymptotically unbiased and reduces the trace-of-covariance of the training targets by a factor pertaining to the reference batch size in the intermediate phase under mild conditions (Section 5). **(4)** We illustrate the theoretical arguments empirically and show that the proposed STF objective improves the performance, stability, and training speed of score-based methods. In particular, it achieves the current state-of-the-art FID score on the CIFAR-10 benchmark when combined with EDM (Section 6).

2 Background on Score-Based Generative Models

In SGMs, the forward process[1] is an SDE with no learned parameter, in the form of:

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w},$$

where $\mathbf{x} \in \mathbb{R}^d$ with $\mathbf{x}(0) \sim p_0$ being the data distribution, $t \in [0, 1]$, $\mathbf{f}\colon \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$, $g\colon [0, 1] \to \mathbb{R}$, and $\mathbf{w} \in \mathbb{R}^d$ is the standard Wiener process. It gradually transforms the data distribution to a known prior as time goes from 0 to 1. Sampling of SGMs is done via a corresponding reverse-time SDE (Anderson, 1982):

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}\bar{t} + g(t)\mathrm{d}\bar{\mathbf{w}},$$

where $\bar{\cdot}$ denotes time traveling backward from 1 to 0. Song et al. (2021b) propose a probability flow ODE that induces the same marginal distribution $p_t(\mathbf{x})$ as the SDE:

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}\bar{t}.$$

[1]For simplicity, we focus on the version where the diffusion coefficient $g(t)$ is independent of $\mathbf{x}(t)$.

Both formulations progressively recover p_0 from the prior p_1. We estimate the score of the transformed data distribution at time t, $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, via a neural network, $\mathbf{s}_\theta(\mathbf{x}, t)$. Specifically, the training objective is a weighted sum of the denoising score-matching (Vincent, 2011):

$$\min_{\theta} \mathbb{E}_{t\sim q_t(t)} \lambda(t) \mathbb{E}_{\mathbf{x}\sim p_0} \mathbb{E}_{\mathbf{x}(t)\sim p_{t|0}(\cdot|\mathbf{x})} \left[\|\mathbf{s}_\theta(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x})\|_2^2 \right], \tag{1}$$

where q_t is the distribution for time variable, *e.g.*, $\mathcal{U}[0, 1]$ for VE/VP (Song et al., 2021b), and $\lambda(t) = \sigma_t^2$ is the positive weighting function to keep the time-dependent loss at the same magnitude (Song et al., 2021b), and $p_{t|0}(\mathbf{x}(t)|\mathbf{x})$ is the transition kernel denoting the conditional distribution of $\mathbf{x}(t)$ given $\mathbf{x}$[2]. Specifically, SGMs "destroy" data according to a diffusion process utilizing Gaussian transition kernels, which result in $p_{t|0}(\mathbf{x}(t)|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I})$. Recent works (Xu et al., 2022b; Rissanen et al., 2022) have also extended the underlying principle from the diffusion process to more general physical processes where the training objective is not necessarily score-related.

3 Understanding the Training Target in Score-Matching Objective

The vanilla denoising score-matching objective at time t is:

$$\ell_{\mathrm{v}}(\theta, t) = \mathbb{E}_{p_0(\mathbf{x})} \mathbb{E}_{p_{t|0}(\mathbf{x}(t)|\mathbf{x})} [\|\mathbf{s}_\theta(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x})\|_2^2], \tag{2}$$

where the network is trained to fit the individual targets $\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x})$ at $(\mathbf{x}(t), t)$ – the "influence" exerted by clean data $\mathbf{x}$ on $\mathbf{x}(t)$. We can swap the order of the sampling process by first sampling $\mathbf{x}(t)$ from p_t and then $\mathbf{x}$ from $p_{0|t}(\cdot|\mathbf{x}(t))$. Thus, $\mathbf{s}_\theta$ has a closed form minimizer:

$$\mathbf{s}_{\mathrm{v}}^*(\mathbf{x}(t), t) = \mathbb{E}_{p_{0|t}(\mathbf{x}|\mathbf{x}(t))} [\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x})] = \nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t)). \tag{3}$$

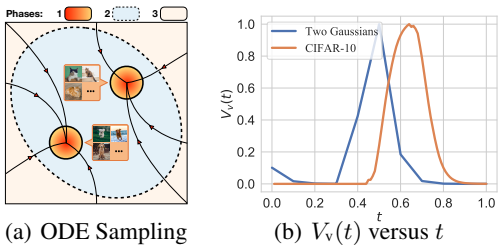
The score field is a conditional expectation of $\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x})$ with respect to the posterior distribution $p_{0|t}$. In practice, a Monte Carlo estimate of this target can have high variance (Owen, 2013; Elvira & Martino, 2021). In particular, when multiple modes of the data distribution have comparable influences on $\mathbf{x}(t)$, $p_{0|t}(\cdot|\mathbf{x}(t))$ is a multi-mode distribution, as also observed in Xiao et al. (2022). Thus the targets $\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x})$ vary considerably across different $\mathbf{x}$ and this can strongly affect the estimated score at $(\mathbf{x}(t), t)$, resulting in slower convergence and worse performance in practical stochastic gradient optimization (Wang et al., 2013).

To quantitatively characterize the variations of individual targets at different time, we propose a metric – the average trace-of-covariance of training targets at time t:

$$\begin{aligned} V_{\mathrm{v}}(t) &= \mathbb{E}_{p_t(\mathbf{x}(t))} \left[\mathrm{Tr}(\mathrm{Cov}_{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}(\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}))) \right] \\ &= \mathbb{E}_{p_t(\mathbf{x}(t))} \mathbb{E}_{p_{0|t}(\mathbf{x}|\mathbf{x}(t))} \left[\|\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x})) - \nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t))\|_2^2 \right]. \end{aligned} \tag{4}$$

We use $V_{\mathrm{v}}(t)$ to define three successive phases relating to the behavior of training targets. As shown in Figure 2(a), the three phases partition the score field into near, intermediate, and far regimes (Phase 1∼3 respectively). Intuitively, $V_{\mathrm{v}}(t)$ peaks in the intermediate phase (Phase 2), where multiple distant modes in the data distribution have comparable influences on the same noisy perturbations, resulting in unstable targets. In Phase 1, the posterior $p_{0|t}$ concentrates around one single mode, thus low variation. In Phase 3, the targets remain similar across modes since $\lim_{t\to 1} p_{t|0}(\mathbf{x}(t)|\mathbf{x}) \approx p_1$ for commonly used transition kernels.

We validate this argument empirically in Figure 2(b), which shows the estimated $V_{\mathrm{v}}(t)$ for a mixture of two Gaussians as well as a subset of CIFAR-10 dataset (Krizhevsky et al., 2009) for a more realistic setting. Here we use VE SDE, *i.e.*, $p_{t|0}(\mathbf{x}(t)|\mathbf{x}) = \mathcal{N}\left(\mathbf{x}, \sigma_m^2 (\frac{\sigma_M}{\sigma_m})^{2t} \boldsymbol{I}\right)$ for some σ_m and σ_M (Song et al., 2021b). $V_{\mathrm{v}}(t)$ exhibits similar phase behavior across t in both toy and realistic cases. Moreover, $V_{\mathrm{v}}(t)$ reaches its maximum value in the intermediate phase, demonstrating the large variations of individual targets. We defer more details to Appendix C.



(a) ODE Sampling (b) $V_{\mathrm{v}}(t)$ versus t

Figure 2: **(a)**: An illustration of the three phases in a two-mode distribution. **(b)**: Estimated $V_{\mathrm{v}}(t)$ for two distributions. We normalize the maximum value to 1 for illustration purposes.

[2]We omit "(0)" from $\mathbf{x}(0)$ when there is no ambiguity.

4 TREATING SCORE AS A FIELD

The vanilla denoising score-matching approach (Equation 3) can be viewed as a Monte Carlo estimator, *i.e.*, $\nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t)) = \mathbb{E}_{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}[\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x})] \approx \frac{1}{n}\sum_{i=1}^{n} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_i)$ where $\mathbf{x}_i$ is sampled from $p_{0|t}(\cdot|\mathbf{x}(t))$ and $n = 1$. The variance of a Monte Carlo estimator is proportional to $\frac{1}{n}$, so we propose to use a larger batch (n) to counter the high variance problem described in Section 3. Since sampling directly from the posterior $p_{0|t}$ is not practical, we first apply importance sampling with the proposal distribution p_0. Specifically, we sample a large reference batch $\mathcal{B}_L = \{\mathbf{x}_i\}_{i=1}^{n} \sim p_0^n$ and get the following approximation:

$$\nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t)) \approx \frac{1}{n}\sum_{i=1}^{n} \frac{p_{0|t}(\mathbf{x}_i|\mathbf{x}(t))}{p_0(\mathbf{x}_i)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_i).$$

The importance weights can be rewritten as $p_{0|t}(\mathbf{x}|\mathbf{x}(t))/p_0(\mathbf{x}) = p_{t|0}(\mathbf{x}(t)|\mathbf{x})/p_t(\mathbf{x}(t))$. However, this basic importance sampling estimator has two issues. The weights now involve an unknown normalization factor $p_t(\mathbf{x}(t))$ and the ratio between the prior and posterior distribution can be large in high dimensional spaces. To remedy these problems, we appeal to self-normalization techniques (Hesterberg, 1995) to further stabilize the training targets:

$$\nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t)) \approx \sum_{i=1}^{n} \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_i)}{\sum_{j=1}^{n} p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_i). \tag{5}$$

We term this new training target in Equation 5 as *Stable Target Field (STF)*. In practice, we sample the reference batch $\mathcal{B}_L = \{\mathbf{x}_i\}_{i=1}^{n}$ from p_0^n and obtain $\mathbf{x}(t)$ by applying the transition kernel to the "first" training data $\mathbf{x}_1$. Taken together, the new STF objective becomes:

$$\ell_{\text{STF}}(\theta, t) = \mathbb{E}_{\{\mathbf{x}_i\}_{i=1}^{n} \sim p_0^n} \mathbb{E}_{\mathbf{x}(t) \sim p_{t|0}(\cdot|\mathbf{x}_1)} \left[\left\| \mathbf{s}_\theta(\mathbf{x}(t), t) - \sum_{k=1}^{n} \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{\sum_{j=1}^{n} p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) \right\|_2^2 \right]. \tag{6}$$

When $n = 1$, STF reduces to the vanilla score-matching (Equation 2). When $n > 1$, STF incorporates a reference batch to stabilize training targets. Intuitively, the new weighted target assigns larger weights to clean data with higher influence on $\mathbf{x}(t)$, *i.e.*, higher transition probability $p_{t|0}(\mathbf{x}(t)|\mathbf{x})$.

