Optimizing DDPM Sampling with Shortcut Fine-Tuning

Denoising Diffusion Probabilistic Models

Forward process:

$$q(x_{t+1}|x_t) = \mathcal{N}(x_{t+1}; \sqrt{1-\beta_{t+1}}x_t, \beta_{t+1}I)$$

Backward process:

$$q(x_t|x_{t+1}, x_0) = \mathcal{N}(x_t; \tilde{\mu}_{t+1}(x_{t+1}, x_0), \tilde{\beta}_{t+1}I),$$

with

$$\tilde{\mu}(x_{t+1}, x_0) = \frac{\sqrt{\tilde{\alpha}_t}\beta_t}{1-\tilde{\alpha}_{t+1}}x_0 + \frac{\sqrt{\alpha_{t+1}}(1-\tilde{\alpha}_t)}{1-\tilde{\alpha}_{t+1}}x_{t+1},$$

where $\alpha_{t+1} = 1 - \beta_{t+1}, \tilde{\alpha}_{t+1} = \prod_{s=1}^{t+1} \alpha_s$.

DDPM:

$$p_t^\theta(x_t|x_{t+1}) = \mathcal{N}(x_t; \mu_{t+1}^\theta(x_{t+1}), \Sigma_{t+1})$$

$$p_{0:T}^\theta = p(x_T) \prod_{t=0}^{T-1} p_t^\theta(x_t|x_{t+1}),$$

Score Matching Loss:

$$J = \mathbb{E}_q \left[\sum_{t=0}^{T-1} D_{KL}(q(x_t|x_{t+1}, x_0), p_t^\theta(x_t|x_{t+1}) \right]$$

Issues with DDPM Samplers

Case 1. Training DDPM with small T

- From Kwon et al. [2022], given a score matching loss J,

$$W_2\left(p_0^\theta, q_0\right) \leq \mathcal{O}\left(\sqrt{J}\right) + I(T)W_2(p_T, q_T),$$

  where W_2 is the Wasserstein-2 distance, $I(T)$ is nonexploding, and $W_2(p_T, q_T) \to 0$ as $T \to \infty$.
- In diffusion process, if T is small, then $p_T \not\approx q_T$, and $W_2(p_T, q_T)$ is not neglectable.

Case 2. Sampling with $T' << T$ subsampling steps

- According to Salimans and Ho [2022] and Xiao et al. [2022], a multistep Gaussian sampler cannot be distilled into a one-step Gaussian sampler without loss of fidelity.
- Existing methods can be viewed as imitation learning, which is suboptimal in many cases.

Main Question

Can we improve DDPM sampling by **not** *following the backward process?*

Integral Probability Metrics (IPM)

Define a critic $f_\alpha : \mathbb{R}^n \to \mathbb{R}$ for each $\alpha \in \mathcal{A}$, where $\mathcal{A}$ is a set of parameters. For a given critic f_α and distributions p_0^θ and q_0, define

$$g(p_0^\theta, f_\alpha, q_0) = \mathop{\mathbb{E}}_{x_0 \sim p_0^\theta} [f_\alpha(x_0)] - \mathop{\mathbb{E}}_{x_0 \sim q_0} [f_\alpha(x_0)] .$$

Suppose that

$$\forall \alpha \in \mathcal{A}, \exists \alpha' \in \mathcal{A} \text{ s.t. } f_\alpha = -f_{\alpha'}, \tag{1}$$

then

$$\Phi \left(p_0^\theta, q_0 \right) = \sup_{\alpha \in \mathcal{A}} g(p_0^\theta, f_\alpha, q_0)$$

is a pseudo-metrics between distributions, called integral probability metrics (IPM).

