CONTENTS

Appendix

A SCORE MATCHING FRAMEWORK

This section is devoted to the derivation of the score based models framework, which can be divided into three parts: forward, backward processes and score approximation.

A.1 FORWARD SDE

Forward SDE can be written as

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

where w_t is the standard Wiener process, which can be roughly understood as $\mathrm{d}\boldsymbol{w}_t \sim \sqrt{\mathrm{d}t} \cdot \mathcal{N}(0, T)$. This equation is usually solved on the interval $t \in [0, 1]$. In the score matching it is assumed that the first term is linear $\boldsymbol{f}(\boldsymbol{x}, t) = f(t)\boldsymbol{x}$. It can be shown that the perturbation kernels of this SDE have the Gaussian distribution:

$$p_t\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right) = \mathcal{N}\left(\boldsymbol{x}_t \mid \boldsymbol{\mu}(\boldsymbol{x}_0, t), \sigma^2(t)\mathbf{I}\right). \tag{2}$$

We use the notation $\boldsymbol{x}_t = \boldsymbol{x}(t)$ for simplicity. Here $\boldsymbol{x}_0$ is the initial condition of the forward SDE (10). In terms of deep learning this is a sample from a dataset, $\boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x}_0)$. The parameters, $\boldsymbol{\mu}(\boldsymbol{x}_0, t)$ and $\sigma^2(t)$, can be found using the Ito formula.

Proposition 1. *Parameters of the perturbation kernels (2) caused by the forward SDE (10) can be found from solution of the following ODEs:*

$$\begin{cases} \mathrm{d}\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x} = \mathbb{E}_{\boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}, t)\mathrm{d}t, \\ \mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}(0) = \boldsymbol{x}_0. \end{cases}$$
$$\begin{cases} \mathrm{d}\,\mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x} = \left(2\mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{x}\boldsymbol{f}(\boldsymbol{x}, t) - \boldsymbol{f}(\boldsymbol{x}, t)\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}\right] + g(t)^2\right)\mathrm{d}t, \\ \mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x}(0) = 0. \end{cases} \tag{3}$$

Proof. Let suppose that $\phi(\boldsymbol{x}, t) : \mathbb{R}^n \to \mathbb{R}$ is an arbitrary twice differentiable function, where $\boldsymbol{x}$ is a solution of the (10). Thus, we can write the Ito formula for the ϕ:

$$\begin{aligned} \mathrm{d}\phi &= \frac{\partial\phi}{\partial t}\mathrm{d}t + \sum_{i=1}^{n}\frac{\partial\phi}{\partial x_i}\mathrm{d}x_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial^2\phi}{\partial x_i \partial x_j}\mathrm{d}x_i\mathrm{d}x_j + ... \\ &= \frac{\partial\phi}{\partial t}\mathrm{d}t + \sum_{i=1}^{n}\frac{\partial\phi}{\partial x_i}\boldsymbol{f}^i\mathrm{d}t + \frac{1}{2}g(t)^2\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial^2\phi}{\partial x_i \partial x_j}\mathrm{d}t + g(t)\sum_{i=1}^{n}\frac{\partial\phi}{\partial x_i}\mathrm{d}\boldsymbol{w}_t^i, \end{aligned} \tag{4}$$

where $\boldsymbol{f}^i$ denotes the i-th component of the vector-valued function $\boldsymbol{f}$. Note that we omitted the arguments for simplicity. Then we can take the expectation from both parts:

$$\begin{aligned} \mathrm{d}\mathbb{E}_{\boldsymbol{x}}\phi &= \mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial\phi}{\partial t}\right)\mathrm{d}t + \frac{1}{2}g(t)^2\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial^2\phi}{\partial x_i \partial x_j}\right)\mathrm{d}t \\ &+ \sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial\phi}{\partial x_i}\boldsymbol{f}^i\right)\mathrm{d}t + g(t)\sum_{i=1}^{n}\underbrace{\mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial\phi}{\partial x_i}\mathrm{d}\boldsymbol{w}_t^i\right)}_{\mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial\phi}{\partial x_i}\right)\mathbb{E}\left(\mathrm{d}\boldsymbol{w}_t^i\right)=0} \\ &= \mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial\phi}{\partial t}\right)\mathrm{d}t + \frac{1}{2}g(t)^2\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial^2\phi}{\partial x_i \partial x_j}\right)\mathrm{d}t + \sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial\phi}{\partial x_i}\boldsymbol{f}^i\right)\mathrm{d}t. \end{aligned} \tag{5}$$

Let say that $\phi = \boldsymbol{x}^k$, then

$$\mathrm{d}\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}^k = \underbrace{\mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial\boldsymbol{x}_k}{\partial t}\right)}_{=0}\mathrm{d}t + \frac{1}{2}g(t)^2\sum_{i=1}^{n}\sum_{j=1}^{n}\underbrace{\mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial^2\boldsymbol{x}^k}{\partial x_i \partial x_j}\right)}_{=0}\mathrm{d}t + \sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial\boldsymbol{x}^k}{\partial x_i}\boldsymbol{f}^i\right)\mathrm{d}t = \mathbb{E}_{\boldsymbol{x}}\boldsymbol{f}^k\mathrm{d}t. \tag{6}$$

Finally, we obtain the first equation in the (3)

$$\mathrm{d}\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x} = \mathbb{E}_{\boldsymbol{x}}\boldsymbol{f}\mathrm{d}t \tag{7}$$

If we put $\phi = \boldsymbol{x}_k^2 - \left(\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}_k\right)^2$, then we will obtain the second equation.

