Stable Target Field for Reduced Variance Score Estimation in Diffusion Models

Pramook Khungurn

April 10, 2023

This note was written as I read the paper "Stable Target Field for Reduced Variance Score Estimation in Diffusion Models" by Xu et al. [XTJ23].

1 Introduction

- This paper argues that the training objective of diffusion models, the **denoising score matching (DSM)** objective, has large variance and leads to suboptimal performance.
- It proposes a generalized version of the objective, called te **stable target field (STF)** objective, that reduces this noise.
  - The idea is to include an additional **reference batch** of examples that are used to calculted weighted conditional scores, which is then use as the target for the score matching objective.
  - The new objective reduces variance but introduces bias.
  - The bias vanishes as the reference batch size increases.
- Results
  - FID of 1.90 on unconditional CIFAR-10 with the EDM settings [KAAL22], an improvement from 1.97.
  - FID improvement on other models such as the VE and VP models of [SSDK+21].
  - Acceltation the training of the VE model on CIFAR-10 by 3.6× with better FID score.

2 Background

- A data item is denoted by $\mathbf{x} \in \mathbb{R}^d$.
- The data distribution is denoted by p_0.
- The forward process is governed by the SDE

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

  where $t \in [0, T]$, $T > 0$, $\mathbf{f} : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$, $g : \mathbb{R} \to \mathbb{R}$, and $\mathbf{w} \in \mathbb{R}^d$ is the standard Wiener process.
- Taking the SDE into account, $\mathbf{x}$ is a now stochastic process, which means that $\mathbf{x}(t)$ is a random variable that depends on time.
  - Let $p_t(\cdot)$ denotes the probability distribution of $\mathbf{x}(t)$.

  - We will also denote $\mathbf{x}(t)$ by $\mathbf{x}_t$.

- According to [KAAL22], we are mostly interested in the SDE where

$$\mathbf{f}(\mathbf{x}, t) = f(t)\mathbf{x}$$

  where $f : \mathbb{R} \to \mathbb{R}$. In this case, we can define

$$\alpha(t) = \exp\left(\int_0^t f(u)\,\mathrm{d}u\right),$$
$$\sigma(t) = \sqrt{\int_0^t \frac{g(u)^2}{\alpha(u)^2}\,\mathrm{d}u}.$$

  Then, we have that

$$p_{t|0}(\mathbf{x}|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}; \alpha(t)\mathbf{x}_0, \sigma^2(t)I)$$

- In a DDPM, we want to train a neural network $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)$ to estimate the score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$.
- The training objective is

$$\min_{\boldsymbol{\theta}} E_{\substack{t\sim q(t),\\ \mathbf{x}_0\sim p_0,\\ \mathbf{x}_t\sim p_{t|0}(\cdot|\mathbf{x}_0)}} \left[\sigma^2(t)\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)\|_2^2\right]$$

  where q_t is the distribution for time variable.

3 Variance of Denoising Score Matching

- Fixing t, the denoising score-matching object becomes

$$\begin{aligned}\ell_{\mathrm{DSM}}(\boldsymbol{\theta}, t) &= \min_{\boldsymbol{\theta}} E_{\substack{\mathbf{x}_0\sim p_0,\\ \mathbf{x}_t\sim p_{t|0}(\cdot|\mathbf{x}_0)}} \left[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)\|_2^2\right] \\ &= \min_{\boldsymbol{\theta}} E_{\mathbf{x}_0\sim p_0}\left[E_{\mathbf{x}_t\sim p_{t|0}(\cdot|\mathbf{x}_0)}\left[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)\|_2^2\right]\right].\end{aligned}$$

  In this objective, we sample $\mathbf{x}_0$ before sampling $\mathbf{x}_t$.

- We can, however, swap the order of sampling so that we sample $\mathbf{x}_t$ before sampling $\mathbf{x}_0$.

$$\ell_{\mathrm{DSM}}(\boldsymbol{\theta}, t) = \min_{\boldsymbol{\theta}} E_{\mathbf{x}_t\sim p_t}\left[E_{\mathbf{x}_0\sim p_{0|t}(\cdot|\mathbf{x}_t)}\left[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)\|_2^2\right]\right].$$

  Here, $\mathbf{s}_{\boldsymbol{\theta}}$ has a closed-form minimizer

$$\mathbf{s}^*_{\mathrm{DSM}}(\mathbf{x}_t, t) = E_{p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)}[\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)] = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t).$$