New Objective

Given a set of critic parameters $\mathcal{A}$, that satisfies (1), and a DDPM sampler with T step, and parameter θ, we want to solve

$$\min_{\theta} \Phi \left(p_0^{\theta}, q_0 \right),$$

or

$$\min_{\theta} \max_{\alpha \in \mathcal{A}} g \left(p_0^{\theta}, f_{\alpha}, q_0 \right) \tag{2}$$

New Objective

Let h_θ defines the stochastic sampling process of DDPM as follows:

$$h_{\theta,T}(x_T) = x_T \tag{3}$$

$$h_{\theta,t}(x_t) = \mu_\theta(h_{\theta,t+1}(x_{t+1})) + \epsilon_{t+1} \tag{4}$$

$$x_0 = h_{\theta,0}(x_T), \tag{5}$$

with $x_T \sim \mathcal{N}(0, I), \epsilon_{t+1} \sim \mathcal{N}(0, \Sigma_{t+1}), t = 0, \cdots, T-1$. Then the objective can be expressed as follows:

$$\Phi\left(p_0^\theta, q_0\right) = \sup_{\alpha \in \mathcal{A}} \left\{ \mathop{\mathbb{E}}_{x_T, \epsilon_{1:T}} [f_\alpha(h_{\theta,0}(x_T)] - \mathop{\mathbb{E}}_{x_0 \sim q_0} [f_\alpha(x_0)] \right\} \tag{6}$$

Shortcut Fine-Tuning (SFT)

Given p_0^θ, q_0, suppose that

$$\forall \theta, \exists \alpha \in \mathcal{A} \text{ s.t. } g(p_0^\theta, f_\alpha, q_0) = \Phi(p_0^\theta, q_0). \tag{7}$$

Let

$$\alpha^\star(p_0^\theta, q_0) \in \{\alpha \in \mathcal{A} | g(p_0^\theta, f_\alpha, q_0) = \Phi(p_0^\theta, q_0)\}. \tag{8}$$

Then if f_α is 1-Lipschitz, then we have

$$\nabla_\theta \Phi(p_0^\theta, q_0) = \nabla_\theta \mathop{\mathbb{E}}_{x_T, \epsilon_{1:T}} \left[f_{\alpha^\star(p_0^\theta, q_0)}\left(h_{\theta,0}(x_T)\right)\right] \tag{9}$$

$$= \mathop{\mathbb{E}}_{x_T, \epsilon_{1:T}} \left[\nabla_\theta f_{\alpha^\star(p_0^\theta, q_0)}\left(h_{\theta,0}(x_T)\right)\right]. \tag{10}$$

Proof.

Theorem 3 from Arjovsky et al. [2017]. □

Potential Issues with SFT

Since $h_{\theta,0}(x_T)$ is a composition of T functions, differentiating it have following potential issues:

- Gradient vanishing may cause the loss of long-distance dependency
- Gradient exploding
- High memory usage

Shortcut Fine-Tuning with Policy Gradient (SFT-PG)

Theorem 1 (Policy gradient equivalence)

If $p^{\theta}_{x_{0:T}}(x_{0:T}) f_{\alpha^\star(p^\theta_0, q_0)}(x_0)$ and $\nabla_\theta p^{\theta}_{x_{0:T}}(x_{0:T}) f_{\alpha^\star(p^\theta_0, q_0)}(x_0)$ are continuous w.r.t θ and $x_{0:T}$, then

$$\nabla_\theta \Phi(p^\theta_0, q_0) = \mathop{\mathbb{E}}_{p^\theta_{x_{0:T}}} \left[f_{\alpha^\star(p^\theta_0, q_0)}(x_0) \nabla_\theta \sum_{t=0}^{T-1} \log p^\theta_t(x_t | x_{t+1}) \right]. \quad (11)$$

Shortcut Fine-Tuning with Policy Gradient (SFT-PG)

Proof.

Let $V(\alpha, \theta) = \mathbb{E}_{x_0 \sim p_0^\theta}[f_\alpha(x_0)] - \mathbb{E}_{x_0 \sim q_0}[f_\alpha(x_0)]$, then by the envelope theorem, we have

$$
\begin{aligned}
\nabla_\theta \Phi(p_0^\theta, q_0) &= \nabla_\theta V(\alpha, \theta)\Big|_{\alpha=\alpha^\star(p_0^\theta, q_0)} \\
&= \nabla_\theta \left[\mathop{\mathbb{E}}_{x_0 \sim p_0^\theta}[f_\alpha(x_0)] - \mathop{\mathbb{E}}_{x_0 \sim q_0}[f_\alpha(x_0)] \Big|_{\alpha=\alpha^\star(p_0^\theta, q_0)} \right] \\
&= \nabla_\theta \left[\mathop{\mathbb{E}}_{x_0 \sim p_0^\theta}[f_\alpha(x_0)] \Big|_{\alpha=\alpha^\star(p_0^\theta, q_0)} \right] \\
&= \nabla_\theta \int p_0^\theta(x_0) f_\alpha(x_0) dx_0 \Big|_{\alpha=\alpha^\star(p_0^\theta, q_0)} \\
&= \nabla_\theta \int \int p_{0:T}^\theta(x_{0:T}) dx_{1:T} f_\alpha(x_0) dx_0 \Big|_{\alpha=\alpha^\star(p_0^\theta, q_0)} \\
&= \nabla_\theta \int p_{0:T}^\theta(x_{0:T}) f_\alpha(x_0) dx_{0:T} \Big|_{\alpha=\alpha^\star(p_0^\theta, q_0)}
\end{aligned}
$$