$$\begin{aligned}\mathrm{d}\mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{x}_k^2 - \left(\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}_k\right)^2\right] = \mathbb{E}_{\boldsymbol{x}}\underbrace{\left(\frac{\partial\left[\boldsymbol{x}_k^2 - \left(\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}_k\right)^2\right]}{\partial t}\right)}_{=-2\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}_k\cdot\mathbb{E}_{\boldsymbol{x}}\boldsymbol{f}_k}\mathrm{d}t + \frac{1}{2}g(t)^2\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial^2\left[\boldsymbol{x}_k^2 - \left(\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}_k\right)^2\right]}{\partial x_i\partial x_j}\right)}_{=2}\mathrm{d}t \\ + \underbrace{\sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{x}}\left(\frac{\partial\left[\boldsymbol{x}_k^2 - \left(\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}_k\right)^2\right]}{\partial x_i}\boldsymbol{f}^i\right)}_{2\mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}_k\boldsymbol{f}_k)}\mathrm{d}t.\end{aligned} \tag{8}$$

Finally, we have the following

$$\mathrm{d}\,\mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x} = \left(2\mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{x}\boldsymbol{f}(\boldsymbol{x},t) - \boldsymbol{f}(\boldsymbol{x},t)\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}\right] + g(t)^2\right)\mathrm{d}t. \tag{9}$$

□

A.1.1 VESDE AND VPSDE

Two the most well known types of the forward SDE are VESDE (Variance Exploding SDE) and VPSDE (Variance Preserving SDE). They can be formulated as the following:

$$\mathrm{d}x = f(t)\boldsymbol{x}\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}_t, \tag{10}$$

VPSDE.

$$f(t) = \frac{1}{2}\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t},\ g(t) = \sqrt{\left(-\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\right)}, \tag{11}$$

where α_t is a bounded decreasing sequence from $\alpha_0 = 1$ to $\alpha_1 = 0$. The parameters of the perturbation kernels can be found from the (3)

$$\begin{aligned}&\mathrm{d}\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x} = \mathbb{E}_{\boldsymbol{x}}\frac{1}{2}\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\boldsymbol{x}\mathrm{d}t = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}\frac{\mathrm{d}\alpha_t}{\alpha_t}, \\ &\int_0^\tau\frac{\mathrm{d}\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}}{\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}} = \int_0^\tau\frac{1}{2}\frac{\mathrm{d}\alpha_t}{\alpha_t} \to \mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}(\tau) = \sqrt{\alpha_\tau}\,\boldsymbol{x}_0.\end{aligned} \tag{12}$$

$$\begin{aligned}&\mathrm{d}\,\mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x} = \left(2\mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{x}^2\frac{1}{2}\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t} - \frac{1}{2}\boldsymbol{x}\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}\right] - \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\right)\mathrm{d}t, \\ &\mathrm{d}\,\mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x} = \left(\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}^2 - \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}(\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x})^2 - \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\right)\mathrm{d}t, \\ &\mathrm{d}\,\mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x} = \left(\mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x} - 1\right)\frac{\mathrm{d}\alpha_t}{\alpha_t} \to \int_0^\tau\frac{\mathrm{d}\,\mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x}}{\mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x} - 1} = \int_0^\tau\frac{\mathrm{d}\alpha_t}{\alpha_t}, \\ &\mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x}(\tau) - 1 = \left(\mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x}(0) - 1\right)\alpha_\tau \to \mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x}(\tau) = 1 - \alpha_\tau.\end{aligned} \tag{13}$$

VESDE.

$$f(t) = 0,\ g(t) = \sqrt{\frac{\mathrm{d}s_t^2}{\mathrm{d}t}}, \tag{14}$$

where s_t is an unbounded increasing sequence, $s_0 = 0$, $s_1 = c$

$$\begin{aligned}&\mathrm{d}\mathbb{E}_{\boldsymbol{x}}\boldsymbol{x} = 0 \to \mathbb{E}_{\boldsymbol{x}}\boldsymbol{x}(\tau) = \boldsymbol{x}_0, \\ &\int_0^\tau\mathrm{d}\,\mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x} = \int_0^\tau\mathrm{d}s_t^2 \to \mathbb{D}_{\boldsymbol{x}}\,\boldsymbol{x}(\tau) = s_\tau^2.\end{aligned} \tag{15}$$

To summarize, the transition kernels take the following form:

$$p_t\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right) = \begin{cases} \textbf{VPSDE} : \mathcal{N}\left(\boldsymbol{x}_t \mid \sqrt{\alpha_t}\, \boldsymbol{x}_0, 1-\alpha_t\right), \alpha_0 = 1, \alpha_1 = 0, \\ \textbf{VESDE} : \mathcal{N}\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0, s_t^2\right), s_0 = 0, s_1 = c. \end{cases} \tag{16}$$

Here we deal with the Variance-Exploding scheme. So, our main equations defining the noising process are the following:

$$\boxed{\begin{array}{c} \mathrm{d}x = \sqrt{\dfrac{\mathrm{d}s_t^2}{\mathrm{d}t}}\mathrm{d}\boldsymbol{w}_t = \mathrm{d}\boldsymbol{w}_{s_t^2}, \\ p_t\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right) = \mathcal{N}\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0, s_t^2\right). \end{array}} \tag{17}$$

A.2 Backward SDE and ODE

A remarkable result from Anderson (1982) states that the reverse of a diffusion process is also a diffusion process, running backwards in time and given by the reverse-time SDE:

$$\mathrm{d}\boldsymbol{x} = \left[\boldsymbol{f}(\boldsymbol{x}, t) - g(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] \mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}_t, \tag{18}$$

In the literature an explicit form of the score function is defined by noise, applied to a clean image, or clean image itself. We are working with the latter one, which can be obtained by Tweedie's formula.

