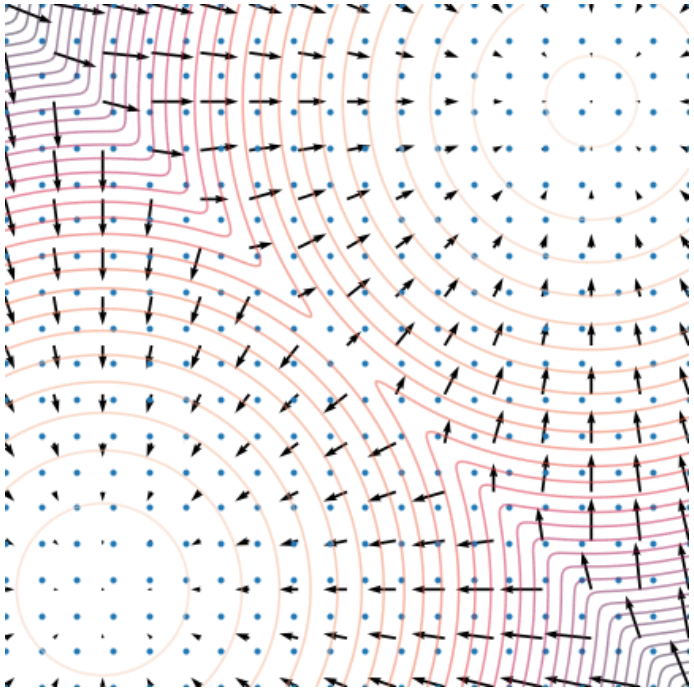
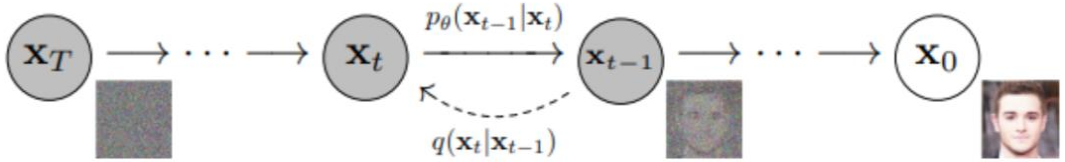
From DDPM to Score

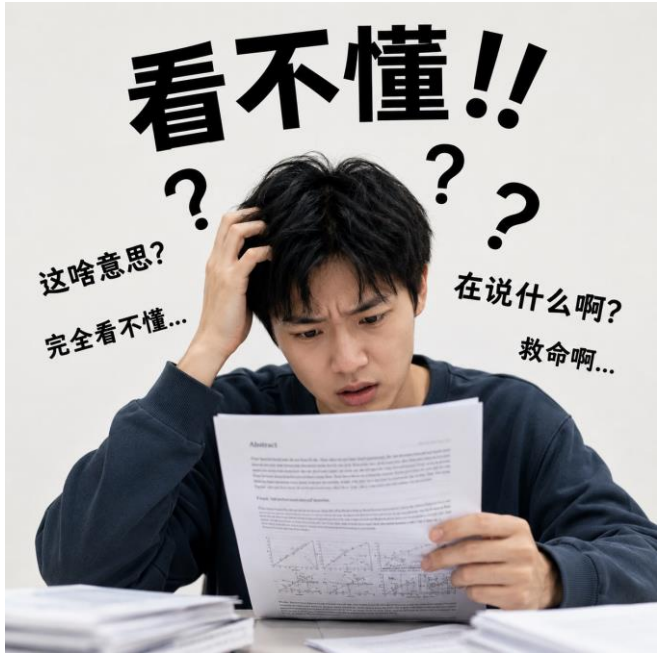


Score

Introduction

DDPM (2020)





step generative approach distilled from a pre-trained diffusion model [44]. The learning is achieved by enforcing a self-consistency in the predicted signal space. Based on this idea, following work [7, 11, 19, 41] have focus on improving the training techniques. However, learning consistency models for conditional generation has yet to be thoroughly studied. In this paper, we compare our method against a baseline approach that enforces self-consistency in an already fine-tuned conditional diffusion model. Our results demonstrate that our conditional distilled model outperforms the baseline approach, indicating the effectiveness of our proposed distillation strategy.