Shortcut Fine-Tuning with Policy Gradient (SFT-PG)

Proof (continued).

$$= \int \nabla_\theta p^\theta_{0:T}(x_{0:T}) f_\alpha(x_0) dx_{0:T} \Big|_{\alpha=\alpha^\star(p^\theta_0, q_0)} \tag{12}$$

$$= \int p^\theta_{0:T}(x_{0:T}) \nabla_\theta \log p^\theta_{0:T}(x_{0:T}) f_\alpha(x_0) dx_{0:T} \Big|_{\alpha=\alpha^\star(p^\theta_0, q_0)} \tag{13}$$

$$= \mathop{\mathbb{E}}_{p^\theta_{0:T}} \left[f_{\alpha^\star(p^\theta_0, q_0)}(x_0) \nabla_\theta \log \left(p_T(x_T) \prod_{t=0}^{T-1} p^\theta_t(x_t | x_{t+1}) \right) \right] \tag{14}$$

$$= \mathop{\mathbb{E}}_{p^\theta_{0:T}} \left[f_{\alpha^\star(p^\theta_0, q_0)}(x_0) \nabla_\theta \sum_{t=0}^{T-1} \log p^\theta_t(x_t | x_{t+1}) \right],$$

where (12) holds by continuity of $p^\theta_{x_{0:T}}(x_{0:T}) f_{\alpha^\star(p^\theta_0, q_0)}(x_0)$ and $\nabla_\theta p^\theta_{x_{0:T}}(x_{0:T}) f_{\alpha^\star(p^\theta_0, q_0)}(x_0)$, (13) holds by the log derivative trick, and (14) holds by the definition of $p^\theta_{0:T}$. □

Connection with Reinforcement Learning

Equation (11) can be viewed as a policy gradient of MDP with finite time horizon T, and

- $a_t = x_{t-1}$
- $\pi(a_t|s_t) = p_t^\theta(x_t|x_{t+1})$
- $R(s_t, a_t) = \begin{cases} f_{\alpha^\star(p_0^\theta, q_0)}(x_0) & t = 0 \\ 0 & \text{otherwise} \end{cases}$

Pros and Cons

Pro

- No gradient vanishing or exploding
- Not necessary to store intermediate gradients of a composite function

Con

- Stochastic policy gradient methods usually suffer from higher variance
  - Can use techniques in RL, such as baseline trick

Monotonic Improvement

Note that $\alpha^\star(p_0^\theta, q_0)$ depends on θ. Hence the gradient update might only be valid for *one step*.

Theorem 2 (The surrogate function of IPM)

Suppose that for given $\alpha \in \mathcal{A}$, and q_0, $g(p_0^\theta, f_\alpha, q_0)$. Then for a given critic $f_{\alpha^\star(p_0^\theta, q_0)}$, there exists $\ell \geq 0$ such that

$$\Phi(p_0^{\theta'}, q_0) \leq g(p_0^{\theta'}, f_{\alpha^\star(p_0^\theta, q_0)}, q_0) + 2\ell \left\|\theta - \theta'\right\|. \qquad (15)$$

Monotonic Improvement

Proof.