Proposition 2. *(Tweedie's formula) The score function produced by Eq. (17) has the following form*

$$\begin{aligned} \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) &= -\frac{1}{s_t^2}\left(\boldsymbol{x} - \mathbb{E}\left[\boldsymbol{x}_0 | \boldsymbol{x}\right]\right), \\ \mathbb{E}\left[\boldsymbol{x}_0 | \boldsymbol{x}\right] &= \int_{\mathbb{R}^n} \boldsymbol{x}_0 p_t(\boldsymbol{x}_0 | \boldsymbol{x}) \mathrm{d}\boldsymbol{x}_0. \end{aligned} \tag{19}$$

Proof. Firstly, according to the chain rule, the following is truth

$$\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) = \frac{\nabla_{\boldsymbol{x}} p_t(\boldsymbol{x})}{p_t(\boldsymbol{x})}. \tag{20}$$

Then, we have to deal with $\nabla_{\boldsymbol{x}} p_t(\boldsymbol{x})$:

$$\begin{aligned} p_t(\boldsymbol{x}) &= \int_{\mathbb{R}^n} p_t(\boldsymbol{x}|\boldsymbol{x}_0) p_{\text{data}}(\boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0,\ p_t(\boldsymbol{x}|\boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{x}_0, s_t^2\right), \\ \nabla_{\boldsymbol{x}} p_t(\boldsymbol{x}) &= \nabla_{\boldsymbol{x}} \int_{\mathbb{R}^n} p_t(\boldsymbol{x}|\boldsymbol{x}_0) p_{\text{data}}(\boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0 \\ &= \int_{\mathbb{R}^n} p_{\text{data}}(\boldsymbol{x}_0) \nabla_{\boldsymbol{x}} p_t(\boldsymbol{x}|\boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0 \\ &= \int_{\mathbb{R}^n} p_{\text{data}}(\boldsymbol{x}_0) \nabla_{\boldsymbol{x}} \left(\left[2\pi s_t^2\right]^{-\frac{n}{2}} \exp \frac{\|\boldsymbol{x} - \boldsymbol{x}_0\|_2^2}{-2s_t^2} \right) \mathrm{d}\boldsymbol{x}_0 \\ &= \int_{\mathbb{R}^n} p_{\text{data}}(\boldsymbol{x}_0) \left[2\pi s_t^2\right]^{-\frac{n}{2}} \exp \frac{\|\boldsymbol{x} - \boldsymbol{x}_0\|_2^2}{-2s_t^2} \nabla_{\boldsymbol{x}} \left(\frac{\|\boldsymbol{x} - \boldsymbol{x}_0\|_2^2}{-2s_t^2} \right) \mathrm{d}\boldsymbol{x}_0 \\ &= \int_{\mathbb{R}^n} p_{\text{data}}(\boldsymbol{x}_0) p_t(\boldsymbol{x}|\boldsymbol{x}_0) \nabla_{\boldsymbol{x}} \left(\frac{\|\boldsymbol{x} - \boldsymbol{x}_0\|_2^2}{-2s_t^2} \right) \mathrm{d}\boldsymbol{x}_0 \\ &= \int_{\mathbb{R}^n} p_{\text{data}}(\boldsymbol{x}_0) p_t(\boldsymbol{x}|\boldsymbol{x}_0) \left[\frac{\boldsymbol{x}_0 - \boldsymbol{x}}{s_t^2} \right] \mathrm{d}\boldsymbol{x}_0 \\ &= -\frac{1}{s_t^2} \left(\int_{\mathbb{R}^n} \boldsymbol{x} p_{\text{data}}(\boldsymbol{x}_0) p_t(\boldsymbol{x}|\boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0 - \int_{\mathbb{R}^n} \boldsymbol{x}_0 p_{\text{data}}(\boldsymbol{x}_0) p_t(\boldsymbol{x}|\boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0 \right) \\ &= -\frac{1}{s_t^2} \left(\boldsymbol{x} \int_{\mathbb{R}^n} p_t(\boldsymbol{x}, \boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0 - \int_{\mathbb{R}^n} \boldsymbol{x}_0 p_t(\boldsymbol{x}, \boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0 \right) \\ &= -\frac{1}{s_t^2} \left(\boldsymbol{x} p_t(\boldsymbol{x}) - \int_{\mathbb{R}^n} \boldsymbol{x}_0 p_t(\boldsymbol{x}_0|\boldsymbol{x}) p_t(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}_0 \right) \\ &= -\frac{p_t(\boldsymbol{x})}{s_t^2} \left(\boldsymbol{x} - \int_{\mathbb{R}^n} \boldsymbol{x}_0 p_t(\boldsymbol{x}_0|\boldsymbol{x}) \mathrm{d}\boldsymbol{x}_0 \right). \end{aligned} \tag{21}$$

So, finally we have the following:

$$\begin{aligned}\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) = \frac{\nabla_{\boldsymbol{x}} p_t(\boldsymbol{x})}{p_t(\boldsymbol{x})} = -\frac{1}{s_t^2}\left(\boldsymbol{x} - \int_{\mathbb{R}^n} \boldsymbol{x}_0 p_t(\boldsymbol{x}_0|\boldsymbol{x})\mathrm{d}\boldsymbol{x}_0\right), \\ \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) = -\frac{1}{s_t^2}\left(\boldsymbol{x} - \mathbb{E}\left[\boldsymbol{x}_0|\boldsymbol{x}\right]\right).\end{aligned} \tag{22}$$

□

The backward SDE has the following form, taking into account the obtained formula and Variance-Exploding scheme

$$\mathrm{d}\boldsymbol{x} = 2\left(\boldsymbol{x} - \mathbb{E}\left[\boldsymbol{x}_0|\boldsymbol{x}\right]\right)\frac{\mathrm{d}s_t}{s_t} + \mathrm{d}\boldsymbol{w}_{s_t^2}, \tag{23}$$

where $\mathrm{d}\boldsymbol{w}_{s_t^2}$ means the standard Wiener process with changed variable, roughly speaking, $\mathrm{d}\boldsymbol{w}_{s_t^2} \sim \sqrt{\mathrm{d}s_t^2} \cdot \mathcal{N}(0, 1)$.