Diffusion Models Adaptations. Leveraging the knowledge of pre-trained models for new tasks, known as model adaptation, has gained significant traction in NLP and computer vision domains. This approach utilizes model adapters [9, 28, 30, 45] and HyperNetworks [1, 6] to effectively adapt pre-trained models to new domains and tasks. In the context of diffusion models, model adapters have been successfully employed to incorporate new conditions into pre-trained models [24, 53]. Our proposed method draws inspiration from these approaches and introduces a novel application of model adapters: distilling the sampling steps of diffusion models. Compared to fine-tuning the entire model [36], our method offers enhanced efficiency and flexibility. It enables the adaptation of multiple tasks using the same backbone model.

3. Background

Continuous-time VP diffusion model. A continuous-time variance-preserving (VP) diffusion model [8, 39] is a special case of diffusion models[1]. It has latent variables $\{\mathbf{z}_t|t \in [0,T]\}$ specified by a noise schedule comprising differentiable functions $\{\alpha_t, \sigma_t\}$ with $\sigma_t^2 = 1 - \alpha_t^2$. The clean data $\mathbf{x} \sim p_{\text{data}}$ is progressively perturbed in a (forward) Gaussian process as in the following Markovian structure:

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}), \quad (1)$$

$$q(\mathbf{z}_t|\mathbf{z}_s) = \mathcal{N}(\mathbf{z}_t; \alpha_{t|s}\mathbf{z}_s, \sigma_{t|s}^2 \mathbf{I}), \quad (2)$$

where $0 \le s < t \le 1$ and $\alpha_{t|s}^2 = \alpha_t/\alpha_s$. Here the latent $\mathbf{z}_t$ is sampled from the combination of the clean data and random noise by using the reparameterization trick [13], which has $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$.

Deterministic sampling. The aforementioned diffusion process that starts from $\mathbf{z}_0 \sim p_{\text{data}}(\mathbf{x})$ and ends at $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$ can be modeled as the solution of an stochastic differential equation (SDE) [43]. The SDE is formed by a vector-value function $f(\cdot,\cdot) : \mathbb{R}^d \to \mathbb{R}^d$, a scalar function $g(\cdot) : \mathbb{R} \to \mathbb{R}$, and the standard Wiener process $\mathbf{w}$ as:

$$\mathrm{d}\mathbf{z}_t = f(\mathbf{z}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}. \quad (3)$$

The overall idea is that the reverse-time SDE that runs backwards in time, can generate samples of p_{data} from the prior distribution $\mathcal{N}(0, \mathbf{I})$. This reverse SDE is given by

$$\mathrm{d}\mathbf{z}_t = [f(\mathbf{z}_t, t) - g(t)^2 \nabla_{\mathbf{z}} \log p_t(\mathbf{z}_t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}, \quad (4)$$

where the $\bar{\mathbf{w}}$ is a also standard Wiener process in reversed time, and $\nabla_{\mathbf{z}} \log p_t(\mathbf{z}_t)$ is the score of the marginal distribution at time t. The score function can be estimated by training a score-based model $s_\theta(\mathbf{z}_t, t) \approx \nabla_z \log p_t(\mathbf{z}_t)$ with score-matching [42] or a denoising network $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, t)$ [8]:

$$s_\theta(\mathbf{z}_t, t) := (\alpha_t \hat{\mathbf{x}}_\theta(\mathbf{z}_t, t) - \mathbf{z}_t)/\sigma_t^2. \quad (5)$$