We drop 0 from p_0^θ and q_0 for convenience.

$$
\begin{aligned}
&\Phi(p^{\theta'}, q) - \Phi(p^\theta, q) \\
&= \int (p^{\theta'}(x) - q(x)) f_{\alpha^\star(p^{\theta'}, q)}(x) dx - \int (p^\theta(x) - q(x)) f_{\alpha^\star(p^\theta, q)}(x) dx \\
&= \int (p^{\theta'}(x) - q(x)) f_{\alpha^\star(p^{\theta'}, q)}(x) dx - \int (p^{\theta'}(x) - q(x)) f_{\alpha^\star(p^\theta, q)}(x) dx \\
&\quad + \int (p^{\theta'}(x) - q(x)) f_{\alpha^\star(p^\theta, q)}(x) dx - \int (p^\theta(x) - q(x)) f_{\alpha^\star(p^\theta, q)}(x) dx \\
&= \int (p^{\theta'}(x) - q(x)) \left(f_{\alpha^\star(p^{\theta'}, q)}(x) - f_{\alpha^\star(p^\theta, q)}(x) \right) dx + \int (p^{\theta'}(x) - p^\theta(x)) f_{\alpha^\star(p^\theta, q)}(x) dx
\end{aligned}
$$

Note

$$
\begin{aligned}
&\int (q(x) - p^{\theta'}(x)) \left(f_{\alpha^\star(p^\theta, q)}(x) - f_{\alpha^\star(p^{\theta'}, q)}(x) \right) dx \\
&= \int (p^\theta(x) - p^{\theta'}(x)) \left(f_{\alpha^\star(p^\theta, q)}(x) - f_{\alpha^\star(p^{\theta'}, q)}(x) \right) dx \\
&\quad - \int (p^\theta(x) - q(x)) \left(f_{\alpha^\star(p^\theta, q)}(x) - f_{\alpha^\star(p^{\theta'}, q)}(x) \right) dx \\
&\leq \int (p^\theta(x) - p^{\theta'}(x)) \left(f_{\alpha^\star(p^\theta, q)}(x) - f_{\alpha^\star(p^{\theta'}, q)}(x) \right) dx, \qquad (16)
\end{aligned}
$$

where inequality (16) holds by the definition of $\alpha^\star(p^\theta, q)$

Monotonic Improvement

Proof (continued).

Hence

$$
\begin{aligned}
& \Phi(p^{\theta'}, q) - \Phi(p^{\theta}, q) \\
\leq & \int (p^{\theta}(x) - p^{\theta'}(x)) \left(f_{\alpha^\star(p^{\theta}, q)}(x) - f_{\alpha^\star(p^{\theta'}, q)}(x) \right) dx + \int (p^{\theta'}(x) - p^{\theta}(x)) f_{\alpha^\star(p^{\theta}, q)}(x) dx \\
= & \left[g\left(p^{\theta}, f_{\alpha^\star(p^{\theta}, q)}, q\right) - g\left(p^{\theta'}, f_{\alpha^\star(p^{\theta}, q)}, q\right) \right] + \left[g\left(p^{\theta'}, f_{\alpha^\star(p^{\theta'}, q)}, q\right) - g\left(p^{\theta}, f_{\alpha^\star(p^{\theta'}, q)}, q\right) \right] \\
& + g\left(p^{\theta'}, f_{\alpha^\star(p^{\theta}, q)}, q\right) - g\left(p^{\theta}, f_{\alpha^\star(p^{\theta}, q)}, q\right) \\
\leq & g\left(p^{\theta'}, f_{\alpha^\star(p^{\theta}, q)}, q\right) - g\left(p^{\theta}, f_{\alpha^\star(p^{\theta}, q)}, q\right) + 2\ell \left\| \theta' - \theta \right\| \qquad (17) \\
= & g\left(p^{\theta'}, f_{\alpha^\star(p^{\theta}, q)}, q\right) - \Phi(p^{\theta}, q) + 2\ell \left\| \theta' - \theta \right\|,
\end{aligned}
$$

where inequality (17) holds by the Lipschitzness of g w.r.t θ for given $\alpha^\star(p^{\theta}, q)$ and $\alpha^\star(p^{\theta'}, q)$, respectively □

Monotonic Improvement

Theorem 2 implies that if θ' is sufficiently close to θ, then gradient descent of $g\left(p^{\theta'}, f_{\alpha^\star(p^\theta,q)}, q\right)$, guarantees the descent of $\Phi(p^{\theta'}, q)$. Hence using Lagrangian multiplier, we can convert the optimization problem into a constrained optimization problem

$$\min_{\theta'} g\left(p^{\theta'}, f_{\alpha^\star(p^\theta,q)}, q\right) \tag{18}$$

$$\text{s.t. } \left\|\theta' - \theta\right\| \leq \delta \tag{19}$$

for some $\delta > 0$.