Similar to our analysis in Section 3, we can again swap the sampling process in Equation 6 so that, for a perturbation $\mathbf{x}(t)$, we sample the reference batch $\mathcal{B}_L = \{\mathbf{x}_i\}_{i=1}^{n}$ from $p_{0|t}(\cdot|\mathbf{x}(t))p_0^{n-1}$, where the first element involves the posterior, and the rest follow the data distribution. Thus, the minimizer of the new objective (Equation 6) is (derivation can be found in Appendix B.1)

$$\mathbf{s}^*_{\text{STF}}(\mathbf{x}(t), t) = \mathbb{E}_{\mathbf{x}_1 \sim p_{0|t}(\cdot|\mathbf{x}(t))} \mathbb{E}_{\{\mathbf{x}_i\}_{i=2}^{n} \sim p_0^{n-1}} \left[\sum_{k=1}^{n} \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{\sum_j p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) \right]. \tag{7}$$

Note that although STF significantly reduces the variance, it introduces bias: the minimizer is no longer the true score. Nevertheless, in Section 5, we show that the bias converges to 0 as $n \to \infty$, while reducing the trace-of-covariance of the training targets by a factor of n when $p_{0|t} \approx p_0$. We further instantiate the STF objective (Equation 6) with transition kernels in the form of $p_{t|0}(\mathbf{x}(t)|\mathbf{x}) = \mathcal{N}(\mathbf{x}, \sigma_t^2 \boldsymbol{I})$, which includes EDM (Karras et al., 2022), VP (through reparameterization) and VE (Song et al., 2021b):

$$\mathbb{E}_{\mathbf{x}_1 \sim p_{0|t}(\cdot|\mathbf{x}(t))} \mathbb{E}_{\{\mathbf{x}_i\}_{i=2}^{n} \sim p_0^{n-1}} \left[\left\| \mathbf{s}_\theta(\mathbf{x}(t), t) - \frac{1}{\sigma_t^2} \sum_{k=1}^{n} \frac{\exp\left(-\frac{\|\mathbf{x}(t)-\mathbf{x}_k\|_2^2}{2\sigma_t^2}\right)}{\sum_j \exp\left(-\frac{\|\mathbf{x}(t)-\mathbf{x}_j\|_2^2}{2\sigma_t^2}\right)} (\mathbf{x}_k - \mathbf{x}(t)) \right\|_2^2 \right].$$

To aggregate the time-dependent STF objective over t, we uniformly sample the time variable t from $[0, 1]$ during training and apply the weighting function $\lambda(t)$. Together, the final training objective for STF is $\mathbb{E}_{t \sim q_t(t)}[\lambda(t)\ell_{\text{STF}}(\theta, t)]$. We summarize the training process in Algorithm 1. The small batch size $|\mathcal{B}|$ is the same as the normal batch size in the vanilla training process. We defer specific use cases of STF objectives combined with various popular SGMs to Appendix A.

Algorithm 1 Learning the stable target field

Input: Training iteration T, Initial model $\mathbf{s}_\theta$, dataset $\mathcal{D}$, learning rate η.
for $t = 1 \dots T$ **do**
  Sample a large reference batch $\mathcal{B}_L$ from $\mathcal{D}$, and subsample a small batch $\mathcal{B} = \{\mathbf{x}_i\}_{i=1}^{|\mathcal{B}|}$ from $\mathcal{B}_L$
  Uniformly sample the time $\{t_i\}_{i=1}^{|\mathcal{B}|} \sim \mathcal{U}[0,1]^{|\mathcal{B}|}$
  Obtain the batch of perturbed samples $\{\mathbf{x}_i(t_i)\}_{i=1}^{|\mathcal{B}|}$ by applying the transition kernel $p_{t|0}$ on $\mathcal{B}$
  Calculate the stable target field of $\mathcal{B}_L$ for all $\mathbf{x}_i(t_i)$:
  $\mathbf{v}_{\mathcal{B}_L}(\mathbf{x}_i(t_i)) = \sum_{\mathbf{x}\in\mathcal{B}_L} \frac{p_{t_i|0}(\mathbf{x}_i(t_i)|\mathbf{x})}{\sum_{\mathbf{y}\in\mathcal{B}_L} p_{t_i|0}(\mathbf{x}_i(t_i)|\mathbf{y})} \nabla_{\mathbf{x}_i(t_i)} \log p_{t_i|0}(\mathbf{x}_i(t_i)|\mathbf{x})$
  Calculate the loss: $\mathcal{L}(\theta) = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} \lambda(t_i)\|\mathbf{s}_\theta(\mathbf{x}_i(t_i), t_i) - \mathbf{v}_{\mathcal{B}_L}(\mathbf{x}_i(t_i))\|_2^2$
  Update the model parameter: $\theta = \theta - \eta\nabla\mathcal{L}(\theta)$
end for
return $\mathbf{s}_\theta$

5 Analysis

In this section, we analyze the theoretical properties of our approach. In particular, we show that the new minimizer $\mathbf{s}^*_{\text{STF}}(\mathbf{x}(t), t)$ (Equation 7) converges to the true score asymptotically (Section 5.1). Then, we show that the proposed STF reduces the trace-of-covariance of training targets propositional to the reference batch size in the intermediate phase, with mild conditions (Section 5.2).

5.1 Asymptotic behavior

Although in general $\mathbf{s}^*_{\text{STF}}(\mathbf{x}(t), t) \neq \nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t))$, the bias shrinks toward 0 with a increasing n. In the following theorem we show that the minimizer of STF objective at $(\mathbf{x}(t), t)$, *i.e.*, $\mathbf{s}^*_{\text{STF}}(\mathbf{x}(t), t)$, is asymptotically normal when $n \to \infty$.

Theorem 1. *Suppose* $\forall t \in [0,1], 0 < \sigma_t < \infty$, *then*

$$\sqrt{n}\left(\mathbf{s}^*_{STF}(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t))\right) \xrightarrow{d} \mathcal{N}\left(\mathbf{0}, \frac{\text{Cov}(\nabla_{\mathbf{x}(t)} p_{t|0}(\mathbf{x}(t)|\mathbf{x}))}{p_t(\mathbf{x}(t))^2}\right) \tag{8}$$

We defer the proof to Appendix B.2. The theorem states that, for commonly used transition kernels, $\mathbf{s}^*_{\text{STF}}(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t))$ converges to a zero mean normal, and larger reference batch size (n) will lead to smaller asymptotic variance. As can be seen in Equation 8, when $n \to \infty$, $\mathbf{s}^*_{\text{STF}}(\mathbf{x}(t), t)$ highly concentrates around the true score $\nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t))$.

5.2 Trace of Covariance

We now highlight the small variations of the training targets in the STF objective compared to the vanilla one. As done in Section 3, we study the trace-of-covariance of training targets in STF:

$$V_{\text{STF}}(t) = \mathbb{E}_{p_t(\mathbf{x}(t))}\left[\text{Tr}\left(\text{Cov}_{p_{0|t}(\cdot|\mathbf{x}(t))p_0^{n-1}}\left(\sum_{k=1}^{n} \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{\sum_j p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)\right)\right)\right].$$

In the following theorem we compare V_{STF} with V_{v}. In particular, we can upper bound $V_{\text{STF}}(t)$ by

Theorem 2. *Suppose* $\forall t \in [0,1], 0 < \sigma_t < \infty$, *then*

$$V_{STF}(t) \leq \frac{1}{n-1}\left(V_v(t) + \frac{\sqrt{3}d}{\sigma_t^2}\sqrt{\mathbb{E}_{p_t(\mathbf{x}(t))} D_f\left(p_0(\mathbf{x}) \parallel p_{0|t}(\mathbf{x}|\mathbf{x}(t))\right)}\right) + O\left(\frac{1}{n^2}\right),$$

where D_f *is an f-divergence with* $f(y) = \begin{cases}(1/y-1)^2 & (y < 1.5)\\ 8y/27 - 1/3 & (y \geq 1.5)\end{cases}$. *Further, when* $n \gg d$ *and* $p_{0|t}(\mathbf{x}|\mathbf{x}(t)) \approx p_0(\mathbf{x})$ *for all* $\mathbf{x}(t)$, $V_{STF}(t) \lessapprox \frac{V_v(t)}{n-1}$.

