Diffusion Neural Sampler: review, caveats and open questions

Jiajun He

Imperial College London

27/03/2025

Collaborators

Joint work with Yuanqi du;

collaborating with Francisco Vargas, Dinghuai Zhang, Shreyas Padhy, RuiKang OuYang;

supervised by Carla Gomes, José Miguel Hernández-Lobato






Sampling

Unnormalized density function:

$$p_{\text{target}}(x) = \frac{\tilde{p}(x)}{Z}, \qquad Z = \int \tilde{p}(x) \mathrm{d}x$$

Obtain sample $x \sim p_{\text{target}}$.

Sampling

Unnormalized density function:

$$p_{\text{target}}(x) = \frac{\tilde{p}(x)}{Z}, \qquad Z = \int \tilde{p}(x)\mathrm{d}x$$

Obtain sample $x \sim p_{\text{target}}$.

👉 Bayesian inference: $p_{\text{target}} \propto \text{likelihood} \times \text{prior}$

Sampling

Unnormalized density function:

$$p_{\text{target}}(x) = \frac{\tilde{p}(x)}{Z}, \qquad Z = \int \tilde{p}(x)\mathrm{d}x$$

Obtain sample $x \sim p_{\text{target}}$.

👉 Bayesian inference: $p_{\text{target}} \propto \text{likelihood} \times \text{prior}$

👉 Boltzmann distribution (molecules, etc): $p_{\text{target}} \propto \exp(-\beta U)$

Sampling – classical approach

Markov chain Monte Carlo (MCMC)

Sampling – classical approach

Markov chain Monte Carlo (MCMC)

For example, unadjusted Langevin dynamics:

$$\mathrm{d}X_t = \nabla \log \tilde{p}(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}W_t$$

Sampling – classical approach

Markov chain Monte Carlo (MCMC)

For example, unadjusted Langevin dynamics:

$$\mathrm{d}X_t = \underbrace{\underbrace{\nabla \log \tilde{p}(X_t)}_{\text{score}} \mathrm{d}t}_{\nabla \log \tilde{p}(X_t)\Delta t} + \underbrace{\sqrt{2}\mathrm{d}W_t}_{\sqrt{2\Delta t}\epsilon,\ \epsilon \sim N(0,1)}$$

Sampling – classical approach

Markov chain Monte Carlo (MCMC)

For example, unadjusted Langevin dynamics:

$$\mathrm{d}X_t = \nabla \log \tilde{p}(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}W_t$$

😔 dependent samples; auto-correlation reduces efficiency sample size

Sampling – classical approach

Markov chain Monte Carlo (MCMC)

For example, unadjusted Langevin dynamics:

$$\mathrm{d}X_t = \nabla \log \tilde{p}(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}W_t$$

😔 dependent samples; auto-correlation reduces efficiency sample size

😔 ergodicity; only guarantee convergence with infinite steps

Neural samplers

Train a neural network to amortize the sampling process

Neural samplers

Train a neural network to amortize the sampling process

- 🙂 independent samples!
- 🙂 can mix in finite time

Neural samplers

Train a neural network to amortize the sampling process

🥹 independent samples!

🥹 can mix in finite time

Neural samplers are in fact generative models:

$$p_{\text{prior}} \longrightarrow p_{\text{target}}$$

generative model
NF, Diffusion, etc.

Diffusion Neural samplers

Train a diffusion (like) model

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,$$

Diffusion Neural samplers

Train a diffusion (like) model

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,$$

transporting samples from p_{prior} to p_{target}:

$$X_0 \sim p_{\text{prior}} \text{ , and want } X_T \sim p_{\text{target}}$$

Diffusion Neural samplers - idea 1

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

Diffusion Neural samplers - idea 1

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

If we have a “target” process

$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}$,

Diffusion Neural samplers - idea 1

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

If we have a “target” process

$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}$,

And $X_t \sim Y_{T-t}$,

Diffusion Neural samplers - idea 1

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

If we have a "target" process

$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}$,

And $X_t \sim Y_{T-t}$,

We will have $X_T \sim Y_{T-T} = Y_0$

Diffusion Neural samplers - idea 1

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

If we have a "target" process

$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}$,

And $X_t \sim Y_{T-t}$, "time-reversal"

We will have $X_T \sim Y_{T-T} = Y_0$

Diffusion Neural samplers - idea 1

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

If we have a "target" process

$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}$,

can be a simple, prescribed function, like 0 or $-\beta_t Y_t$

And $X_t \sim Y_{T-t}$, "time-reversal"

We will have $X_T \sim Y_{T-T} = Y_0$

Diffusion Neural samplers - idea 1

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

If we have a "target" process

$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}$,

can be a simple, prescribed function, like 0 or $-\beta_t Y_t$

And $X_t \sim Y_{T-t}$, "time-reversal"

We will have $X_T \sim Y_{T-T} = Y_0$

Diffusion Neural samplers - idea 1

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}}$, we want $X_T \sim p_{\text{target}}$.