Such backward SDE satisfies a special ordinary differential equation (ODE) that allows deterministic sampling given $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$. This is known as the *probability flow* (PF) ODE [43] and is given by

$$\mathrm{d}\mathbf{z}_t = [f(\mathbf{z}_t, t) - \frac{1}{2} g^2(t) s_\theta(\mathbf{z}_t, t)]\mathrm{d}t, \quad (6)$$

where $f(\mathbf{z}_t, t) = \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\mathbf{z}_t$, $g^2(t) = \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\sigma_t^2$ with respect to $\{\alpha_t, \sigma_t\}$ and t according to [12]. This ODE can be solved numerically with diffusion samplers like DDIM [40], where starting from $\hat{\mathbf{z}}_T \sim \mathcal{N}(0, \mathbf{I})$, we update for $s = t - \Delta t$:

$$\hat{\mathbf{z}}_s := \alpha_s \hat{\mathbf{x}}_\theta(\hat{\mathbf{z}}_t, t) + \sigma_s(\hat{\mathbf{z}}_t - \alpha_t \hat{\mathbf{x}}_\theta(\hat{\mathbf{z}}_t, t))/\sigma_t, \quad (7)$$

till we reach $\hat{\mathbf{z}}_0$.

Diffusion models parameterizations. Leaving aside the aforementioned way of parametrizing diffusion models with a denoising network (signal prediction) or a score model (noise prediction equation 5), in this work, we adopt a parameterization that mixes both the score (or noise) and the signal prediction. Existing methods include either predicting the noise $\hat{\epsilon}_\theta(\mathbf{x}_t, t)$ and the signal $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, t)$ separately using a single network [5], or predicting a combination of noise and signal by expressing them in a new term, like the velocity model $\hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) \approx \alpha_t \epsilon - \sigma_t \mathbf{x}$ [36]. Note that one can derive an estimation of the signal and the noise from the velocity one,

$$\hat{\mathbf{x}} = \alpha_t \mathbf{z}_t - \sigma_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t), \text{ and } \hat{\epsilon} = \alpha_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) + \sigma_t \mathbf{z}_t. \quad (8)$$

Similarly, DDIM update rule (equation 7) can be rewritten in terms of the velocity parametrization:

$$\hat{\mathbf{z}}_s := \alpha_s(\alpha_t \hat{\mathbf{z}}_t - \sigma_t \hat{\mathbf{v}}_\theta(\hat{\mathbf{z}}_t, t)) + \sigma_s(\alpha_t \hat{\mathbf{v}}_\theta(\hat{\mathbf{z}}_t, t) + \sigma_t \hat{\mathbf{z}}_t). \quad (9)$$

Self-consistency property. To accelerate inference, [44] introduced the idea of consistency models. Let $s_\theta(\cdot, t)$

[1] What we discussed based on the variance preserving (VP) form of SDE [43] is equivalent to most general diffusion models like Denoising Diffusion Probabilistic Models (DDPM) [8].

CoDi: Conditional Diffusion Distillation for Higher-Fidelity and Faster Image Generation, 2024

Score（2021, 2019）

$$p(x) = \frac{1}{Z} \exp\left(-\frac{E(x)}{kT}\right)$$

Boltzmann

Target

$$p_\theta(\mathbf{x})$$

In order to let

$$\begin{cases} \int p_\theta(\mathbf{x})\mathrm{d}\mathbf{x} = 1 \\ \boldsymbol{p_\theta > 0} \end{cases}$$

Define

$$\begin{cases} p_\theta(\mathbf{x}) = \dfrac{e^{-f_\theta(\mathbf{x})}}{Z_\theta} \\ Z_\theta = \int e^{-f_\theta(x)} dx > 0 \end{cases}$$

To get θ^* , introduce

$$L(\theta) = \prod_{i=1}^{N} p_\theta(x_i)$$

$$\ell(\theta) = \sum_{i=1}^{n} \log p_\theta(x_i)$$

$$\theta^* = \arg\max_\theta \sum_{i=1}^{n} \log p_\theta(x_i)$$

Write out

$$\log p_\theta(\mathbf{x}) = -f_\theta(\mathbf{x}) - \log Z_\theta$$

$$p_\theta(\mathbf{x}) = \frac{e^{-f_\theta(\mathbf{x})}}{Z_\theta}$$

$$Z_\theta = \int e^{-f_\theta(x)} dx > 0$$

Try

$$\theta^* = \arg\max_\theta \sum_{i=1}^{n} \log p_\theta(x_i)$$

$$\nabla_\theta \log p_\theta(x) = -\nabla_\theta f_\theta(x) + \mathbb{E}_{x' \sim p_\theta}[\nabla_\theta f_\theta(x')]$$

It's hard!