We defer the proof to Appendix B.3. The second term that involves f-divergence D_f is necessary to capture how the coefficients, *i.e.*, $p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)/\sum_j p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)$ used to calculate the weighted

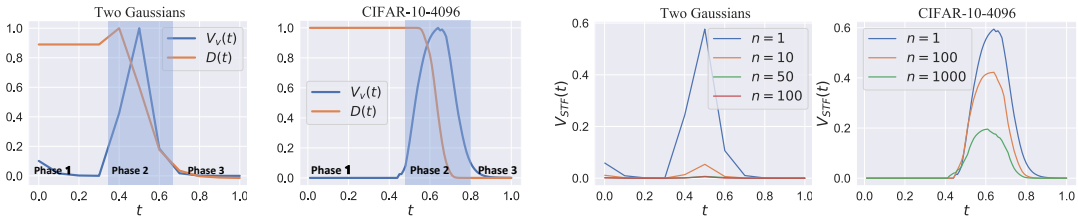
score target, vary across different samples $\mathbf{x}(t)$. This term decreases monotonically as a function of t. In Phase 1, $p_{0|t}(\mathbf{x}|\mathbf{x}(t))$ differs substantially from $p_0(\mathbf{x})$ and the divergence term D_f dominates. In contrast to the upper bound, both $V_{\text{STF}}(t)$ and $V_{\text{v}}(t)$ have minimal variance at small values of t since the training target is always dominated by one $\mathbf{x}$. The theorem has more relevance in Phase 2, where the divergence term decreases to a value comparable to $V_{\text{v}}(t)$. In this phase, we empirically observe that the ratio of the two terms in the upper bound ranges from 10 to 100. Thus, when we use a large reference batch size (in thousands), the theorem implies that STF offers a considerably lower variance (by a factor of 10 or more) relative to the vanilla objective. In Phase 3, the second term vanishes to 0, as $p_t \approx p_{t|0}$ with large σ_t for commonly used transition kernels. As a result, STF reduces the average trace-of-covariance of the training targets by at least $n-1$ times in the far field.

Together, we demonstrate that the STF targets have diminishing bias (Theorem 1) and are much more stable during training (Theorem 2). These properties make the STF objective more favorable for SGMs training with stochastic gradient optimization.

6 Experiments

In this section, we first empirically validate our theoretical analysis in Section 5, especially for variance reduction in the intermediate phase (Section 6.1). Next, we show that the STF objective improves various score-based models on image generation tasks in terms of image quality (Section 6.2). In particular, STF achieves state-of-the-art performance on top of EDM. In addition, we demonstrate that STF accelerates the training of score-based models (Section 6.3), and improves the convergence speed and final performance with an increasing reference batch size (Section 6.3).

6.1 Variance reduction in the intermediate phase



(a) (b) (c) (d)

Figure 3: **(a, b)**: $V_{\text{v}}(t)$ and $D(t)$ versus t. We normalize the maximum values to 1 for illustration purposes. **(c, d)**: $V_{\text{STF}}(t)$ with a varying reference batch size n.

The proposed Algorithm 1 utilizes a large reference batch to calculate the stable target field instead of the individual target. In addition to the theoretical analysis in Section 5, we provide further empirical study to characterize the intermediate phase and verify the variance reduction effects by STF. Apart from $V(t)$, we also quantify the average divergence between the posterior $p_{0|t}(\cdot|\mathbf{x}(t))$ and the data distribution p_0 at time t (introduced in Theorem 2): $D(t) = \mathbb{E}_{p_t(\mathbf{x}(t))}\left[D_f\left(p_{0|t}(\mathbf{x}|\mathbf{x}(t)) \parallel p_0(\mathbf{x})\right)\right]$. Intuitively, the number of high-density modes in $p_{0|t}(\cdot|\mathbf{x}(t))$ grows as $D(t)$ decreases. To investigate their behaviors, we construct two synthetic datasets: (1) a 64-dimensional mixture of two Gaussian components (**Two Gaussians**), and (2) a subset of 4096 images of CIFAR-10 (**CIFAR-10-4096**).

Figure 3(a) and Figure 3(b) show the behaviors of $V_{\text{v}}(t)$ and $D(t)$ on Two Gaussian and CIFAR-10-4096. In both settings, $V_{\text{v}}(t)$ reaches its peak in the intermediate phase (Phase 2), while $D(t)$ gradually decreases over time. These results agree with our theoretical understanding from Section 3. In Phase 2 and 3, several modes of the data distribution have noticeable influences on the scores, but only in Phase 2 are the influences much more distinct, leading to high variations of the individual target $\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}), \mathbf{x} \sim p_{0|t}(\cdot|\mathbf{x}(t))$.

Figure 3(c) and Figure 3(d) further show the relationship between $V_{\text{STF}}(t)$ and the reference batch size n. Recall that when $n = 1$, STF degenerates to individual target and $V_{\text{STF}}(t) = V_{\text{v}}(t)$. We observe that $V_{\text{STF}}(t)$ decreases when enlarging n. In particular, the predicted relation $V_{\text{STF}}(t) \lessapprox V_{\text{v}}(t)/(n-1)$ in Theorem 2 holds for the two Gaussian datasets where D_f is small. On the high dimensional dataset

CIFAR-10-4096, the stable target field can still greatly reduce the training target variance with large reference batch sizes n.

6.2 IMAGE GENERATION

Table 1: CIFAR-10 sample quality (FID, Inception) and number of function evaluation (NFE).

Methods	Inception ↑	FID ↓	NFE ↓
StyleGAN2-ADA (Karras et al., 2020)	9.83	2.92	1
DDPM (Ho et al., 2020)	9.46	3.17	1000
NCSNv2 (Song & Ermon, 2020)	8.40	10.87	1161
PFGM (Xu et al., 2022b)	9.68	2.48	104
VE (Song et al., 2021b)			
vanilla - RK45	9.27	8.90	264
STF (ours) - RK45	9.52 ↑	5.51 ↓	200
vanilla - PC	9.68	2.75	2000
STF (ours) - PC	9.86 ↑	2.66 ↓	2000
VP (Song et al., 2021b)			
vanilla - DDIM	9.20	5.16	100
STF (ours) - DDIM	9.28 ↑	5.06 ↓	100
vanilla - RK45	9.46	2.90	140
STF (ours) - RK45	9.43 ↓	2.99 ↑	140
EDM (Karras et al., 2022)			
vanilla - Heun, NCSN++	9.82	1.98	35
STF (ours) - Heun, NCSN++	**9.93** ↑	**1.90** ↓	35
vanilla - Heun, DDPM++	9.78	1.97	35
STF (ours) - Heun, DDPM++	9.79 ↑	1.92 ↓	35

We demonstrate the effectiveness of the new objective on image generation tasks. We consider CIFAR-10 (Krizhevsky et al., 2009), CelebA 64×64 (Yang et al., 2015) and ImageNet 64×64 (Deng et al., 2009) datasets. We set the reference batch size n to 4096 (CIFAR10) and 1024 (CelebA 64^2, ImageNet 64^2). We choose the current state-of-the-art score-based method EDM (Karras et al., 2022) as the baseline, and replace the vanilla objective with our STF objective during training. We also apply STF to two other popular score-based models, VE/VP SDEs (Song et al., 2021b). For a fair comparison, we directly adopt the architectures and the hyper-parameters in Karras et al. (2018) and Song et al. (2021b) for EDM and VE/VP respectively. In particular, we use the improved NCSN++/DDPM++ models (Karras et al., 2022) in the EDM scheme. To highlight the stability issue, we train three models with different seeds for VE on CIFAR-10. We provide more experimental details in Appendix D.1.

Numerical Solver. The reverse-time ODE and SDE in scored-based models are compatible with any general-purpose solvers. We use the adaptive solver RK45 method (Dormand & Prince, 1980; Song et al., 2021b) (RK45) for VE/VP and the popular DDIM solver (Song et al., 2021a) for VP. We adopt Heun's 2nd order method (Heun) and the time discretization proposed by Karras et al. (2022) for EDM. For SDEs, we apply the predictor-corrector (PC) sampler used in (Song et al., 2021b). We denote the methods in a objective-sampler format, *i.e.*, **A-B**, where $\mathbf{A} \in \{\text{vanilla, STF}\}$ and $\mathbf{B} \in \{\text{RK45, PC, DDIM, Heun}\}$. We defer more details to Appendix D.2.

Results. For quantitative evaluation of the generated samples, we report the FID scores (Heusel et al., 2017) (lower is better) and Inception (Salimans et al., 2016) (higher is better). We measure the sampling speed by the average NFE (number of function evaluations). We also include the results of several popular generative models (Karras et al., 2020; Ho et al., 2020; Song & Ermon, 2019; Xu et al., 2022b) for reference.

Table 1 and Table 2 report the sample quality and the sampling speed on unconditional generation of CIFAR-10, CelebA 64^2 and ImageNet 64^2. Our main findings are: **(1) STF achieves new state-of-the-art FID scores for unconditional generation on CIFAR-10 benchmark.** As shown in Table 1, The STF objective obtains a FID of 1.90 when incorporated with the EDM scheme.