However, there is a backward ODE, providing the same marginal probability densities, p_t, as the backward SDE, which in practice usually gives better results in terms of quality and efficiency.

$$\mathrm{d}\boldsymbol{x} = \left[\boldsymbol{f}(\boldsymbol{x}, t) - \frac{1}{2}g(t)^2\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right]\mathrm{d}t, \tag{24}$$

or in the case of VE scheme and Tweedie's formula

$$\boxed{\mathrm{d}\boldsymbol{x} = \left(\boldsymbol{x} - \mathbb{E}\left[\boldsymbol{x}_0|\boldsymbol{x}\right]\right)\frac{\mathrm{d}s_t}{s_t}.} \tag{25}$$

A.3 Approximation of the score function

The conditional expectation, $\mathbb{E}\left[\boldsymbol{x}_0|\boldsymbol{x}\right]$, cannot be calculated analytically since the conditional distribution, $p_t(\boldsymbol{x}_0|\boldsymbol{x})$, is unavailable. Thus, we have to utilize a neural network to approximate it. To this end, a least square regression problem is solved.

$$\begin{aligned}\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_0)p_t(\boldsymbol{x}|\boldsymbol{x}_0)}\|\boldsymbol{x}_0 - \boldsymbol{x}_\theta(\boldsymbol{x}, t)\|_2^2, \\ \mathcal{L}(\theta) \to \min_\theta\end{aligned} \tag{26}$$

Then, we approximate the conditional expectation using a trained neural network $\mathbb{E}\left[\boldsymbol{x}_0|\boldsymbol{x}\right] \approx \boldsymbol{x}_{\theta^*}(\boldsymbol{x}, t)$.

A.4 Summary

Below we summarize three main parts of the score-based models.

$$\boxed{\begin{array}{c}\textbf{1. Forward process} \\ \mathrm{d}x = \mathrm{d}\boldsymbol{w}_{s_t^2}, \\ p_t\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right) = \mathcal{N}\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0, s_t^2\right). \\ \\ \textbf{2. Backward process} \\ \mathrm{d}\boldsymbol{x} = -\frac{1}{2}\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\mathrm{d}s_t^2, \\ \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) = -\frac{1}{s_t^2}\left(\boldsymbol{x} - \mathbb{E}\left[\boldsymbol{x}_0|\boldsymbol{x}\right]\right). \\ \\ \textbf{3. Score approximation} \\ \mathbb{E}_{t\sim\mathcal{U}(0,1)}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_0)p_t(\boldsymbol{x}|\boldsymbol{x}_0)}\|\boldsymbol{x}_0 - \boldsymbol{x}_\theta(\boldsymbol{x}, t)\|_2^2 \to \min_\theta, \\ \mathbb{E}\left[\boldsymbol{x}_0|\boldsymbol{x}\right] \approx \boldsymbol{x}_{\theta^*}(\boldsymbol{x}, t).\end{array}} \tag{27}$$

B FAST SOLVERS

In this paper we consider two the most popular types of fast solvers, EDM and DPM-Solver, which are currently the state-of-the-art in the image generation.

DPM-Solver. DPM-Solver is heavily based on the well-known in the ODE literature exponential integrators methods. Firstly, we can rewrite Eq. (24) using Proposition 2, reparameterization through the noise $\epsilon(x,t) = \frac{1}{s_t}\left(\boldsymbol{x} - \mathbb{E}\left[\boldsymbol{x}_0|\boldsymbol{x}\right]\right)$ and the semi-linear structure of the diffusion ODE.

$$\begin{aligned} \mathrm{d}\boldsymbol{x} &= \boldsymbol{f}(\boldsymbol{x},t)\mathrm{d}t - \frac{1}{2}g(t)^2\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})\mathrm{d}t \\ &= f(t)\boldsymbol{x}\mathrm{d}t + \frac{g(t)^2}{2s_t}\epsilon(x,t)\mathrm{d}t. \end{aligned} \tag{28}$$

The exponential integrator method enter an auxiliary variable, in general case called as *transition matrix*, $\phi(s,t)$. Denoting s, τ as the starting and ending points, respectively.

$$\begin{cases} \mathrm{d}\left(\phi(t,\tau)\boldsymbol{x}\right) = \phi(t,\tau)\dfrac{g(t)^2}{2s_t}\epsilon(x,t)\mathrm{d}t, \\ \dfrac{\partial\phi(r,t)}{\partial r} = f(r)\phi(r,t),\ \phi(t,t) = 1. \end{cases} \tag{29}$$

$$\begin{cases} \displaystyle\int_s^\tau \mathrm{d}\left(\phi(t,\tau)\boldsymbol{x}\right) = \int_s^\tau \phi(t,\tau)\frac{g(t)^2}{2s_t}\epsilon(x,t)\mathrm{d}t, \\ \phi(s,t) = e^{\int_s^t f(r)\mathrm{d}r}. \end{cases} \tag{30}$$

Solving the above system of equations, one can obtain the following:

$$\boldsymbol{x}_\tau = e^{\int_s^\tau f(r)\mathrm{d}r}\boldsymbol{x}_s + \int_s^\tau \left(e^{\int_t^\tau f(r)\mathrm{d}r}\frac{g(t)^2}{2s_t}\epsilon(x,t)\right)\mathrm{d}t. \tag{31}$$

Transition matrix can be calculated analytically for a chosen scheme (VP or VE), however an exact form of $\epsilon(x,t)$ is unknown, which makes the second integral intractable. The authors of DPM-solver use the Taylor expansion of $\epsilon(x,t)$ w.r.t t.