How about Langevin Dynamics?

$$\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \epsilon \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \sqrt{2\epsilon}\, \mathbf{z}_i, \quad i = 0, 1, \cdots, K \qquad z_i \sim \mathcal{N}(0, I) \quad \epsilon \in [10^{-4}, 10^{-2}]$$

It's lucky that

$$\nabla_x \log p_\theta(x) = \nabla_x[-f_\theta(x) - \log Z_\theta] = -\nabla_x f_\theta(x)$$

Design a "score-based model", and a "score function"

$$\boxed{s_\theta(x) \approx \nabla_x \log p_\theta(x)}$$

Optimize

$$\mathbb{E}_{p(\mathbf{x})}[\| \nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x}) \|_2^2]$$

Score-Matching(2005)

$$\mathcal{L} = \mathbb{E}_{p(\mathbf{x})}\left[\mathrm{tr}(\nabla_x \mathbf{s}_\theta(\mathbf{x})) + \frac{1}{2}\|\mathbf{s}_\theta(\mathbf{x})\|^2\right]$$

In practice:

$$\mathbb{E}_{p(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2]$$

$$\nabla_{x_t} \log p(x_t \mid x_0) = -\frac{\epsilon}{\sigma_t}$$

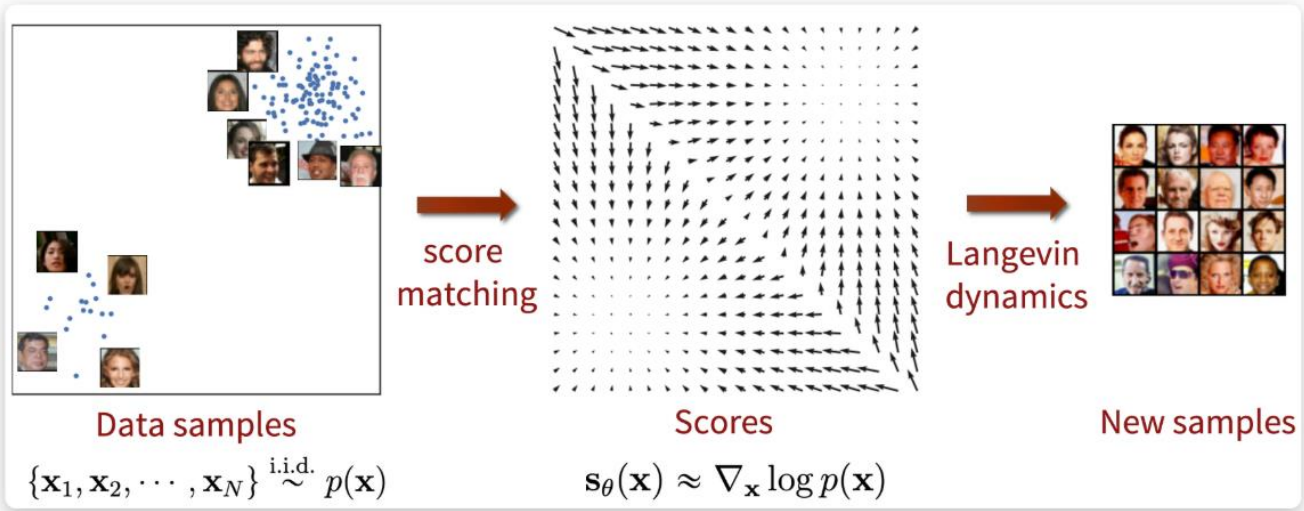
$$s_\theta(x) \approx \nabla_x \log p_\theta(x)$$

After we get $\mathbf{s}_\theta(\mathbf{x})$

Langevin Dynamics:

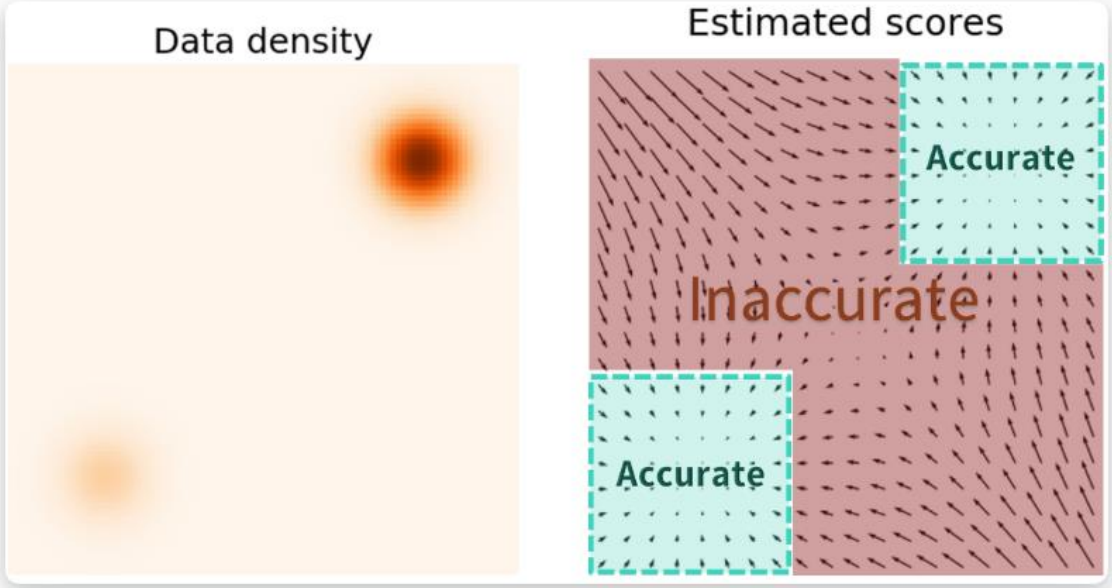
$$\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \epsilon \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \sqrt{2\epsilon}\,\mathbf{z}_i, \quad i = 0, 1, \cdots, K$$

$$\mathbf{x}_0, \mathbf{z}_i \sim \mathcal{N}(0, I)$$



But Performance suffers.

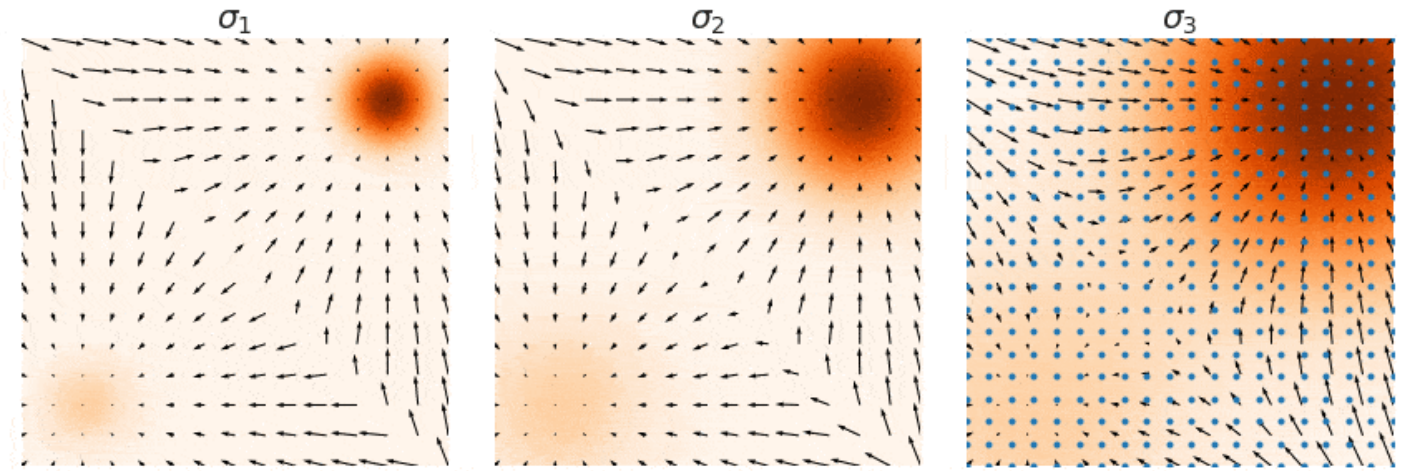
$$\mathbb{E}_{p(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2] = \int p(\mathbf{x}) \|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2 \mathrm{d}\mathbf{x}$$



Estimated scores are only accurate in high density regions.