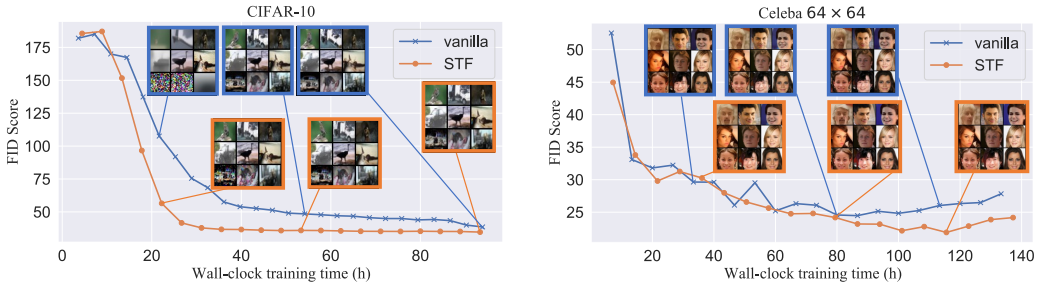
To the best of our knowledge, this is the lowest FID score on the unconditional CIFAR-10 generation task. In addition, the STF objective consistently improves the EDM across the two architectures. **(2) The STF objective improves the performance of different score-based models.** We observe that the STF objective improves the FID/Inception scores of VE/VP/EDM on CIFAR-10, for most ODE and SDE samplers. STF consistently provides performance gains for VE across datasets. Remarkably, our objective achieves much better sample quality using ODE samplers for VE, with an FID score gain of 3.39 on CIFAR-10, and 2.22 on Celeba 64^2. For VP, STF provides better results on the popular DDIM sampler, while suffering from a slight performance drop when using the RK45 sampler. **(3) The STF objective stabilizes the converged VE model with the RK45 sampler.** In Appendix E.1, we report the standard deviations of performance metrics for converged models with different seeds on CIFAR-10 with VE. We observe that models trained with the STF objective give more consistent results, with a smaller standard deviation of used metrics.

Table 2: FID and NFE on CelebA 64^2 and ImageNet 64^2

Methods/NFEs	FID ↓	NFE ↓
CelebA 64^2 **- *RK45***		
VE (vanilla)	7.56	260
VE (STF)	**5.34**	266
CelebA 64^2 **- *PC***		
VE (vanilla)	9.13	2000
VE (STF)	**8.28**	2000
ImageNet 64^2 **- *PC***		
VE (vanilla)	25.96	2000
VE (STF)	**24.33**	2000

We further provide generated samples in Appendix F. One interesting observation is that when using the RK45 sampler for VE on CIFAR-10, the generated samples from the STF objective do not contain noisy images, unlike the vanilla objective on CIFAR-10.

6.3 Accelerating training of SGMs



(a) CIFAR-10 (b) CelebA 64×64

Figure 4: FID and generated samples throughout training on (a) CIFAR-10 and (b) CelebA 64^2.

The variance-reduction techniques in neural network training can help to find better optima and achieve faster convergence rate (Wang et al., 2013; Defazio et al., 2014; Johnson & Zhang, 2013). In Figure 4, we demonstrate the FID scores every 50k iterations during the course of training. Since our goal is to investigate relative performance during the training process, and because the FID scores computed on 1k samples are strongly correlated with the full FID scores on 50k sample (Song & Ermon, 2020), we report FID scores on 1k samples for faster evaluations. We apply ODE samplers for FID evaluation, and measure the training time on two NVIDIA A100 GPUs. For a fair comparison, we report the average FID scores of models trained by the vanilla and STF objective on VE versus the wall-clock training time (h).

The STF objective achieves better FID scores with the same training time, although the calculation of the target field by a large reference batch introduces slight overhead (Algorithm 1). In Figure 4(a), we show that the STF objective drastically accelerates the training of score-based models on CIFAR-10. The STF objective achieves comparable FID scores with $3.6\times$ less training time (25h versus 90h). For CelebA 64^2 datasets, the training time improvement is less significant than on CIFAR-10. Our hypothesis is that the STF objective is more effective when there are multiple well-separated modes in data distribution, *e.g.*, the ten classes in CIFAR-10, making the vanilla objective suffer from relatively larger variations in the intermediate

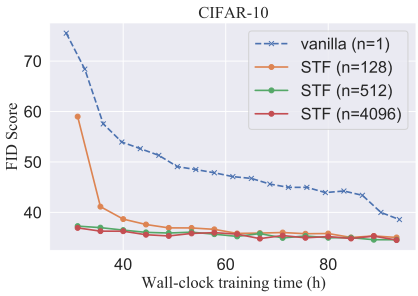


Figure 5: FID scores in the training course with varying reference batch size.

phase. In addition, we observe that the converged models have better final performance when pairing with the STF objective on both datasets.

6.4 EFFECTS OF THE REFERENCE BATCH SIZE

According to our theory (Theorem 2), the upper bound of the trace-of-covariance of the STF target decreases proportionally to the reference batch size. Here we study the effects of the reference batch size (n) on model performances during training. The FID scores are evaluated on $1k$ samples using the RK45 sampler. As shown in Figure 5, models converge faster and produce better samples when increasing n. It suggests that smaller variations of the training targets can indeed speed up training and improve the final performances of score-based models.

7 RELATED WORK

Different phases of SGMs. The idea of SGMs having different phases has been explored in prior works though the motivations and definitions vary (Karras et al., 2022; Choi et al., 2022). Karras et al. (2022) argues that the training targets are difficult and unnecessary to learn in the very near field (small t in our Phase 1), whereas the training targets are always dissimilar to the true targets in the intermediate and far field (our Phase 2 and Phase 3). As a result, their solution is sampling t with a log-normal distribution to emphasize the relevant region (relatively large t in our Phase 1). In contrast, we focus on reducing large training target variance in the intermediate and far field, and propose STF to better estimate the true target (cf. Karras et al. (2022)). Choi et al. (2022) identifies a key region where the model learns perceptually rich contents, and determines the training weights $\lambda(t)$ based on the signal-to-noise ratio (SNR) at different t. As SNR is monotonically decreasing over time, the resulting up-weighted region does not match our Phase 2 characterization. In general, our proposed STF method reduces the training target variance in the intermediate field and is complementary to previous improvements of SGMs.

Importance sampling. The technique of importance sampling has been widely adopted in machine learning community, such as debiasing generative models (Grover et al., 2019), counterfactual learning (Swaminathan & Joachims, 2015) and reinforcement learning (Metelli et al., 2018). Prior works using importance sampling to improve generative model training include reweighted wake-sleep (RWS) (Bornschein & Bengio, 2014) and importance weighted autoencoders (IWAE) (Burda et al., 2015). RWS views the original wake-sleep algorithm (Hinton et al., 1995) as importance sampling with one latent variable, and proposes to sample multiple latents to obtain gradient estimates with lower bias and variance. IWAE utilizes importance sampling with multiple latents to achieve greater flexibility of encoder training and tighter log-likelihood lower bound compared to the standard variational autoencoder (Kingma & Welling, 2013; Rezende et al., 2014).

Variance reduction for Fisher divergence. One popular approach to score-matching is to minimize the Fisher divergence between true and predicted scores (Hyvärinen & Dayan, 2005). Wang et al. (2020) links the Fisher divergence to denoising score-matching (Vincent, 2011) and studies the large variance problem (in $O(1/\sigma_t^4)$) of the Fisher divergence when $t \to 0$. They utilize a control variate to reduce the variance. However, this is typically not a concern for current SGMs as the time-dependent objective can be viewed as multiplying the Fisher divergence by $\lambda(t) = \sigma_t^2$, resulting in a finite-variance objective even when $t \to 0$.

8 CONCLUSION

We identify large target variance as a significant training issue affecting SGMs. We define three phases with distinct behaviors, and show that the high-variance targets appear in the intermediate phase. As a remedy, we present a generalized score-matching objective, *Stable Target Field* (STF), whose formulation is analogous to the self-normalized importance sampling via a large reference batch. Albeit no longer an unbiased estimator, our proposed objective is asymptotically unbiased and reduces the trace-of-covariance of the training targets, which we demonstrate theoretically and empirically. We show the effectiveness of our method on image generation tasks, and show that STF improves the performance, stability, and training speed over various state-of-the-art SGMs. Future directions include a principled study on the effect of different reference batch sampling procedures. Our presented approach is uniformly sampling from the whole dataset $\{\mathbf{x}_i\}_{i=2}^n \sim p_0^{n-1}$, so we expect that training SGMs with a reference batch of more samples in the neighborhood of $\mathbf{x}_1$ (the sample from which $\mathbf{x}(t)$ is perturbed) would lead to an even better estimation of the score field.

REFERENCES

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

Jörg Bornschein and Yoshua Bengio. Reweighted wake-sleep. *arXiv preprint arXiv:1406.2751*, 2014.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11472–11481, 2022.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

J. R. Dormand and P. J. Prince. A family of embedded runge-kutta formulae. *Journal of Computational and Applied Mathematics*, 6:19–26, 1980.

Víctor Elvira and Luca Martino. Advances in importance sampling. *Wiley StatsRef: Statistics Reference Online*, 2021.

Aditya Grover, Jiaming Song, Ashish Kapoor, Kenneth Tran, Alekh Agarwal, Eric J Horvitz, and Stefano Ermon. Bias correction of learned generative models using likelihood-free importance weighting. *Advances in neural information processing systems*, 32, 2019.

Tim Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37:185–194, 1995.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

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

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ArXiv*, abs/1710.10196, 2018.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *ArXiv*, abs/2006.06676, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2836–2844, 2021.

Alberto Maria Metelli, Matteo Papini, Francesco Faccio, and Marcello Restelli. Policy optimization via importance sampling. In *NeurIPS*, 2018.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.

Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

Severi Rissanen, Markus Heinonen, and A. Solin. Generative modelling with inverse heat dissipation. *ArXiv*, abs/2206.13397, 2022.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *ArXiv*, abs/1606.03498, 2016.

Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In *ICML*, 2021.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ArXiv*, abs/2010.02502, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *ArXiv*, abs/2006.09011, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. In *NIPS*, 2015.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23:1661–1674, 2011.

Chong Wang, X. Chen, Alex Smola, and E. Xing. Variance reduction for stochastic gradient optimization. In *NIPS*, 2013.

Ziyu Wang, Shuyu Cheng, Li Yueru, Jun Zhu, and Bo Zhang. A wasserstein minimum velocity approach to learning unnormalized models. In *International Conference on Artificial Intelligence and Statistics*, pp. 3728–3738. PMLR, 2020.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=JprM0p-q0Co.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: a geometric diffusion model for molecular conformation generation. *ArXiv*, abs/2203.02923, 2022a.

Yilun Xu, Ziming Liu, Max Tegmark, and Tommi Jaakkola. Poisson flow generative models. *arXiv preprint arXiv:2209.11178*, 2022b.

Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. From facial parts responses to face detection: A deep learning approach. *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3676–3684, 2015.

Appendix

A STF SPECIFIED WITH POPULAR SGMS

Here, we detail the practically used STF objectives in Section 6, which are built on the popular instances of SGMs, *e.g.*, VE, VP (Song et al., 2021b), and EDM (Karras et al., 2022).

VE and EDM For VE and EDM, the transition kernel is in the form of

$$p_{t|0}(\mathbf{x}(t)|\mathbf{x}) = \mathcal{N}\left(\mathbf{x},\ \sigma_t^2 \boldsymbol{I}\right) \quad t \in [0, 1].$$

VE has $\sigma_t = \sigma_m \left(\frac{\sigma_M}{\sigma_m}\right)^t$ for some fixed σ_m and σ_M. EDM has $\sigma_t = (t\sigma_M^{\frac{1}{\rho}} + (1-t)\sigma_m^{\frac{1}{\rho}})^\rho$ for some σ_m and σ_M, with ρ set to 7 in practice. The STF objective for both VE and EDM at $\mathbf{x}(t)$ is then in the following form:

$$\mathbb{E}_{\mathbf{x}_1 \sim p_{0|t}(\cdot|\mathbf{x}(t))} \mathbb{E}_{\{\mathbf{x}_i\}_{i=2}^n \sim p_0^{n-1}} \left[\left\| \mathbf{s}_\theta(\mathbf{x}(t), t) - \frac{1}{\sigma_t^2} \sum_{k=1}^n \frac{\exp\left(-\frac{\|\mathbf{x}(t)-\mathbf{x}_k\|_2^2}{2\sigma_t^2}\right)}{\sum_j \exp\left(-\frac{\|\mathbf{x}(t)-\mathbf{x}_j\|_2^2}{2\sigma_t^2}\right)} (\mathbf{x}_k - \mathbf{x}(t)) \right\|_2^2 \right].$$

VP VP in its original formulation has the transition kernel as

$$p_{t|0}(\mathbf{x}(t)|\mathbf{x}) = \mathcal{N}\left(e^{-\frac{1}{4}t^2(\beta_M-\beta_m)-\frac{1}{2}t\beta_m}\mathbf{x},\ \boldsymbol{I} - \boldsymbol{I}e^{-\frac{1}{2}t^2(\beta_M-\beta_m)-t\beta_m}\right),$$

for some β_m and β_M. The STF objective for VP at $\mathbf{x}(t)$ is

$$\mathbb{E}_{\mathbf{x}_1 \sim p_{0|t}} \mathbb{E}_{\{\mathbf{x}_i\}_{i=2}^n \sim p_0^{n-1}} \left[\left\| \mathbf{s}_\theta(\mathbf{x}(t), t) - \frac{1}{\sigma_t^2} \sum_{k=1}^n \frac{\exp\left(-\frac{\|\mathbf{x}(t)-e^{\beta_t}\mathbf{x}_k\|_2^2}{2\sigma_t^2}\right)}{\sum_j \exp\left(-\frac{\|\mathbf{x}(t)-e^{\beta_t}\mathbf{x}_j\|_2^2}{2\sigma_t^2}\right)} \left(e^{\beta_t}\mathbf{x}_k - \mathbf{x}(t)\right) \right\|_2^2 \right],$$

where $\beta_t = -\frac{1}{4}t^2(\beta_M-\beta_m)-\frac{1}{2}t\beta_m$, and $\sigma_t = \sqrt{1-e^{2\beta_t}}$. Note that as shown in Karras et al. (2022), VP's transition kernel can be reparameterized in the form of $\mathcal{N}(\mathbf{x}, \sigma_t^2\boldsymbol{I})$ with a correspondingly revised sampling process. Adopting this formulation, we would have the STF objective for VP the same as the one for VE and EDM with a different σ_t.

B PROOFS

B.1 DERIVATION OF EQUATION 7

Recall that the STF objective (Equation 6) at time t is

$$\ell_{\text{STF}}(\theta, t) = \mathbb{E}_{\{\mathbf{x}_i\}_{i=1}^n \sim p_0^n} \mathbb{E}_{\mathbf{x}(t) \sim p_{t|0}(\cdot|\mathbf{x}_1)} \left[\left\| \mathbf{s}_\theta(\mathbf{x}(t), t) - \sum_{k=1}^n \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{\sum_{j=1}^n p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) \right\|_2^2 \right].$$

Swapping the sampling order and we get

$$\ell_{\text{STF}}(\theta, t) = \mathbb{E}_{\mathbf{x}(t) \sim p_t} \mathbb{E}_{\{\mathbf{x}_i\}_{i=1}^n \sim p_{0|t}(\cdot|\mathbf{x}(t)) p_0^{n-1}} \left[\left\| \mathbf{s}_\theta(\mathbf{x}(t), t) - \sum_{k=1}^n \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{\sum_{j=1}^n p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) \right\|_2^2 \right].$$

This means that with input $\mathbf{x}(t)$ and t, the model is optimized with

$$\mathbb{E}_{\{\mathbf{x}_i\}_{i=1}^n \sim p_{0|t}(\cdot|\mathbf{x}(t)) p_0^{n-1}} \left[\left\| \mathbf{s}_\theta(\mathbf{x}(t), t) - \sum_{k=1}^n \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{\sum_{j=1}^n p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) \right\|_2^2 \right].$$

Taking its derivative w.r.t. $\mathbf{s}_\theta(\mathbf{x}(t), t)$ results in

$$\mathbb{E}_{\{\mathbf{x}_i\}_{i=1}^n \sim p_{0|t}(\cdot|\mathbf{x}(t))p_0^{n-1}} \left[2 \left(\mathbf{s}_\theta(\mathbf{x}(t), t) - \sum_{k=1}^n \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{\sum_{j=1}^n p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) \right) \right].$$

Setting it to 0, we have

$$\mathbf{s}_\theta(\mathbf{x}(t), t) - \mathbb{E}_{\{\mathbf{x}_i\}_{i=1}^n \sim p_{0|t}(\cdot|\mathbf{x}(t))p_0^{n-1}} \left[\sum_{k=1}^n \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{\sum_{j=1}^n p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) \right] = 0.$$

Thus, we arrive at the minimizer of the STF objective (Equation 7):

$$\mathbf{s}^*_{\text{STF}}(\mathbf{x}(t), t) = \mathbb{E}_{\mathbf{x}_1 \sim p_{0|t}(\cdot|\mathbf{x}(t))} \mathbb{E}_{\{\mathbf{x}_i\}_{i=2}^n \sim p_0^{n-1}} \left[\sum_{k=1}^n \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{\sum_j p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) \right].$$

B.2 Proof for Theorem 1

Theorem 1. *Suppose* $\forall t \in [0, 1], 0 < \sigma_t < \infty$, *then*

$$\sqrt{n} \left(\mathbf{s}^*_{STF}(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t)) \right) \xrightarrow{d} \mathcal{N} \left(\mathbf{0}, \frac{\text{Cov}(\nabla_{\mathbf{x}(t)} p_{t|0}(\mathbf{x}(t)|\mathbf{x}))}{p_t(\mathbf{x}(t))^2} \right) \tag{8}$$

Proof. Recall that $\mathbf{s}^*_{\text{STF}}(\mathbf{x}(t), t)$ is calculated via Equation 7. The mean of the denominator in the expectation $\frac{1}{n} \sum_{j=1}^n p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)$, for large n, approximate $\frac{1}{n-1} \sum_{j=2}^n p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)$, which in turn, $\frac{1}{n-1} \sum_{j=2}^n p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j) \xrightarrow{p} p_t(\mathbf{x}(t))$ by WLLN.

Similarly, for the mean of the remaining terms in the expectation, by CLT, we have

$$\sqrt{n} \left(\frac{1}{n} \sum_{k=1}^n p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) - p_t(\mathbf{x}(t)) \nabla_{\mathbf{x}(t)} \log p_t(\mathbf{x}(t)) \right)$$
$$\xrightarrow{d} \mathcal{N} \left(\mathbf{0}, \ \text{Cov}(\nabla_{\mathbf{x}(t)} p_{t|0}(\mathbf{x}(t)|\mathbf{x})) \right)$$

Putting them together via Slutsky's theorem, we conclude the proof. □

B.3 Proof for Theorem 2

Theorem 2. *Suppose* $\forall t \in [0, 1], 0 < \sigma_t < \infty$, *then*

$$V_{STF}(t) \leq \frac{1}{n-1} \left(V_v(t) + \frac{\sqrt{3}d}{\sigma_t^2} \sqrt{\mathbb{E}_{p_t(\mathbf{x}(t))} D_f \left(p_0(\mathbf{x}) \parallel p_{0|t}(\mathbf{x}|\mathbf{x}(t)) \right)} \right) + O\left(\frac{1}{n^2} \right),$$

where D_f *is an f-divergence with* $f(y) = \begin{cases} (1/y - 1)^2 & (y < 1.5) \\ 8y/27 - 1/3 & (y \geq 1.5) \end{cases}$. *Further, when* $n \gg d$ *and* $p_{0|t}(\mathbf{x}|\mathbf{x}(t)) \approx p_0(\mathbf{x})$ *for all* $\mathbf{x}(t)$, $V_{STF}(t) \lessapprox \frac{V_v(t)}{n-1}$.