Want a sample process (prior to target),

If we have a “target” process

$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$

To be the **time-reversal,**

g can be a simple, prescribed function, like 0 or $-\beta_t Y_t$

And $X_t \sim Y_{T-t}$, “time-reversal”

of a simple target process (target to prior)

We will have $X_T \sim Y_{T-T} = Y_0$

Diffusion Neural samplers - idea 1

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

Want a sample process (prior to target),

If we have a “target” process

$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t$ ...

To be the **time-reversal,**

can be a simple, prescribed function, like 0 or $-\beta_t Y_t$

And $X_t \sim Y_{T-t}$, “time-reversal”

of a simple target process (target to prior)

We will have $X_T \sim Y_{T-}$...

How to achieve this?

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad X_0 \sim p_{\mathrm{prior}}$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2 \Delta t\right), \qquad X_0 \sim p_{\mathrm{prior}}$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

$$Y_{t_n} \sim N\left(Y_{t_n} | Y_{t_{n-1}} + g\left(Y_{t_{n-1}}, t\right)\Delta t, 2\sigma^2 \Delta t\right), \qquad Y_0 \sim p_{\mathrm{target}}$$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad X_0 \sim p_{\mathrm{prior}}$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

$$Y_{t_n} \sim N\left(Y_{t_n} | Y_{t_{n-1}} + g\left(Y_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad Y_0 \sim p_{\mathrm{target}}$$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad X_0 \sim p_{\mathrm{prior}}$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

$$Y_{t_n} \sim N\left(Y_{t_n} | Y_{t_{n-1}} + g\left(Y_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad Y_0 \sim p_{\mathrm{target}}$$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad X_0 \sim p_{\text{prior}}$$

$$p_{\text{prior}}(X_0) N\left(X_{t_1} \middle| X_0\right) N\left(X_{t_2} \middle| X_{t_1}\right) \ldots N(X_{t_N} | X_{t_{N-1}})$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

$$Y_{t_n} \sim N\left(Y_{t_n} | Y_{t_{n-1}} + g\left(Y_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad Y_0 \sim p_{\text{target}}$$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad X_0 \sim p_{\text{prior}}$$

$$p_{\text{prior}}(X_0)N\left(X_{t_1} \middle| X_0\right)N\left(X_{t_2} \middle| X_{t_1}\right) \dots N(X_{t_N} | X_{t_{N-1}})$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

$$Y_{t_n} \sim N\left(Y_{t_n} | Y_{t_{n-1}} + g\left(Y_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad Y_0 \sim p_{\text{target}}$$

$$p_{\text{target}}(Y_0)N\left(Y_{t_1} \middle| Y_0\right)N\left(Y_{t_2} \middle| Y_{t_1}\right) \dots N(Y_{t_N} | Y_{t_{N-1}})$$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad X_0 \sim p_{\text{prior}}$$

$$p_{\text{prior}}(X_0) N\left(X_{t_1} \middle| X_0\right) N\left(X_{t_2} \middle| X_{t_1}\right) \ldots N(X_{t_N} | X_{t_{N-1}})$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

$$Y_{t_n} \sim N\left(Y_{t_n} | Y_{t_{n-1}} + g\left(Y_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad Y_0 \sim p_{\text{target}}$$

$$p_{\text{target}}(Y_0) N\left(Y_{t_1} \middle| Y_0\right) N\left(Y_{t_2} \middle| Y_{t_1}\right) \ldots N(Y_{t_N} | Y_{t_{N-1}})$$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2 \Delta t\right), \qquad X_0 \sim p_{\text{prior}}$$

$$p_{\text{prior}}(X_0) N\left(X_{t_1} \middle| X_0\right) N\left(X_{t_2} \middle| X_{t_1}\right) \dots N(X_{t_N} | X_{t_{N-1}})$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

$$Y_{t_n} \sim N\left(Y_{t_n} | Y_{t_{n-1}} + g\left(Y_{t_{n-1}}, t\right)\Delta t, 2\sigma^2 \Delta t\right), \qquad Y_0 \sim p_{\text{target}}$$