Baseline Trick

Theorem 3 (Baseline trick)

If $p_t^\theta(x_t|x_{t+1})$ and $\nabla_\theta p_t^\theta(x_t|x_{t+1})$ are continuous, then

$$\begin{aligned} &\mathop{\mathbb{E}}_{p^\theta_{0:T}}\left[f_\alpha(x_0)\sum_{t=0}^{T-1}\nabla_\theta \log p_t^\theta(x_t|x_{t+1})\right] \\ =& \mathop{\mathbb{E}}_{p^\theta_{0:T}}\left[\left(f_\alpha(x_0) - V^\omega_{t+1}(x_{t+1})\right)\sum_{t=0}^{T-1}\nabla_\theta \log p_t^\theta(x_t|x_{t+1})\right] \end{aligned} \tag{20}$$

Baseline Trick

Proof.

It is suffice to show that

$$\mathop{\mathbb{E}}_{p_{0:T}}\left[V^{\omega}_{t+1}(x_{t+1})\nabla_\theta \log p^{\theta}_t(x_t|x_{t+1})\right] = 0.$$

Note that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{p_{0:T}}\left[V^{\omega}_{t+1}(x_{t+1})\nabla_\theta \log p^{\theta}_t(x_t|x_{t+1})\right] \\
&= \mathop{\mathbb{E}}_{p^{\theta}_{t+1:T}}\left[\mathop{\mathbb{E}}_{x_{0:t}{}^{\theta}}\left[V^{\omega}_{t+1}(x_{t+1})\nabla_\theta \log p^{\theta}_t(x_t|x_{t+1})|x_{t+1:T}\right]\right] \\
&= \mathop{\mathbb{E}}_{p^{\theta}_{t+1:T}}\left[\mathop{\mathbb{E}}_{p^{\theta}_t}\left[V^{\omega}_{t+1}(x_{t+1})\nabla_\theta \log p^{\theta}_t(x_t|x_{t+1})|x_{t+1:T}\right]\right].
\end{aligned}$$

Proof (continued).

But then since $p_t^\theta(x_t|x_{t+1})$ and $\nabla_\theta p_t^\theta(x_t|x_{t+1})$ are continuous,

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{p_t^\theta}\left[V_{t+1}^\omega(x_{t+1})\nabla_\theta \log p_t^\theta(x_t|x_{t+1})|x_{t+1:T}\right] \\
=&V_{t+1}^\omega(x_{t+1})\int p_t^\theta(x_t|x_{t+1})\nabla_\theta \log p_t^\theta(x_t|x_{t+1})dx_t \\
=&V_{t+1}^\omega(x_{t+1})\int \nabla_\theta p_t^\theta(x_t|x_{t+1})dx_t \qquad (21)\\
=&V_{t+1}^\omega(x_{t+1})\nabla_\theta \int p_t^\theta(x_t|x_{t+1})dx_t \\
=&0, \qquad (22)
\end{aligned}
$$

where equality (21) holds by the log derivative trick, and (22) holds by $\int p_t^\theta(x_t|x_{t+1})dx_t = 1$. □