Proof. **Step 1: Make the likelihood weighting coefficients "independent"**

We first apply Hoeffding's inequality for the set $\{\mathbf{x}_i\}_{i=2}^n \sim p_0^{n-1}$ to make the summation $\sum_{j=2}^n p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)$ concentrate to its expectation $(n-1)p_t(\mathbf{x}(t))$. Since $p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j) \in (0, \frac{1}{(\sqrt{2\pi}\sigma_t)^d})$, we have

$$\Pr \left[\left| \sum_{j=2}^n p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j) - (n-1)p_t(\mathbf{x}(t)) \right| \geq n^{\gamma_1} \right] \leq 2e^{\frac{-2n^{2\gamma_1}(\sqrt{2\pi}\sigma_t)^d}{(n-1)}},$$

$\forall \gamma_1 \in (\frac{1}{2}, 1)$.

Thus the summation can be re-expressed as:

$$\begin{aligned}\sum_{j=2}^{n} p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j) &= (1 - O(2e^{\frac{-2n^{2\gamma_1}(\sqrt{2\pi}\sigma_t)^d}{(n-1)}}))[(n-1)p_t(\mathbf{x}(t)) + O(n^{\gamma_1})] \\ &\quad + O(2e^{\frac{-2n^{2\gamma_1}(\sqrt{2\pi}\sigma_t)^d}{(n-1)}})O((n-1)\frac{1}{(\sqrt{2\pi}\sigma_t)^d}) \\ &= (n-1)p_t(\mathbf{x}(t)) + O(ne^{-2n^{2\gamma_1-1}(\sqrt{2\pi}\sigma_t)^d})\end{aligned}$$

The coefficient for $\mathbf{x}_1$ is then

$$\begin{aligned}\frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_1)}{\sum_{j=1}^{n} p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} &= \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_1)}{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_1) + \sum_{j=2}^{n} p_{t|0}(\mathbf{x}(t)|x_j)} \\ &= \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_1)}{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_1) + (n-1)p_t(\mathbf{x}(t)) + O(ne^{-2n^{2\gamma_1-1}(\sqrt{2\pi}\sigma_t)^d})} \\ &= O(\frac{1}{n})\end{aligned}$$

The coefficient for $x_k, k \in \{2, \dots, n\}$ is:

$$\begin{aligned}\frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{\sum_{j=1}^{n} p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} &= \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{(n-1)p_t(\mathbf{x}(t)) + p_{t|0}(\mathbf{x}(t)|\mathbf{x}_1) + O(ne^{-2n^{2\gamma_1-1}(\sqrt{2\pi}\sigma_t)^d})} \\ &= \frac{1}{(n-1)}\frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{p_t(\mathbf{x}(t))} + O(\frac{1}{n^2}) + O(ne^{-2n^{2\gamma_1-1}(\sqrt{2\pi}\sigma_t)^d})\end{aligned}$$

Step 2: Re-express the trace-of-covariance by the "independent" weights

Plugging in the above formulation of coefficients, we can rewrite the trace-of-covariance for the new target as:

$$\begin{aligned}&V_{\text{STF}}(\mathbf{x}(t), t) \\ &= \mathbb{E}_{\mathbf{x}_1 \sim p_{0|t}(\mathbf{x}|\mathbf{x}(t))}\mathbb{E}_{\{\mathbf{x}_i\}_{i=2}^n \sim p^{n-1}(\mathbf{x})}\|\sum_{k=1}^{n} \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{\sum_{j=1}^{n} p_{t|0}(\mathbf{x}(t)|x_j)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) \\ &\qquad - \mathbb{E}_{\mathbf{x}_1 \sim p_{0|t}(\mathbf{x}|\mathbf{x}(t))}\mathbb{E}_{\{\mathbf{x}_i\}_{i=2}^n \sim p^{n-1}(\mathbf{x})}\sum_{k=1}^{n} \frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{\sum_{j=1}^{n} p_{t|0}(\mathbf{x}(t)|\mathbf{x}_j)} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)\|_2^2 \\ &\leq \mathbb{E}_{\mathbf{x}_1 \sim p_{0|t}(\mathbf{x}|\mathbf{x}(t))}\mathbb{E}_{\{\mathbf{x}_i\}_{i=2}^n \sim p^{n-1}(\mathbf{x})}\|\sum_{k=2}^{n} \frac{1}{(n-1)}\frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{p_t(\mathbf{x}(t))} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) \\ &\qquad - \mathbb{E}_{\mathbf{x}_1 \sim p_{0|t}(\mathbf{x}|\mathbf{x}(t))}\mathbb{E}_{\{\mathbf{x}_i\}_{i=2}^n \sim p^{n-1}(\mathbf{x})}\sum_{k=2}^{n} \frac{1}{(n-1)}\frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{p_t(\mathbf{x}(t))} \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) \\ &\qquad + \sum_{k=2}^{n} O(\frac{1}{n^2})\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k) - O(\frac{1}{n})\mathbb{E}_{p(\mathbf{x})}[\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x})]\|_2^2 + O(\frac{1}{n^2}) \\ &= \frac{1}{(n-1)^2}\sum_{k=2}^{n} \operatorname{Tr}\left(\operatorname{Cov}_{\mathbf{x}_k \sim p(\mathbf{x})}(\frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k)}{p_t(\mathbf{x}(t))}\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}_k))\right) + O(\frac{1}{n^2}) \\ &= \frac{1}{(n-1)}\operatorname{Tr}\left(\operatorname{Cov}_{\mathbf{x} \sim p(\mathbf{x})}(\frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x})}{p_t(\mathbf{x}(t))}\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}))\right) + O(\frac{1}{n^2}) \qquad (9)\end{aligned}$$

Step 3: Upper bound the new trace-of-covariance term

Next, we examine the new trace-of-covariance term:

$$
\begin{aligned}
&\operatorname{Tr}\left(\operatorname{Cov}_{\mathbf{x}\sim p_0(\mathbf{x})}(\frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x})}{p_t(\mathbf{x}(t))}\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x}))\right)\\
=&\sum_{i=1}^{d}\mathbb{E}_{p_0(\mathbf{x})}\left[(\frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x})}{p_t(\mathbf{x}(t))}\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x}))_i)^2\right]\\
&\qquad-\left(\mathbb{E}_{p_0(\mathbf{x})}\left[\frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x})}{p_t(\mathbf{x}(t))}\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x}))_i)\right]\right)^2\\
=&\sum_{i=1}^{d}\mathbb{E}_{p_0(\mathbf{x})}\left[(\frac{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}{p_0(\mathbf{x})}\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x}))_i)^2\right]\\
&\qquad-\left(\mathbb{E}_{p_0(\mathbf{x})}\left[\frac{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}{p_0(\mathbf{x}(t))}\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x}))_i)\right]\right)^2\\
=&\sum_{i=1}^{d}\mathbb{E}_{p_0(\mathbf{x})}\left[(\frac{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}{p_0(\mathbf{x})}\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x}))_i)^2\right]-(\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t))_i)^2\\
=&\sum_{i=1}^{d}\mathbb{E}_{p_0(\mathbf{x})}\left[(\frac{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}{p_0(\mathbf{x})}\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x}))_i)^2\right]-(\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t))_i)^2\\
&\qquad+\mathbb{E}_{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}\left[(\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x})_i)^2\right]-\mathbb{E}_{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}\left[(\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x})_i)^2\right]\\
=&V_{\mathrm{v}}(\mathbf{x}(t),t)+\sum_{i=1}^{d}\mathbb{E}_{p_{0|t}}\left[(\frac{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}{p(\mathbf{x})}-1)\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x})_i^2\right]\\
\leq&V_{\mathrm{v}}(\mathbf{x}(t),t)+\sum_{i=1}^{d}\sqrt{\mathbb{E}_{p_{0|t}}\left[(\frac{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}{p(\mathbf{x})}-1)^2\right]\mathbb{E}_{p_{0|t}}\left[\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x})_i^4\right]}
\end{aligned}
\tag{10}
$$

We can further upper bound the trace-of-covariance term in Equation 9:

$$
\begin{aligned}
&V_{\mathrm{STF}}(\mathbf{x}(t),t)\\
=&\frac{1}{(n-1)}\operatorname{Tr}(\operatorname{Cov}_{\mathbf{x}\sim p(\mathbf{x})}(\frac{p_{t|0}(\mathbf{x}(t)|\mathbf{x})}{p_t(\mathbf{x}(t))}\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x})))+O(\frac{1}{n^2})\\
\leq&\frac{1}{n-1}\left(V_{\mathrm{v}}(\mathbf{x}(t),t)+\sum_{i=1}^{d}\sqrt{\mathbb{E}_{p_{0|t}}\left[(\frac{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}{p(\mathbf{x})}-1)^2\right]\mathbb{E}_{p_{0|t}}\left[\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x})_i^4\right]}\right)\\
&\qquad+O(\frac{1}{n^2})
\end{aligned}
$$