EDM. This framework proposes to reformulate the previous ODE/SDE (24, 18) using $\boldsymbol{f}(\boldsymbol{x},t) = 0$ and $g(t) = \sqrt{2t}$.

$$\begin{aligned} \mathrm{d}\boldsymbol{x} &= -t\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})\mathrm{d}t, \\ \mathrm{d}\boldsymbol{x} &= -t\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})\mathrm{d}t + \beta(t)t^2\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})\mathrm{d}t + \sqrt{2\beta(t)}t\mathrm{d}t. \end{aligned} \tag{32}$$

To approximate the above equations authors utilize Heun's 2nd order method and the following schedule design:

$$t_{i<N} = \left(t_{\text{max}}^{1/\rho} + \frac{i}{N-1}\left(t_{\text{min}}^{1/\rho} - t_{\text{max}}^{1/\rho}\right)\right)^\rho,\ t_N = 0. \tag{33}$$

This choice was motivated by truncation error analysis. Here N is the number of sampling steps, $\rho = 7$ and $t_{\text{max}} = 80, t_{\text{min}} = 0.002$.

At this moment, the described solvers are the state-of-the-art in the field of image generation. Thus, in our experiments we adhere to the settings described in the corresponding papers.

C DISTILLATION TECHNIQUES

In this work we consider three the most popular distillation techniques: knownledge, progressive and consistency distillation.

Consistency distillation. This type of distillation introduces a new type of generative models called *consistency models*. Assume that $\boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x}_0)$ is a sample from a real data distribution and $p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$ is transition kernels caused by the forward process (**??**). Then consistency model is $f(\boldsymbol{x}_t, t) = \boldsymbol{x}_0,\ \forall t$.

Transition from diffusion to consistency model occurs due to the self-consistency property.

$$f(\boldsymbol{x}_{t_n}, t_n) = f(\boldsymbol{x}_{t_{n-1}}, t_{n-1}) = ... = f(\boldsymbol{x}_{t_0}, t_0). \tag{34}$$

In practice we enforce this property solving the following optimization problem:

$$\mathcal{L}^{\text{CD}}(\theta) = \mathbb{E}_{n\sim\mathcal{U}(2,N)}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_0)p_{t_n}(\boldsymbol{x}|\boldsymbol{x}_0)q^{\phi}_{t_{n-1}}(\hat{\boldsymbol{x}}|\boldsymbol{x})}\|\boldsymbol{x}_{\theta^-}(\hat{\boldsymbol{x}}, t_{n-1}) - \boldsymbol{x}_{\theta}(\boldsymbol{x}, t_n)\|_2^2 \to \min_{\theta}, \tag{35}$$

where $q^{\phi}_{t_{n-1}}(\hat{\boldsymbol{x}}|\boldsymbol{x})$ is a single step of ODE solver, from t_n to t_{n-1}, using a fixed pretrained diffusion model with parameters ϕ, $\theta^- = \mathbf{EMA}(\theta)$ and $\boldsymbol{x}_\theta(\boldsymbol{x}, t_n)$ is initialized from a pretrained diffusion model that approximates the conditional expectation 2. Moreover, the consistency model is parameterized as $\boldsymbol{x}_\theta(\boldsymbol{x}, t_1) = \boldsymbol{x}$.

It was shown that, under some regularity conditions, the difference between the estimated and real consistency models becomes infinitely small.

$$\sup_{n,\boldsymbol{x}} \|\boldsymbol{x}_\theta(\boldsymbol{x}, t_n) - f(\boldsymbol{x}, t_n)\| = O(\Delta t^p), \tag{36}$$

where $p + 1$ is the local error of ODE solver and $\Delta t = \max_n |t_{n+1} - t_n|$.

After the model is distilled, stochastic sampling is used to produce images, one step of which has the following form:

$$\begin{aligned} \boldsymbol{x}_{t_n} &\sim p_t(\boldsymbol{x}_{t_n}|\boldsymbol{x}_0), \\ \boldsymbol{x}_0 &= \boldsymbol{x}_\theta(\boldsymbol{x}, t_n). \end{aligned} \tag{37}$$

Progressive distillation. This strategy consists of several stages at each of which distilled model aims to halve the number of steps as a teacher model. A single stage can be written as the following:

$$\mathcal{L}^{\text{PD}}(\theta) = \mathbb{E}_{n\sim\mathcal{U}(3,N)}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_0)p_{t_n}(\boldsymbol{x}|\boldsymbol{x}_0)q^{\phi}_{t_{n-2}}(\hat{\boldsymbol{x}}|\boldsymbol{x})}\|\boldsymbol{x}_\theta(\boldsymbol{x}, t_n) - \boldsymbol{x}'\|_2^2 \to \min_{\theta}. \tag{38}$$

Here $q^{\phi}_{t_{n-2}}$ is two steps ($t_n \to t_{n-1} \to t_{n-2}$) of the DDIM sampler using the teacher model with parameters ϕ. Concretely,

$$\begin{aligned} q^{\phi}_{t_{n-2}}(\hat{\boldsymbol{x}}|\boldsymbol{x}) &= \delta(\hat{\boldsymbol{x}} - \boldsymbol{x}_{t_{n-2}}(\boldsymbol{x})), \\ \boldsymbol{x}_{t_{n-2}}(\boldsymbol{x}) &= \boldsymbol{x}_\phi(\boldsymbol{x}_{t_{n-1}}) + s_{t_{n-2}}\frac{\boldsymbol{x}_{t_{n-1}} - \boldsymbol{x}_\phi(\boldsymbol{x}_{t_{n-1}})}{s_{t_{n-1}}}, \\ \boldsymbol{x}_{t_{n-1}} &= \boldsymbol{x}_\phi(\boldsymbol{x}) + s_{t_{n-1}}\frac{\boldsymbol{x} - \boldsymbol{x}_\phi(\boldsymbol{x})}{s_{t_n}}, \end{aligned} \tag{39}$$

where $\boldsymbol{x}_\phi$ is the teacher model. Note that to obtain this discretization we use VE scheme (27) and the Euler integration rule. The target for distilled model, $\boldsymbol{x}'$, can be derived from the consideration that during distillation we aim to approximate two steps of the DDIM sampler in only one step.

$$\begin{aligned} \boldsymbol{x}'_{t_{n-2}} &= \boldsymbol{x}' + s_{t_{n-2}}\frac{\boldsymbol{x} - \boldsymbol{x}'}{s_{t_n}} = \boldsymbol{x}_{t_{n-2}}, \\ \to\ \boldsymbol{x}' &= \frac{\boldsymbol{x}_{t_{n-2}} - s_{t_{n-2}}/s_{t_n}\boldsymbol{x}}{1 - s_{t_{n-2}}/s_{t_n}}. \end{aligned} \tag{40}$$

After the model is distilled, the DDIM solver is used to generate samples.