Multi-scale perturbation

$$p_{\sigma_i}(\mathbf{x}) = \int p(\mathbf{y})\mathcal{N}(\mathbf{x};\mathbf{y},\sigma_i^2 I)\mathrm{d}\mathbf{y} \qquad \sigma_1 < \sigma_2 < \cdots < \sigma_L \qquad \mathbf{y} \sim p_{\text{data}}(\mathbf{y})$$



Inference

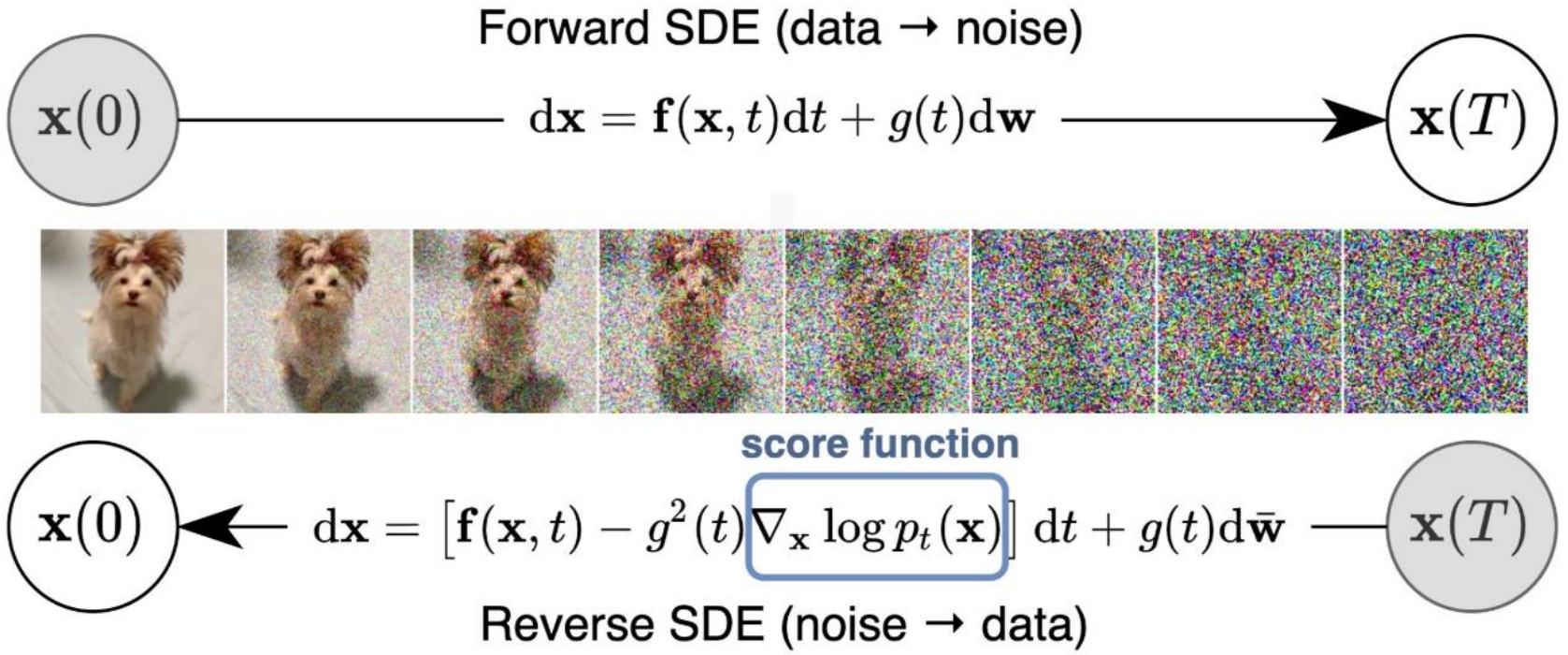
```
for i = L, L-1, ..., 1:
    for k = 1, ..., K:
        z ~ N(0, I)
        x ← x + ε_i ∇_x log p_{σ_i}(x) + √(2ε_i) z
return x
```