Taking the expectation w.r.t $p_t(\mathbf{x}(t))$ for both sides, we get

$$
\begin{aligned}
& V_{\text{STF}}(t) \\
\leq\ & \mathbb{E}_{p_t(\mathbf{x}(t))}\Bigg[\frac{1}{n-1}\Bigg(V_{\text{v}}(\mathbf{x}(t),t) \\
& \quad + \sum_{i=1}^{d}\sqrt{\mathbb{E}_{p_{0|t}}\left[(\frac{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}{p(\mathbf{x})}-1)^2\right]\mathbb{E}_{p_{0|t}}\left[\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x})_i^4\right]}\Bigg)+O(\frac{1}{n^2})\Bigg] \\
\leq\ & \frac{1}{n-1}\left(V_{\text{v}}(t)+\sum_{i=1}^{d}\mathbb{E}_{p_t(\mathbf{x}(t))}\sqrt{\mathbb{E}_{p_{0|t}}\left[(\frac{p_{0|t}(\mathbf{x}|\mathbf{x}(t))}{p(\mathbf{x})}-1)^2\right]\mathbb{E}_{p_{0|t}}\left[\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x})_i^4\right]}\right) \\
& \quad + O(\frac{1}{n^2}) \\
\leq\ & \frac{1}{n-1}\left(V_{\text{v}}(t)+\sum_{i=1}^{d}\sqrt{\mathbb{E}_{p_t(\mathbf{x}(t))}D_f\left(p_0(\mathbf{x})\parallel p_{0|t}(\mathbf{x}|\mathbf{x}(t))\right)}\sqrt{\mathbb{E}_{p_{0,t}}\left[\nabla_{\mathbf{x}(t)}\log p_{t|0}(\mathbf{x}(t)|\mathbf{x})_i^4\right]}\right) \\
& \quad + O(\frac{1}{n^2}) \qquad\qquad \text{(Concavity of } x^{\frac{1}{2}}\text{, Cauchy's inequality)} \\
\leq\ & \frac{1}{n-1}\left(V_{\text{v}}(t)+d\sqrt{\mathbb{E}_{z\sim\mathcal{N}(0,\sigma^2)}[\frac{z^4}{\sigma_t^8}]}\sqrt{\mathbb{E}_{p_t(\mathbf{x}(t))}D_f\left(p_0(\mathbf{x})\parallel p_{0|t}(\mathbf{x}|\mathbf{x}(t))\right)}\right)+O(\frac{1}{n^2}) \\
\leq\ & \frac{1}{n-1}\left(V_{\text{v}}(t)+\frac{\sqrt{3}d}{\sigma_t^2}\sqrt{\mathbb{E}_{p_t(\mathbf{x}(t))}D_f\left(p_0(\mathbf{x})\parallel p_{0|t}(\mathbf{x}|\mathbf{x}(t))\right)}\right)+O(\frac{1}{n^2})
\end{aligned}
$$

where D_f is an f-divergence with $f(y)=\begin{cases}(1/y-1)^2 & (y<1.5)\\ 8y/27-1/3 & (y\geq 1.5)\end{cases}$. Note that we choose this particular form of $f(y)$ since it is the convex function with the tightest upper bound on $(\frac{1}{y}-1)^2$. □

C Details for the behavior of $V_{\text{v}}(t)$

In Section 3, we demonstrate the behavior of $V_{\text{v}}(t)$ in the three phases on Two Gaussians and a subset of CIFAR-10. Here we provide more details about the two datasets.

The distribution of the two Gaussian is $\frac{1}{2}\mathcal{N}(\boldsymbol{\mu},\hat{\sigma}^2\boldsymbol{I}_{64\times 64})+\frac{1}{2}\mathcal{N}(-\boldsymbol{\mu},\hat{\sigma}^2\boldsymbol{I}_{64\times 64})$, where $\boldsymbol{\mu}=0.1\cdot\mathbf{1}\in\mathbb{R}^{\mathbf{64}}$, and $\hat{\sigma}=1e-4$. We estimate all the integrals in Equation 4 by sampling 1k points from the corresponding distributions. For the subset of CIFAR-10, we uniformly sample 4096 images from CIFAR-10 dataset, and assign uniform distribution on the discrete set. We also approximate $V_{\text{v}}(t)$ by Monte Carlo estimation and sample 200 perturbations for each t. We use VE SDE for all simulations, and set $\sigma_m=1e-2,\sigma_M=50$.

Interestingly, $V_{\text{v}}(t)$ is relatively large for the Two Gaussians distribution compared to CIFAR-10 (see Figure 2(b)) when $t\to 0$. This can be explained by their continuous and discrete natures. For the two Gaussian distribution, we can rewrite $V_{\text{v}}(t)$ as

$$
V_{\text{v}}(t)=\mathbb{E}_{\mathbf{x}(t)\sim p_t(\mathbf{x}(t))}\left[\frac{\hat{\sigma}^2}{\sigma_t^2(\sigma_t^2+\hat{\sigma}^2)}+4\frac{\alpha(\mathbf{x}(t))(1-\alpha(\mathbf{x}(t)))\|\boldsymbol{\mu}\|\sigma_t^4}{\sigma_t^4+\hat{\sigma}^2}\right]
$$

where $\alpha(\mathbf{x})=\frac{1}{1+\exp^{-4\mathbf{x}^T\boldsymbol{\mu}}}$ can be regarded as the probability that $\mathbf{x}$ comes from the Gaussian component $\mathcal{N}(\boldsymbol{\mu},\hat{\sigma}^2\boldsymbol{I}_{64\times 64})$. When $t\to 0$, obviously $\alpha(\mathbf{x}(t))(1-\alpha(\mathbf{x}(t)))\to 0$, and the term $\frac{\alpha(\mathbf{x}(t))(1-\alpha(\mathbf{x}(t)))\|\boldsymbol{\mu}\|\sigma_t^4}{\sigma_t^4+\hat{\sigma}^2}$ vanishes. Hence $\lim_{t\to 0}V_{\text{v}}(t)\approx\lim_{t\to 0}\frac{\hat{\sigma}^2}{\sigma_t^2(\sigma_t^2+\hat{\sigma}^2)}=\frac{\hat{\sigma}^2}{\sigma_m^2(\sigma_m^2+\hat{\sigma}^2)}$, which can not be neglected when σ_m is small. On the other hand, we can effectively view the 4069 discrete samples as a mixture of 4096 0-variance Gaussians, *i.e.*, $\hat{\sigma}=0$. Thus by similar reasoning we could see $\lim_{t\to 0}V_{\text{v}}(t)\approx\lim_{t\to 0}\frac{\hat{\sigma}^2}{\sigma_t^2(\sigma_t^2+\hat{\sigma}^2)}=0$.

D EXPERIMENTAL DETAILS

In this section, we include more details about the training and sampling of score-based models by the STF and vanilla objectives. All the experiments are run on two NVIDIA A100 GPUs.

D.1 TRAINING

We consider the CIFAR-10, CelebA 64^2, and ImageNet 64^2 in image generation tasks. Following Song & Ermon (2020), we first center-crop the CelebA images and then resize them to 64×64. For VE/VP, we use the same set of hyper-parameters and the NCSN++/DDPM++ backbones and the continuous-time training objectives for forward SDEs in Song et al. (2021b). For EDM, we adopt the improved hyper-parameters and architectures for NCSN++ in Karras et al. (2022). We set the reference batch size n to 4096 (VE,VP), 1024 (EDM) on CIFAR-10, 1024 on CelebA 64^2 and 1024 on ImageNet 64^2. The training iteration is 1.3M on CIFAR-10 and 1M on CelebA 64^2 for VE/VP, and 200M images for EDM (Karras et al., 2022). The small batch size $\mathcal{B}$ in Algorithm 1 is the same as the batch size in the baseline score-based methods. For model selection, we pick the checkpoint with the lowest FID per 50k iterations on 10k samples for computing all the scores, as in Song et al. (2021b) for VE/VP, and per 2.5M images on 50k samples as in Karras et al. (2022) for EDM.

To measure the stability of converged VE models, we repeat the experiment 3 times on CIFAR-10 for vanilla and STF objectives, using different random seeds.

We quantitatively study the training overhead of STF. All the numbers are measured on two NVIDIA A100 GPUs. In Table 3 and Table 4, we report the wall-clock training time (s) per 50 iterations/50k images on VE/EDM. We can see that the STF introduces additional overhead after incorporating the large reference batch. Since the calculation of the mini-batch target does not involve neural networks, the STF does not take significantly longer training time. Indeed, in Section 6.3 we show that STF achieves comparable or better performance within a shorter training time.

Table 3: Wall-clock training time (s) per 50 iterations on VE with NCSN++ (Song et al., 2021b)

Dataset-Method	CIFAR-10 - vanilla	CIFAR-10 - STF	CelebA - vanilla	CelebA - STF
Wall-clock time	13	16	24	26

Table 4: Wall-clock training time (s) per 50k images on EDM with improved NCSN++ (Karras et al., 2022)

Dataset-Method	CIFAR-10 - vanilla	CIFAR-10 - STF
Wall-clock time	98.5	101.5
Memory per GPU (G)	36.05	40.64

D.2 SAMPLING

We adopt the RK45 method for the backward ODE sampling of VE, and the DDIM sampler (Song et al., 2021a) for VP. For RK45 sampler of VE, we use the function implemented in `scipy.integrate.solve_ivp` with the tolerances `atol`=$1e-5/1e-4$, `rtol`=$1e-5/1e-4$ for CIFAR-10/CelebA 64^2. As in Song et al. (2021b), we set the terminal time to $1e-5/1e-3$ for VE/VP. For EDM, we adopt Heun's 2nd order method and the discretization scheme in (Karras et al., 2022), with 35 NFE.