Knowledge distillation. Knowledge distillation is the most simplest technique among considered, but at the same time the most computationally expensive. In this case the optimization problem formulates as follows:

$$\mathcal{L}^{\text{KD}}(\theta) = \mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_0)p_{t_N}(\boldsymbol{x}|\boldsymbol{x}_0)q^{\phi}_{t_0}(\hat{\boldsymbol{x}}|\boldsymbol{x})}\|\boldsymbol{x}_\theta(\boldsymbol{x}, t_N) - \hat{\boldsymbol{x}}\|_2^2 \to \min_{\theta}. \tag{41}$$

Here $q^{\phi}_{t_0}(\hat{\boldsymbol{x}}|\boldsymbol{x})$ is N steps of ODE solver, from t_N to 0. Beyond computational complexity, the distilled model loses its ability to multi-step generation which leads to a quality degradation. To fix

this problem, we consider the following modification (Multi-Step Knowledge Distillation) of the knowledge distillation:

$$\mathcal{L}^{\text{MSKD}}(\theta) = \mathbb{E}_{n\sim\mathcal{U}(2,N)p_{\text{data}}(\boldsymbol{x}_0)p_{t_N}(\boldsymbol{x}|\boldsymbol{x}_0)q^{\phi}_{t_0}(\hat{\boldsymbol{x}}|\boldsymbol{x})q^{\theta}_{t_n}(\boldsymbol{x'}|\boldsymbol{x})}\|\boldsymbol{x}_\theta(\boldsymbol{x}', t_n) - \hat{\boldsymbol{x}}\|_2^2 \to \min_\theta. \tag{42}$$

In this case we additionally train the model on the intermediate timesteps, t_n, and not only on the last one, t_N. This modification allows us to utilize ODE solver at the inference phase. In our experiments we use 1-6 steps of DDIM sampler.

D Analysis

D.1 Properties of the approximated score function

To understand the properties of the optimized network and, as a consequence, the approximated score function, we can obtain the optimal solution of Eq. (26) analytically.

Proposition 3. *(Karras) The optimal solution to the least squares regression problem Eq. (26) has the following form:*

$$\boldsymbol{x}_\theta(\boldsymbol{x}, t) = \frac{\sum_{j=1}^N \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0^j, s_t^2\right)\boldsymbol{x}_0^j}{\sum_{j=1}^N \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0^j, s_t^2\right)}. \tag{43}$$

Proof. We assume that $p_{\text{data}}(\boldsymbol{x}_0)$ is an experimental distribution, consisting of N samples, $\mathbf{X} = \{\boldsymbol{x}_0^1, ..., \boldsymbol{x}_0^N\}$. In other words

$$p_{\text{data}}(\boldsymbol{x}_0) = \frac{1}{N}\sum_{j=1}^N \delta(\boldsymbol{x}_0 - \boldsymbol{x}_0^j). \tag{44}$$

Then,

$$\begin{aligned}
\mathcal{L}(\boldsymbol{x}_\theta) &= \mathbb{E}_{t\sim\mathcal{U}(0,1)}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_0)}\int_{\mathbb{R}^n}\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0, s_t^2\right)\|\boldsymbol{x}_0 - \boldsymbol{x}_\theta(\boldsymbol{x}, t)\|_2^2\mathrm{d}\boldsymbol{x} \\
&= \frac{1}{N}\sum_{j=1}^N\int_0^1 \mathcal{U}(t|0,1)\mathrm{d}t\int_{\mathbb{R}^n}\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0^j, s_t^2\right)\|\boldsymbol{x}_0^j - \boldsymbol{x}_\theta(\boldsymbol{x}, t)\|_2^2\mathrm{d}\boldsymbol{x} \\
&= \int_0^1 \mathrm{d}t\int_{\mathbb{R}^n}\frac{1}{N}\sum_{j=1}^N\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0^j, s_t^2\right)\|\boldsymbol{x}_0^j - \boldsymbol{x}_\theta(\boldsymbol{x}, t)\|_2^2\mathrm{d}\boldsymbol{x} \\
&= \int_0^1 \mathrm{d}t\int_{\mathbb{R}^n}\mathcal{L}(\boldsymbol{x}_\theta, \boldsymbol{x}, t)\mathrm{d}\boldsymbol{x},
\end{aligned} \tag{45}$$

$\mathcal{L}(\theta)$ can be optimized independently for given $\boldsymbol{x}$ and t. That is, we have to find the following minimum:

$$\mathcal{L}(\boldsymbol{x}_\theta, \boldsymbol{x}, t) \to \min_{\boldsymbol{x}_\theta}. \tag{46}$$