Score-Based Generative Modeling through SDEs

$$\mathbf{x}_{i,k+1} = \mathbf{x}_{i,k} + \epsilon_i \nabla_{\mathbf{x}} \log p_{\sigma_i}(\mathbf{x}_{i,k}) + \sqrt{2\epsilon_i}\, \mathbf{z}_{i,k}, \qquad k = 1, \ldots, K, \quad i = L, \ldots, 1$$

$$\sigma_1 < \sigma_2 < \cdots < \sigma_L$$

When L goes infinite, the Langevin process can be described as a SDE



step generative approach distilled from a pre-trained diffusion model [44]. The learning is achieved by enforcing a self-consistency in the predicted signal space. Based on this idea, following work [7, 11, 19, 41] have focus on improving the training techniques. However, learning consistency models for conditional generation has yet to be thoroughly studied. In this paper, we compare our method against a baseline approach that enforces self-consistency in an already fine-tuned conditional diffusion model. Our results demonstrate that our conditional distilled model outperforms the baseline approach, indicating the effectiveness of our proposed distillation strategy.

Diffusion Models Adaptations. Leveraging the knowledge of pre-trained models for new tasks, known as model adaptation, has gained significant traction in NLP and computer vision domains. This approach utilizes model adapters [9, 28, 30, 45] and HyperNetworks [1, 6] to effectively adapt pre-trained models to new domains and tasks. In the context of diffusion models, model adapters have been successfully employed to incorporate new conditions into pre-trained models [24, 53]. Our proposed method draws inspiration from these approaches and introduces a novel application of model adapters: distilling the sampling steps of diffusion models. Compared to fine-tuning the entire model [36], our method offers enhanced efficiency and flexibility. It enables the adaptation of multiple tasks using the same backbone model.

3. Background

Continuous-time VP diffusion model. A continuous-time variance-preserving (VP) diffusion model [8, 39] is a special case of diffusion models[1]. It has latent variables $\{\mathbf{z}_t | t \in [0, T]\}$ specified by a noise schedule comprising differentiable functions $\{\alpha_t, \sigma_t\}$ with $\sigma_t^2 = 1 - \alpha_t^2$. The clean data $\mathbf{x} \sim p_{\text{data}}$ is progressively perturbed in a (forward) Gaussian process as in the following Markovian structure:

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}), \tag{1}$$

$$q(\mathbf{z}_t|\mathbf{z}_s) = \mathcal{N}(\mathbf{z}_t; \alpha_{t|s} \mathbf{z}_s, \sigma_{t|s}^2 \mathbf{I}), \tag{2}$$

where $0 \leq s < t \leq 1$ and $\alpha_{t|s}^2 = \alpha_t / \alpha_s$. Here the latent $\mathbf{z}_t$ is sampled from the combination of the clean data and random noise by using the reparameterization trick [13], which has $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$.

Deterministic sampling. The aforementioned diffusion process that starts from $\mathbf{z}_0 \sim p_{\text{data}}(\mathbf{x})$ and ends at $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$ can be modeled as the solution of an stochastic differential equation (SDE) [43]. The SDE is formed by a vector-value function $f(\cdot, \cdot) : \mathbb{R}^d \to \mathbb{R}^d$, a scalar function $g(\cdot) : \mathbb{R} \to \mathbb{R}$, and the standard Wiener process $\mathbf{w}$ as:

$$\mathrm{d}\mathbf{z}_t = f(\mathbf{z}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}. \tag{3}$$