We use the predictor-corrector (PC) sampler for reverse-time SDE. We follow Song et al. (2021b) to set the Euler-Maruyama method as the predictor and the Langevin dynamics (MCMC) as the corrector.

D.3 EVALUATIONS

For the evaluation, we compute the Fréchet distance between 50000 samples and the pre-computed statistics of CIFAR-10 and ImageNet 64^2. For CelebA 64^2, we adopt the setting in Song & Ermon (2020) where the distance is computed between 10000 samples and the test set.

E EXTRA EXPERIMENTS

E.1 STABILITY OF CONVERGED MODELS

In Table 5, we report the sample quality measured by FID/Inception score, and their standard deviations across random seeds on CIFAR-10. We can see that models trained with STF objective have lower variations of their final performances, in most cases. In particular, the standard deviation decreases from 4.41 to 0.06 for RK45 sampler on VE. It suggests that the STF objective can stabilize the performance of converged models.

Table 5: CIFAR-10 sample quality (FID, Inception) and number of function evaluation (NFE), with standard deviation.

Methods	Inception ↑	FID ↓	NFE ↓
RK45 method (ODE)			
VE (vanilla)	9.27 ± 0.12	8.90 ± 4.41	264 ± 52
VE (STF)	$\mathbf{9.52 \pm 0.06}$	$\mathbf{5.51 \pm 0.06}$	200 ± 8
PC sampler (SDE)			
VE (vanilla)	9.68 ± 0.12	2.75 ± 0.13	2000
VE (STF)	$\mathbf{9.86 \pm 0.05}$	$\mathbf{2.66 \pm 0.13}$	2000

E.2 EFFECTS OF STEP SIZE

In Figure 6, we show the FID scores with the number of function evaluations of ODE samplers on CIFAR-10 and CelebA 64^2 . To vary the NFE, we adjust the error tolerance in the RK45 method. The sample quality of the STF objective degrades gracefully when decreasing the NFE. The STF objective consistently outperforms the vanilla one for all NFEs on CIFAR-10, and largely improves over the baseline when setting the tolerance to $5e-3$ on CelebA 64^2 . It suggests that the STF has greater robustness to different step sizes.

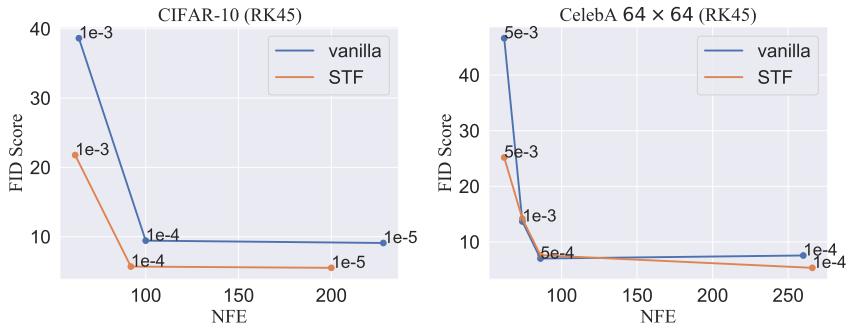


Figure 6: FID versus NFE using ODE samplers on CIFAR-10 (left) and CelebA 64×64 (right)

F EXTENDED SAMPLES

We provide extended samples from score-based models trained by vanilla/STF objective on CIFAR-10 and CelebA 64^2 by ODE samplers. For systematic comparison, we visualize samples from models trained on different seeds. We also provide samples generated by the state-of-the-art model — STF with EDM framework.

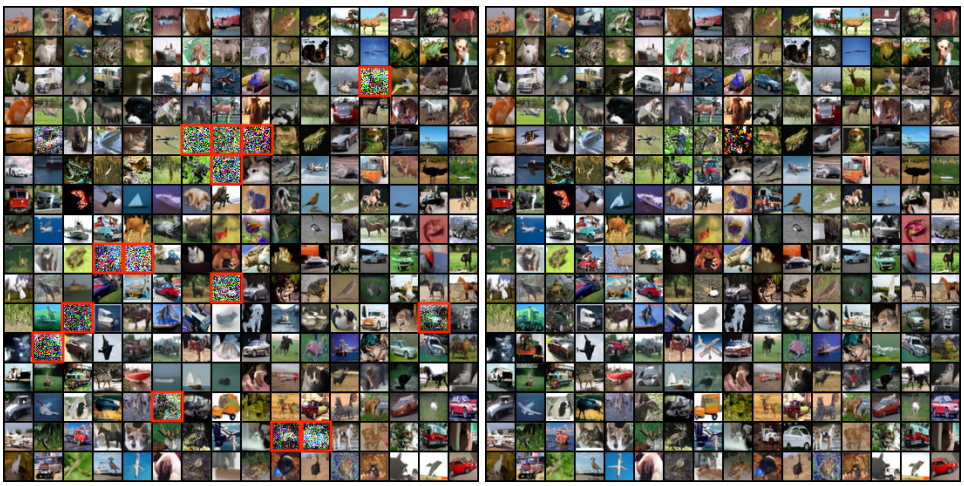
F.1 CIFAR-10

In Figure 7, we visualize the samples produced by different methods across random seeds for VE. We use the RK45 sampler for sampling. We observe that the model trained by the vanilla objective can produce noisy images (in red boxes), and the image quality has great variability across different random seeds. In contrast, models trained by STF objective generate clean and consistent samples with varying random seeds.

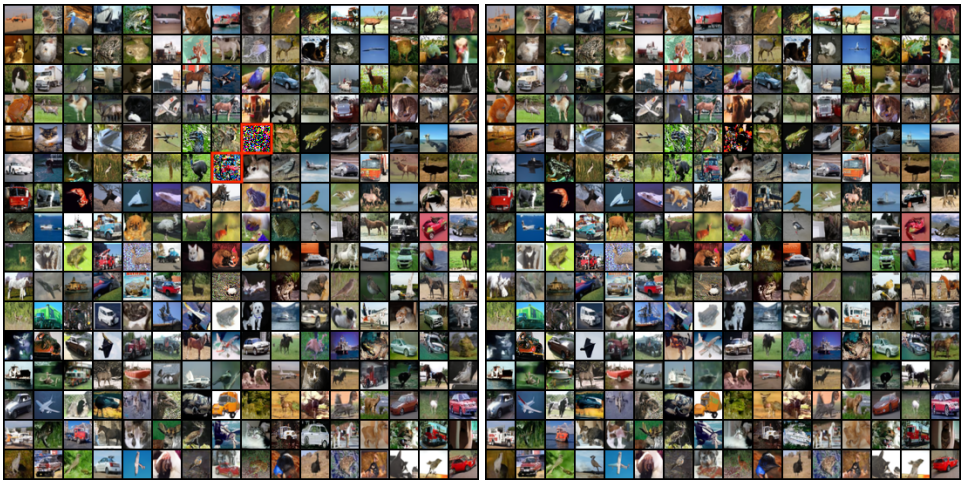
In Figure 8, we further provide samples from a model trained by STF under the EDM framework (Karras et al., 2022). The model is the current state-of-the-art on the unconditional CIFAR-10 generation task.

F.2 CELEBA 64×64

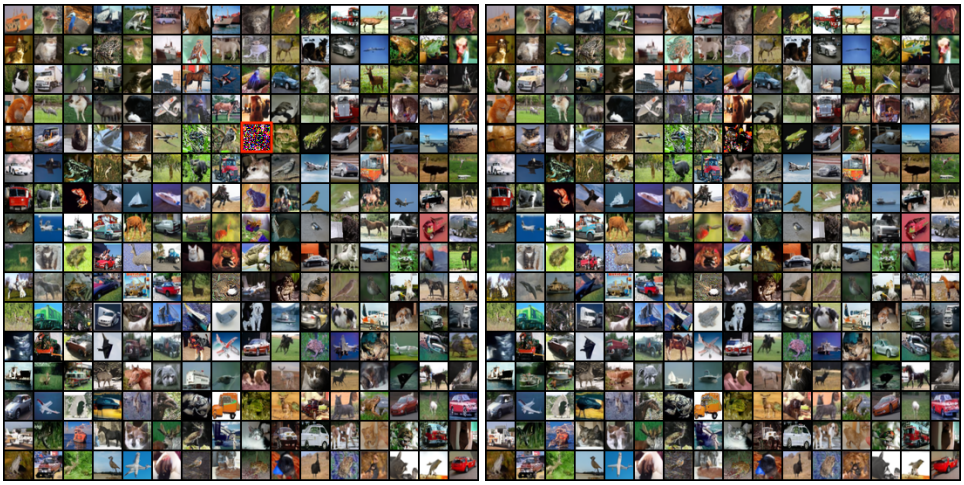
In Figure 9, we provide samples from models trained on vanilla and STF objectives with VE.



(a) vanilla-1 (b) STF-1



(c) vanilla-2 (d) STF-2



(e) vanilla-3 (f) STF-3

Figure 7: Samples generated by three different final models of vanilla (left column) and STF (right column) on CIFAR-10. Red boxes indicate noisy images.

Figure 8: CIFAR-10 samples from STF using EDM model. The FID is 1.90 and NFE is 35.

(a) vanilla

(b) STF

Figure 9: Samples generated by models trained on vanilla (top) and STF (bottom) on CelebA 64^2 with VE.