Taking into account the convexity of the problem

$$\begin{aligned}
\nabla_{\boldsymbol{x}_\theta}\mathcal{L}(\boldsymbol{x}_\theta, \boldsymbol{x}, t) &= 0, \\
\nabla_{\boldsymbol{x}_\theta}\mathcal{L}(\boldsymbol{x}_\theta, \boldsymbol{x}, t) &= \nabla_{\boldsymbol{x}_\theta}\left(\frac{1}{N}\sum_{j=1}^N\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0^j, s_t^2\right)\|\boldsymbol{x}_0^j - \boldsymbol{x}_\theta(\boldsymbol{x}, t)\|_2^2\right) \\
&= \frac{1}{N}\sum_{j=1}^N\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0^j, s_t^2\right)\nabla_{\boldsymbol{x}_\theta}\|\boldsymbol{x}_0^j - \boldsymbol{x}_\theta(\boldsymbol{x}, t)\|_2^2 \\
&= \frac{-2}{N}\sum_{j=1}^N\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0^j, s_t^2\right)\left[\boldsymbol{x}_0^j - \boldsymbol{x}_\theta(\boldsymbol{x}, t)\right].
\end{aligned} \tag{47}$$

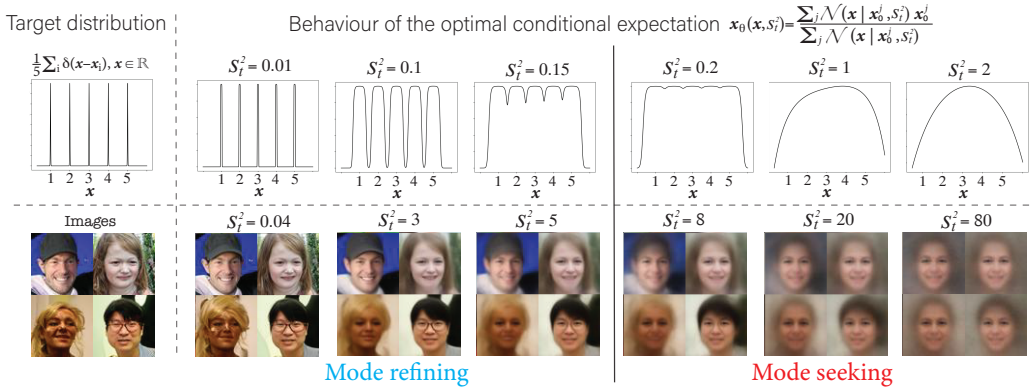

Figure 1: The behaviour of the optimal conditional expectation, $\mathbb{E}\left[\boldsymbol{x}_0|\boldsymbol{x}\right]$, for the toy distribution (top row) and images (bottom row). The behaviour leads to a rough border which separates two parts – mode refining and mode seeking . We argue that the mode seeking part **inevitably** requires significant sampler steps number, which blocks them from efficient sampling (1-10 steps).

$$\frac{-2}{N}\sum_{j=1}^{N}\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0^j, s_t^2\right)\left[\boldsymbol{x}_0^j - \boldsymbol{x}_\theta(\boldsymbol{x}, t)\right] = 0,$$
$$\boldsymbol{x}_\theta(\boldsymbol{x}, t) = \frac{\sum_{j=1}^{N}\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0^j, s_t^2\right)\boldsymbol{x}_0^j}{\sum_{j=1}^{N}\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0^j, s_t^2\right)}. \tag{48}$$

□

This result gives us the form of the optimal score function for the empirical distribution, $p_{\text{data}}(\cdot)$.

$$\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x}) = -\frac{1}{s_t^2}\left(\boldsymbol{x} - \frac{\sum_{j=1}^{N}\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0^j, s_t^2\right)\boldsymbol{x}_0^j}{\sum_{j=1}^{N}\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{x}_0^j, s_t^2\right)}\right). \tag{49}$$

To better understand its properties we can visualize the corresponding conditional expectation. Figure 1 demonstrates the behaviour of $\mathbb{E}\left[\boldsymbol{x}_0|\boldsymbol{x}\right]$ with respect to the different s_t (we denote it as σ for simplicity). In this example we consider two distributions:

- Toy, in which the target variable is one dimensional and the target distribution is the sum of five delta functions.
- Images, where the target variable is a sample from the FFHQ-64 dataset.

As it can be seen, there is a rough border which separates two parts. We call them as the mode refining and seeking. The reason is that the latter demonstrates common features of a entire distribution while the former deals with particular modes of this distribution.

We state that ODE/SDE samplers of the backward equation have to utilize significant number of steps to get through the mode seeking part. This fact blocks them from achieving high quality on small steps (1-10). In other words, samplers firstly need to "grope" objects in the mixture and then tune its fine-grained details. We believe that for efficient sampling the mode seeking part should be completed in a few steps (1-4). However, the optimal solution of the least square regression problem Eq. (26) does not allow us to make it possible.

D.2 Properties of the distilled diffusion models

Similar to diffusion models we can obtain the optimal form of distilled models.

Proposition 4. *The optimal solution to the consistency distillation problem Eq. (35) has the following form:*

$$\boldsymbol{x}_\theta(\boldsymbol{x}, t) = \boldsymbol{x}_0^k. \tag{50}$$

Proof. Assuming

$$p_{\text{data}}(\boldsymbol{x}_0) = \frac{1}{M}\sum_{j=1}^{M}\delta(\boldsymbol{x}_0 - \boldsymbol{x}_0^j). \tag{51}$$