Baseline Trick

With some mild assumptions, we have

$$
\begin{aligned}
Var =& \mathop{\mathbb{E}}_{p^{\theta}_{0:T}} \left[\left(\sum_{t=0}^{T-1} (f_\alpha(x_0) - V_{t+1}(x_{t+1})) \nabla_\theta p_t^\theta(x_t|x_{t+1}) \right)^2 \right] \\
& - \left(\mathop{\mathbb{E}}_{p^{\theta}_{0:T}} \left[\sum_{t=0}^{T-1} (f_\alpha(x_0) - V_{t+1}(x_{t+1})) \nabla_\theta p_t^\theta(x_t|x_{t+1}) \right] \right)^2 \\
\frac{\partial Var}{\partial V_{t+1}} =& \frac{\partial}{\partial V_{t+1}} \mathop{\mathbb{E}}_{p^{\theta}_{0:T}} \left[\left(\sum_{t=0}^{T-1} (f_\alpha(x_0) - V_{t+1}(x_{t+1})) \nabla_\theta p_t^\theta(x_t|x_{t+1}) \right)^2 \right] \\
=& \frac{\partial}{\partial V_{t+1}} \mathop{\mathbb{E}}_{p^{\theta}_{0:T}} \left[\left((f_\alpha(x_0) - V_{t+1}(x_{t+1})) \nabla_\theta p_t^\theta(x_t|x_{t+1}) \right)^2 \right] \\
=& \frac{\partial}{\partial V_{t+1}} \mathop{\mathbb{E}}_{p^{\theta}_{0:T}} \left[-2 f_\alpha(x_0) V_{t+1}(x_{t+1}) \left(\nabla_\theta p_t^\theta(x_t|x_{t+1}) \right)^2 \right] \\
& + \frac{\partial}{\partial V_{t+1}} \mathop{\mathbb{E}}_{p^{\theta}_{0:T}} \left[(V_{t+1}(x_{t+1}))^2 \left(\nabla_\theta p_t^\theta(x_t|x_{t+1}) \right)^2 \right] \\
=& -2 \mathop{\mathbb{E}}_{p^{\theta}_{0:T}} \left[f_\alpha(x_0) \left(\nabla_\theta p_t^\theta(x_t|x_{t+1}) \right)^2 \right] + 2 \mathop{\mathbb{E}}_{p^{\theta}_{0:T}} \left[V_{t+1}(x_{t+1}) \left(\nabla_\theta p_t^\theta(x_t|x_{t+1}) \right)^2 \right].
\end{aligned}
$$

Baseline Trick

Hence

$$\frac{\partial Var}{\partial V_{t+1}}(x_{t+1}) = -2 \underset{p^\theta_{0:T}}{\mathbb{E}} \left[f_\alpha(x_0) \left(\nabla_\theta p^\theta_t(x_t|x_{t+1}) \right)^2 |x_{t+1} \right] \tag{23}$$

$$+ 2V_{t+1}(x_{t+1}) \underset{p^\theta_{0:T}}{\mathbb{E}} \left[\left(\nabla_\theta p^\theta_t(x_t|x_{t+1}) \right)^2 |x_{t+1} \right] \tag{24}$$

and to set (23) to 0, we must have

$$V_{t+1}(x_{t+1}) = \frac{\mathbb{E}_{p^\theta_{0:T}} \left[f_\alpha(x_0) \left(\nabla_\theta p^\theta_t(x_t|x_{t+1}) \right)^2 |x_{t+1} \right]}{\mathbb{E}_{p^\theta_{0:T}} \left[\left(\nabla_\theta p^\theta_t(x_t|x_{t+1}) \right)^2 |x_{t+1} \right]} \tag{25}$$

However, in practice, one usually use

$$V_{t+1}(x_{t+1}, \alpha) = \underset{p^\theta_{0:T}}{\mathbb{E}} \left[f_\alpha(x_0)|x_{t+1} \right] \tag{26}$$

as a proxy.

Baseline Trick

To train V_{t+1}^{ω}, we use

$$R_B(\alpha, \omega, \theta) = \mathop{\mathbb{E}}_{p_{0:T}^{\theta}} \left[\sum_{t=0}^{T-1} \left(V_{t+1}^{\omega}(x_{t+1}) - V_{t+1}(x_{t+1}, \alpha) \right)^2 \right] \quad (27)$$

Regularizing Critic

Choice of $\mathcal{A}$ is important. Here are some examples of $\mathcal{A}$ and regularization techniques for the critic corresponding to such set of parameters.

Lipschitz regularization: $\mathcal{A} = \{\alpha | \, \|f_\alpha\|_L \leq 1\}$, then $f_{\alpha^\star(p_0^\theta, q_0)}$ satisfies

$$\left\| \nabla_{x_0} f_{\alpha^\star(p_0^\star, q_0)}(x_0) \right\| = 1 \tag{28}$$

almost everywhere.

Proof.

Corollary 1 of Gulrajani et al. [2017]. □

Hence to enforce Lipschitzness of f_α, we can use

$$R_{GP}(\alpha, \theta) = \mathop{\mathbb{E}}_{\hat{x}_0} \left[(\|\nabla_{x_0} f_\alpha(\hat{x}_0)\| - 1)^2 \right] \tag{29}$$

as a regularizer, where $\hat{x}_0$ is uniformly sampled from the line segment between $x_0' \sim p_0^\theta$ and $x_0'' \sim q_0$.