The overall idea is that the reverse-time SDE that runs backwards in time, can generate samples of p_{data} from the prior distribution $\mathcal{N}(0, \mathbf{I})$. This reverse SDE is given by

$$\mathrm{d}\mathbf{z}_t = [f(\mathbf{z}_t, t) - g(t)^2 \nabla_{\mathbf{z}} \log p_t(\mathbf{z}_t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}, \tag{4}$$

where the $\bar{\mathbf{w}}$ is a also standard Wiener process in reversed time, and $\nabla_{\mathbf{z}} \log p_t(\mathbf{z}_t)$ is the score of the marginal distribution at time t. The score function can be estimated by training a score-based model $s_\theta(\mathbf{z}_t, t) \approx \nabla_z \log p_t(\mathbf{z}_t)$ with score-matching [42] or a denoising network $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, t)$ [8]:

$$s_\theta(\mathbf{z}_t, t) := (\alpha_t \hat{\mathbf{x}}_\theta(\mathbf{z}_t, t) - \mathbf{z}_t)/\sigma_t^2. \tag{5}$$

Such backward SDE satisfies a special ordinary differential equation (ODE) that allows deterministic sampling given $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$. This is known as the *probability flow* (PF) ODE [43] and is given by

$$\mathrm{d}\mathbf{z}_t = [f(\mathbf{z}_t, t) - \frac{1}{2} g^2(t) s_\theta(\mathbf{z}_t, t)]\mathrm{d}t, \tag{6}$$

where $f(\mathbf{z}_t, t) = \frac{\mathrm{d} \log \alpha_t}{\mathrm{d}t} \mathbf{z}_t$, $g^2(t) = \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d} \log \alpha_t}{\mathrm{d}t}\sigma_t^2$ with respect to $\{\alpha_t, \sigma_t\}$ and t according to [12]. This ODE can be solved numerically with diffusion samplers like DDIM [40], where starting from $\hat{\mathbf{z}}_T \sim \mathcal{N}(0, \mathbf{I})$, we update for $s = t - \Delta t$:

$$\hat{\mathbf{z}}_s := \alpha_s \hat{\mathbf{x}}_\theta(\hat{\mathbf{z}}_t, t) + \sigma_s(\hat{\mathbf{z}}_t - \alpha_t \hat{\mathbf{x}}_\theta(\hat{\mathbf{z}}_t, t))/\sigma_t, \tag{7}$$

till we reach $\hat{\mathbf{z}}_0$.

Diffusion models parametrizations. Leaving aside the aforementioned way of parametrizing diffusion models with a denoising network (signal prediction) or a score model (noise prediction equation 5), in this work, we adopt a parameterization that mixes both the score (or noise) and the signal prediction. Existing methods include either predicting the noise $\hat{\epsilon}_\theta(\mathbf{x}_t, t)$ and the signal $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, t)$ separately using a single network [5], or predicting a combination of noise and signal by expressing them in a new term, like the velocity model $\hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) \approx \alpha_t \epsilon - \sigma_t \mathbf{x}$ [36]. Note that one can derive an estimation of the signal and the noise from the velocity one,

$$\hat{\mathbf{x}} = \alpha_t \mathbf{z}_t - \sigma_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t), \text{ and } \hat{\epsilon} = \alpha_t \hat{\mathbf{v}}_\theta(\mathbf{z}_t, t) + \sigma_t \mathbf{z}_t. \tag{8}$$

Similarly, DDIM update rule (equation 7) can be rewritten in terms of the velocity parametrization:

$$\hat{\mathbf{z}}_s := \alpha_s(\alpha_t \hat{\mathbf{z}}_t - \sigma_t \hat{\mathbf{v}}_\theta(\hat{\mathbf{z}}_t, t)) + \sigma_s(\alpha_t \hat{\mathbf{v}}_\theta(\hat{\mathbf{z}}_t, t) + \sigma_t \hat{\mathbf{z}}_t). \tag{9}$$

Self-consistency property. To accelerate inference, [44] introduced the idea of consistency models. Let $s_\theta(\cdot, t)$

[1] What we discussed based on the variance preserving (VP) form of SDE [43] is equivalent to most general diffusion models like Denoising Diffusion Probabilistic Models (DDPM) [8].

Thanks for your Listening!