Then,

$$\begin{aligned}\mathcal{L}^{\text{CD}}(\boldsymbol{x}_\theta) &= \mathbb{E}_{n\sim\mathcal{U}(2,N)}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_0)p_{t_n}(\boldsymbol{x}|\boldsymbol{x}_0)q^{\phi}_{t_{n-1}}(\hat{\boldsymbol{x}}|\boldsymbol{x})}\|\boldsymbol{x}_\theta(\hat{\boldsymbol{x}}, t_{n-1}) - \boldsymbol{x}_\theta(\boldsymbol{x}, t_n)\|_2^2 \\ &= \mathbb{E}_{n\sim\mathcal{U}(2,N)}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_0)p_{t_n}(\boldsymbol{x}|\boldsymbol{x}_0)}\int_{\mathbb{R}^n}\delta(\hat{\boldsymbol{x}} - \boldsymbol{x}_{t_{n-1}}(\boldsymbol{x}))\|\boldsymbol{x}_\theta(\hat{\boldsymbol{x}}, t_{n-1}) - \boldsymbol{x}_\theta(\boldsymbol{x}, t_n)\|_2^2\mathrm{d}\hat{\boldsymbol{x}},\end{aligned} \tag{52}$$

where we use $q^{\phi}_{t_{n-1}}(\hat{\boldsymbol{x}}|\boldsymbol{x}) = \delta(\hat{\boldsymbol{x}} - \boldsymbol{x}_{t_{n-1}}(\boldsymbol{x}))$ and note $\boldsymbol{x}_{t_{n-1}}(\boldsymbol{x})$ as one step of ODE solver with parameters ϕ, from t_n to t_{n-1}, starting from $\boldsymbol{x}$.

$$\begin{aligned}\mathcal{L}^{\text{CD}}(\boldsymbol{x}_\theta) &= \mathbb{E}_{n\sim\mathcal{U}(2,N)}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_0)p_{t_n}(\boldsymbol{x}|\boldsymbol{x}_0)}\|\boldsymbol{x}_\theta(\boldsymbol{x}_{t_{n-1}}(\boldsymbol{x}), t_{n-1}) - \boldsymbol{x}_\theta(\boldsymbol{x}, t_n)\|_2^2 \\ &= \mathbb{E}_{n\sim\mathcal{U}(2,N)}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}_0)}\int_{\mathbb{R}^n}\mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_0, s_{t_n}^2)\|\boldsymbol{x}_\theta(\boldsymbol{x}_{t_{n-1}}(\boldsymbol{x}), t_{n-1}) - \boldsymbol{x}_\theta(\boldsymbol{x}, t_n)\|_2^2\mathrm{d}\boldsymbol{x} \\ &= \frac{1}{N}\sum_{n=2}^{N}\frac{1}{M}\sum_{j=1}^{M}\int_{\mathbb{R}^n}\mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_0^j, s_{t_n}^2)\|\boldsymbol{x}_\theta(\boldsymbol{x}_{t_{n-1}}(\boldsymbol{x}), t_{n-1}) - \boldsymbol{x}_\theta(\boldsymbol{x}, t_n)\|_2^2\mathrm{d}\boldsymbol{x} \\ &= \frac{1}{N}\sum_{n=2}^{N}\int_{\mathbb{R}^n}\mathrm{d}\boldsymbol{x}\frac{1}{M}\sum_{j=1}^{M}\mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_0^j, s_{t_n}^2)\|\boldsymbol{x}_\theta(\boldsymbol{x}_{t_{n-1}}(\boldsymbol{x}), t_{n-1}) - \boldsymbol{x}_\theta(\boldsymbol{x}, t_n)\|_2^2 \\ &= \frac{1}{N}\sum_{n=2}^{N}\int_{\mathbb{R}^n}\mathrm{d}\boldsymbol{x}\mathcal{L}(\boldsymbol{x}_\theta, \boldsymbol{x}, t_n).\end{aligned} \tag{53}$$

$\mathcal{L}(\boldsymbol{x}_\theta, \boldsymbol{x}, t_n)$ can be optimized independently for given $\boldsymbol{x}$ and n.

$$\begin{aligned}\nabla_{\boldsymbol{x}_\theta}\mathcal{L}(\boldsymbol{x}_\theta, \boldsymbol{x}, t_n) &= \nabla_{\boldsymbol{x}_\theta}\left(\frac{1}{M}\sum_{j=1}^{M}\mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_0^j, s_{t_n}^2)\|\boldsymbol{x}_\theta(\boldsymbol{x}_{t_{n-1}}(\boldsymbol{x}), t_{n-1}) - \boldsymbol{x}_\theta(\boldsymbol{x}, t_n)\|_2^2\right) = 0, \\ \boldsymbol{x}_\theta(\boldsymbol{x}, t_n) &= \frac{\sum_{j=1}^{M}\mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_0^j, s_{t_n}^2)\boldsymbol{x}_\theta(\boldsymbol{x}_{t_{n-1}}(\boldsymbol{x}), t_{n-1})}{\sum_{j=1}^{M}\mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_0^j, s_{t_n}^2)}.\end{aligned} \tag{54}$$

Taking into account that $\boldsymbol{x} \sim p_{t_n}(\boldsymbol{x}|\boldsymbol{x}_0)$, we can assume $\boldsymbol{x} = \boldsymbol{x}_0^k + s_{t_n}\boldsymbol{z},\ \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{z}|0, 1)$. Then,

$$\boldsymbol{x}_\theta(\boldsymbol{x}, t_n) = \boldsymbol{x}_\theta(\boldsymbol{x}_{t_{n-1}}(\boldsymbol{x}), t_{n-1}). \tag{55}$$

Let $n = 2$.

$$\boldsymbol{x}_\theta(\boldsymbol{x}, t_2) = \boldsymbol{x}_\theta(\boldsymbol{x}_{t_1}(\boldsymbol{x}), t_1) \overset{(1)}{=} \boldsymbol{x}_{t_1}(\boldsymbol{x}) = \boldsymbol{x}_{t_1}(\boldsymbol{x}_0^k + s_{t_2}\boldsymbol{z}) \overset{(2)}{=} \boldsymbol{x}_0^k, \tag{56}$$

where (1) is due to parameterization of the consistency model and (2) because t_1 is the starting point ($s_{t_1} = 0$). □

As it can be seen, distilled models predict distribution modes directly without iterative denoising. This leads to the acceleration in the low-step regime. However, it is much easier for neural network to learn iterative denoising since this task is simpler compared to direct prediction. Thus, diffusion models demonstrate better quality of generated distribution.