Regularizing Critic

Reusing baseline: Empirically reusing R_B as a regularizer was beneficial. Here is some intuition behind its benefit:

- V^{ω}_{t+1} can be viewed as a proxy of expected value of f_α from previous step
- R_B can be viewed as minimizing big change in expected valued of f_α, hence stabilize the training
- Also makes V^{ω}_{t+1} to fit better, because its loss is reused

Then to train the critic, the objective would be to maximize

$$L(\alpha, \omega, \theta) = g\left(p_0^\theta, f_\alpha, q_0\right) - \lambda R_B(\alpha, \omega, \theta) \tag{30}$$

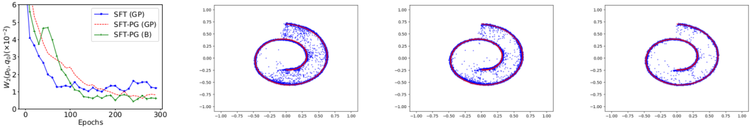
Algorithm

Algorithm 1 Shortcut Fine-Tuning with Policy Gradient and Baseline Regularization (SFT-PG (B))

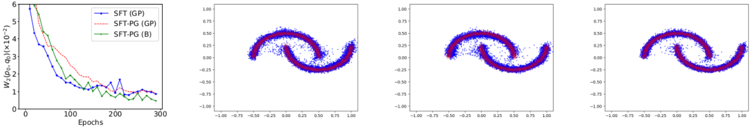
Inputs:
$n_{\text{critic}}, n_{\text{generator}}$, batchsize m, critic parameters α, baseline function parameters ω, pretrained generator θ, regularization hyperparameter λ

while θ not converge **do**
  Initialize buffer $\mathcal{B}$ as $\emptyset$
  for $i = 0, \cdots, n_{\text{critic}}$ **do**
    Obtain m i.i.d. samples from $p^{\theta}_{x_{0:T}}$
    Add all $\{x_{t+1}, x_t, x_0, t\}$ to $\mathcal{B}$
    Obtain m i.i.d. samples from q_0
    Update α and ω to maximize (30)
  end for
  for $j = 0, \cdots, n_{\text{generator}}$ **do**
    Obtain m samples of $\{x_{t+1}, x_t, x_0, t\}$ from $\mathcal{B}$
    Update θ by policy gradient using (20)
  end for
end while

Experiments



(a) Training curves of swiss roll (b) Roll, SFT (GP) (c) Roll, SFT-PG (GP) (d) Roll, SFT-PG (B)



(e) Training curves of moons (f) Moons, SFT (GP) (g) Moons, SFT-PG (GP) (h) Moons, SFT-PG (B)

Figure 1: Toy dataset experiments: swiss roll (top), two moons (bottom)

Experiments

Method	$W_2(p_0^\theta, q_0)$ ($\times 10^{-2}$)
$T = 10$, DDPM	8.29
$T = 100$, DDPM	2.36
$T = 1000$, DDPM	1.78
$T = 10$, SFT-PG (B)	**0.64**

Table 1: DDPM vs SFT on swiss roll dataset

Experiments

(a) CIFAR10, Initialization (b) CIFAR10, SFT-PG (B) (c) CelebA, Initialization (d) CelebA, SFT-PG (B)

Figure 2: Image dataset experiments: CIFAR-10 (a), (b) / CelebA (c), (d)

Experiments

Method	CIFAR-10 (FID)	CelebA (FID)
DDPM	34.76	36.69
FastDPM	29.43	28.98
Analytic-DPM	22.94	28.99
SN-DDPM	16.33	20.60
DDPM ($T = 1000$)	3.03	3.26
SFT-PG (B)	**2.28**	**2.01**

Table 2: FID on CIFAR-10 and CelebA for $T' = 10$

Thank You

Q & A

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 2017.

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein GANs. In *Advances in Neural Information Processing Systems*, 2017.

D. Kwon, Y. Fan, and K. Lee. Score-based generative modeling secretly minimizes the wasserstein distance. In *Advances in Neural Information Processing Systems*, 2022.

T. Salimans and J. Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Z. Xiao, K. Kreis, and A. Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations*, 2022.