- What we have been doing is estimating $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ by sampling $\mathbf{x}_0$ according to p_0 and then $\mathbf{x}_t$ according to $p_{t|0}(\cdot|\mathbf{x}_0)$ and taking $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)$ as a single-sample Monte Carlo estimate.
- The above estimate can have high variance, especially when multiple data items in have comparable influences on $\mathbf{x}_t$. This can slow down convergence and degrades performance of the optimized $\mathbf{s}_{\boldsymbol{\theta}}$.
- The paper characterize the variation of the targets at difference times with the following metrics:

$$\begin{aligned}V_{\mathrm{DSM}}(t) &= E_{\mathbf{x}_t\sim p_t}\left[\mathrm{tr}(\mathrm{Cov}_{\mathbf{x}_0\sim p_{0|t}(\cdot,\mathbf{x}_t)}[\nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)])\right] \\ &= E_{\mathbf{x}_t\sim p_t}\left[E_{\mathbf{x}_0\sim p_{0|t}(\cdot|\mathbf{x}_t)}\left[\|\nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\|_2^2\right]\right] \\ &= E_{\mathbf{x}_0\sim p_0}\left[E_{\mathbf{x}_t\sim p_{t|0}(\cdot|\mathbf{x}_0)}\left[\|\nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\|_2^2\right]\right].\end{aligned}$$

- $V_{\text{DSM}}(t)$ is close to 0 at $t = 0$, and it is low at $t = T$. It peaks somewhere between $t = 0$ and $t = T$.
  - The location around where V_{DSM} peaks is called the "intermediate phase" by the paper.
  - The behavior shows up for a toy dataset with 2 Gaussians and CIFAR-10.

4 Stable Target Field

- The ideal target for score matching is given by

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = E_{p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)}\big[\nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\big].$$

- Since it is impractical to sample from $p_{0|t}(\cdot|\mathbf{x}_t)$ directly, we sample $\mathbf{x}_0$ with distribution p_0. Then, we estimate the score as:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \approx \frac{p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)}{p_0(\mathbf{x}_0)} \nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)$$

  If we do this n times, the estimate becomes

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \approx \frac{1}{n} \sum_{i=1}^{n} \frac{p_{0|t}(\mathbf{x}_0^{(i)}|\mathbf{x}_t)}{p_0(\mathbf{x}_0^{(i)})} \nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(i)}).$$

  However, there is one problem with the above formula: we do not know how to compute $p_{0|t}(\mathbf{x}_0^{(i)}|\mathbf{x}_t)$. This can be remedy by appealing to Bayes' rule:

$$\frac{p_{0|t}(\mathbf{x}_0^{(i)}|\mathbf{x}_t)}{p_0(\mathbf{x}_0^{(i)})} = \frac{p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(i)})}{p_t(\mathbf{x}_t)}.$$

  So,

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \approx \frac{1}{n} \sum_{i=1}^{n} \frac{p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(i)})}{p_t(\mathbf{x}_t)} \nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(i)}).$$

  Nevertheless, we do not know an efficient way to compute $p_t(\mathbf{x}_t)$, so we estimate it with:

$$p_t(\mathbf{x}_t) \approx \frac{1}{n} \sum_{j=1}^{n} p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(j)}).$$

  As a result, the score estimate becomes

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \approx \sum_{i=1}^{n} \frac{p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(i)})}{\sum_{j=1}^{n} p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(j)})} \nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(i)}).$$

  This estimate is called the **stable target field (STF)**.

- In practice, we same a reference batch $\mathcal{B}_L = \{\mathbf{x}_0^{(i)}\}_{i=1}^n$ from p_0^n (the probability of sampling n samples from p_0). We then obtain $\mathbf{x}_t$ by corrupting the first element $\mathbf{x}_0^{(1)}$ from the batch. The new object is:

$$\ell_{\text{STF}}(\boldsymbol{\theta}, t) = E_{\{\mathbf{x}_0^{(i)}\}_{i=1}^n \sim p_0^n}\left[E_{\mathbf{x}_t \sim p_{t|0}(\cdot|\mathbf{x}_0^{(1)})}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \sum_{i=1}^{n} \frac{p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(i)})}{\sum_{j=1}^{n} p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(j)})} \nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(i)})\right\|^2\right]\right].$$

- Because $p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\alpha(t)\mathbf{x}_0, \sigma^2(t)I)$, we have that

$$p_t(\mathbf{x}_t|\mathbf{x}_0) \propto \exp\bigg(-\frac{\|\mathbf{x}_t - \alpha(t)\mathbf{x}_0\|^2}{2\sigma^2(t)}\bigg).$$

So,

$$\ell_{\text{STF}}(\boldsymbol{\theta}, t) = E_{\{\mathbf{x}_0^{(i)}\}_{i=1}^n \sim p_0^n}\Bigg[E_{\mathbf{x}_t \sim p_{t|0}(\cdot|\mathbf{x}_0^{(1)})}\Bigg[\Big\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \frac{1}{\sigma_t^2}\sum_{i=1}^n \frac{\exp(-\|\mathbf{x}_t - \alpha_t\mathbf{x}_0^{(i)}\|^2/(2\sigma_t^2))}{\sum_{j=1}^n \exp(-\|\mathbf{x}_t - \alpha_t\mathbf{x}_0^{(j)}\|^2/(2\sigma_t^2))}(\alpha_t\mathbf{x}_0^{(i)} - \mathbf{x}_t)\Big\|^2\Bigg]\Bigg].$$

- The final training object is $E_{t\sim q_t}[\lambda(t)\ell_{\text{STF}(\boldsymbol{\theta},t)}]$.