$$Y_t \sim X_{T-t}$$

$$p_{\text{target}}(Y_0) N\left(Y_{t_1} \middle| Y_0\right) N\left(Y_{t_2} \middle| Y_{t_1}\right) \dots N(Y_{t_N} | Y_{t_{N-1}})$$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad X_0 \sim p_{\text{prior}}$$

$$p_{\text{prior}}(X_0) N\left(X_{t_1} | X_0\right) N\left(X_{t_2} | X_{t_1}\right) \dots N\left(X_{t_N} | X_{t_{N-1}}\right)$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

$$Y_{t_n} \sim N\left(Y_{t_n} | Y_{t_{n-1}} + g\left(Y_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad Y_0 \sim p_{\text{target}}$$

$$Y_t \sim X_{T-t}$$

$$p_{\text{target}}(\cancel{Y_0}) N\left(\cancel{Y_{t_1}} | \cancel{Y_0}\right) N\left(Y_{t_2} | Y_{t_1}\right) \dots N\left(Y_{t_N} | Y_{t_{N-1}}\right)$$

X_{t_N} $\quad X_{t_{N-1}}$ $\quad \dots \quad \dots \quad \dots \quad \dots$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad X_0 \sim p_{\text{prior}}$$

$$p_{\text{prior}}(X_0) N\left(X_{t_1} \middle| X_0\right) N\left(X_{t_2} \middle| X_{t_1}\right) \dots N(X_{t_N} | X_{t_{N-1}})$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

$$Y_{t_n} \sim N\left(Y_{t_n} | Y_{t_{n-1}} + g\left(Y_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad Y_0 \sim p_{\text{target}}$$

$$p_{\text{target}}(X_{t_N}) N\left(X_{t_{N-1}} \middle| X_{t_N}\right) N\left(X_{t_{N-2}} \middle| X_{t_{N-1}}\right) \dots N(X_{t_0} | X_{t_1})$$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2 \Delta t\right), \qquad X_0 \sim p_{\mathrm{prior}}$$

$$p_{\mathrm{prior}}(X_0) N\left(X_{t_1} \middle| X_0\right) N\left(X_{t_2} \middle| X_{t_1}\right) \dots N(X_{t_N} | X_{t_{N-1}}) \qquad \coloneqq q(X_{0:t_N})$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

$$Y_{t_n} \sim N\left(Y_{t_n} | Y_{t_{n-1}} + g\left(Y_{t_{n-1}}, t\right)\Delta t, 2\sigma^2 \Delta t\right), \qquad Y_0 \sim p_{\mathrm{target}}$$

$$p_{\mathrm{target}}(X_{t_N}) N\left(X_{t_{N-1}} \middle| X_{t_N}\right) N\left(X_{t_{N-2}} \middle| X_{t_{N-1}}\right) \dots N(X_{t_0} | X_{t_1}) \qquad \coloneqq p(X_{0:t_N})$$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad X_0 \sim p_{\mathrm{prior}}$$

$$p_{\mathrm{prior}}(X_0)N\left(X_{t_1} | X_0\right)N\left(X_{t_2} | X_{t_1}\right) \dots N(X_{t_N} | X_{t_{N-1}}) \quad := q(X_{0:t_N})$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

$$Y_{t_n} \sim N\left(Y_{t_n} | Y_{t_{n-1}} + g\left(Y_{t_{n-1}}, t\right)\Delta t, 2\sigma^2\Delta t\right), \qquad Y_0 \sim p_{\mathrm{target}}$$

$$p_{\mathrm{target}}(X_{t_N})N\left(X_{t_{N-1}} | X_{t_N}\right)N\left(X_{t_{N-2}} | X_{t_{N-1}}\right) \dots N(X_{t_0} | X_{t_1}) \quad := p(X_{0:t_N})$$

Diffusion Neural samplers - idea 1.1

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

$$X_{t_n} \sim N\left(X_{t_n} | X_{t_{n-1}} + f_\theta\left(X_{t_{n-1}}, t\right)\Delta t, 2\sigma^2 \Delta t\right), \qquad X_0 \sim p_{\text{prior}}$$

$$p_{\text{prior}}(X_0) N\left(X_{t_1} \middle| X_0\right) N\left(X_{t_2} \middle| X_{t_1}\right) \dots N(X_{t_N} | X_{t_{N-1}}) \qquad := q(X_{0:t_N})$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

$$Y_{t_n} \sim N\left(Y_{t_n} | Y_{t_{n-1}} + g\left(Y_{t_{n-1}}, t\right)\Delta t, 2\sigma^2 \Delta t\right), \qquad Y_0 \sim p_{\text{target}}$$

$$\tilde{p}_{\text{target}}(X_{t_N}) N\left(X_{t_{N-1}} \middle| X_{t_N}\right) N\left(X_{t_{N-2}} \middle| X_{t_{N-1}}\right) \dots N(X_{t_0} | X_{t_1}) \qquad := \tilde{p}(X_{0:t_N})$$