- In the real training algorithm, however, we do not work with only one $\mathbf{x}_0^{(1)}$. Instead, we sample a large reference batch $\mathcal{B}_L$. From it, we subsample a small batch $\mathcal{B}$. The size of the small batch size is the size of the batch in normal training.

- The full training algorithm is as follows.

  while not satisfied **do**

  Sample a large batch $\mathcal{B}_L$ from the data distribution.

  Sample a small batch $\mathcal{B}$ from the large batch.

  Sample times $t_1, t_2, \ldots, t_{|\mathcal{B}|}$ according to the distribution q_t.

  Compute corrupted examples according to the times being applied to the corresponding elements in the small batch.

  Compute the stable target field for each of the corrupted example in the small batch.

  Calculate the loss and update the model parameters.

  end while

5 Theoretical Results

- Let

$$\mathbf{s}^*_{\text{STF}}(\mathbf{x}_t, t) = E_{\mathbf{x}_0^{(1)}\sim p_{0|T}(\cdot|\mathbf{x}_t)}\Bigg[E_{\{\mathbf{x}_0^{(i)}\}_{i=2}^n \sim p_0^{n-1}}\Bigg[\sum_{i=1}^n \frac{p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(i)})}{\sum_{j=1}^n p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(j)})}\nabla_{\mathbf{x}_t}\log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(i)})\Bigg]\Bigg].$$

  This is the stable target field, which we optimize $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ against.

- **Theorem 1.** *Let $t \in [0, T]$. Suppose $0 < \sigma_t < \infty$. Then,*

$$\sqrt{n}(\mathbf{s}^*_{\text{STF}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)) \to \mathcal{N}\bigg(\mathbf{0}, \frac{\text{Cov}(\nabla_{\mathbf{x}_t}p_{t|0}(\mathbf{x}_t|\mathbf{x}_0))}{p_t(\mathbf{x}_t)^2}\bigg)$$

  where the convergence is in distribution.

  The theorem says that, as $n \to \infty$, we have that $\mathbf{s}^*_{\text{STF}}(\mathbf{x}_t, t)$ approaches $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)$.

- Now, the variance of trace matric of the stable target field.

$$V_{\text{STF}}(t) = E_{\mathbf{x}_t\sim p_t}\Bigg[\text{tr}\bigg(\text{Cov}_{\substack{\mathbf{x}_0^{(1)}\sim p_{0|t}(\cdot|\mathbf{x}_t),\\ \{\mathbf{x}_0^{(i)}\}_{i=2}^n\sim p_0^{n-1}}}\bigg(\sum_{i=1}^n \frac{p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(i)})}{\sum_{j=1}^n p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(j)})}\nabla_{\mathbf{x}_t}\log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0^{(i)})\bigg)\bigg)\Bigg]$$

- **Theorem 2.** *Let $t \in [0, T]$. Suppose $0 < \sigma_t < \infty$. Then,*

$$V_{\mathrm{STF}}(t) \leq \frac{1}{n-1}\left(V_{\mathrm{DSM}}(t) + \frac{\sqrt{3}d}{\sigma_t^2}\sqrt{E_{\mathbf{x}_t \sim p_t}[D_f(p_0(\mathbf{x}_0)) \| p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)]}\right) + O\left(\frac{1}{n^2}\right)$$

where D_f is an f-divergence with

$$f(y) = \begin{cases} 1/(y-1)^2, & y < 1.5 \\ 8y/27 - 1/3, & y \geq 1.5 \end{cases}.$$

When, $n \gg d$ and $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) \approx p_0(\mathbf{x}_0)$ for all $\mathbf{x}_t$, then $V_{\mathrm{STf}}(t) \leq V_{\mathrm{DSM}(t)}/(n-1)$ approximately.

The threorem says that the variance of the stable target field is smaller than the variance of denoising score matching. This happens when t is large, which implies that $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) \approx p_0(\mathbf{x}_0)$.

References

[KAAL22] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022.

[SSDK+21] 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*, 2021.

[XTJ23] Yilun Xu, Shangyuan Tong, and Tommi Jaakkola. Stable target field for reduced variance score estimation in diffusion models, 2023.