Diffusion Neural samplers - idea 1.1

Match $q(X_{0:t_N})$ with $\tilde{p}(X_{0:t_N})$:

Diffusion Neural samplers - idea 1.1

Match $q\left(X_{0:t_N}\right)$ with $\tilde{p}\left(X_{0:t_N}\right)$:

$$D_{\mathrm{KL}}[q\left(X_{0:t_N}\right)||\tilde{p}\left(X_{0:t_N}\right)] = \mathrm{E}_q\left[\log\frac{q\left(X_{0:t_N}\right)}{\tilde{p}\left(X_{0:t_N}\right)}\right]$$

Diffusion Neural samplers - idea 1.1

Match $q(X_{0:t_N})$ with $\tilde{p}(X_{0:t_N})$:

$$D_{\mathrm{KL}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{E}_q\left[\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})}\right]$$

$$D_{\mathrm{LV}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{Var}_\pi\left[\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})}\right]$$

Diffusion Neural samplers - idea 1.1

Match $q(X_{0:t_N})$ with $\tilde{p}(X_{0:t_N})$:

$$D_{\mathrm{KL}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{E}_q\left[\log \frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})}\right]$$

$$D_{\mathrm{LV}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{Var}_{\pi}\left[\log \frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})}\right]$$

It is fine to have a different sampling process

Diffusion Neural samplers - idea 1.1

Match $q(X_{0:t_N})$ with $\tilde{p}(X_{0:t_N})$:

$$D_{\mathrm{KL}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{E}_q\left[\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})}\right]$$

$$D_{\mathrm{LV}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{Var}_\pi\left[\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})}\right]$$

$$D_{\mathrm{TB}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{E}_\pi\left[\left(\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})} - k\right)^2\right]$$

Diffusion Neural samplers - idea 1.1

Match $q(X_{0:t_N})$ with $\tilde{p}(X_{0:t_N})$:

$$D_{\mathrm{KL}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{E}_q\left[\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})}\right]$$

$$D_{\mathrm{LV}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{Var}_\pi\left[\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})}\right]$$

$$D_{\mathrm{TB}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{E}_\pi\left[\left(\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})} - k\right)^2\right]$$

Other choices exist, including sub-TB, DB, etc…

Diffusion Neural samplers - idea 1.1

Match $q\left(X_{0:t_N}\right)$ with $\tilde{p}\left(X_{0:t_N}\right)$:

Let's go continuous!

$$D_{\mathrm{KL}}[q\left(X_{0:t_N}\right)||\tilde{p}\left(X_{0:t_N}\right)] = \mathrm{E}_q\left[\log\frac{q\left(X_{0:t_N}\right)}{\tilde{p}\left(X_{0:t_N}\right)}\right]$$

$$D_{\mathrm{LV}}[q\left(X_{0:t_N}\right)||\tilde{p}\left(X_{0:t_N}\right)] = \mathrm{Var}_\pi\left[\log\frac{q\left(X_{0:t_N}\right)}{\tilde{p}\left(X_{0:t_N}\right)}\right]$$

$$D_{\mathrm{TB}}[q\left(X_{0:t_N}\right)||\tilde{p}\left(X_{0:t_N}\right)] = \mathrm{E}_\pi\left[\left(\log\frac{q\left(X_{0:t_N}\right)}{\tilde{p}\left(X_{0:t_N}\right)} - k\right)^2\right]$$

Other choices exist, including sub-TB, DB, etc…

Diffusion Neural samplers - idea 1.1

Match $q(X_{0:t_N})$ with $\tilde{p}(X_{0:t_N})$: ➡ $\overrightarrow{\mathbf{Q}}(X), \overleftarrow{\mathbf{P}}(X)$

$$D_{\mathrm{KL}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{E}_q\left[\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})}\right]$$

$$D_{\mathrm{LV}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{Var}_\pi\left[\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})}\right]$$

$$D_{\mathrm{TB}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{E}_\pi\left[\left(\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})} - k\right)^2\right]$$

Other choices exist, including sub-TB, DB, etc…

Diffusion Neural samplers - idea 1.1

Match $q\left(X_{0:t_N}\right)$ with $\tilde{p}\left(X_{0:t_N}\right)$: ➡ $\vec{\mathbf{Q}}(X), \overleftarrow{\mathbf{P}}(X)$

$$D_{\mathrm{KL}}[\vec{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{E}_{\vec{\mathbf{Q}}}\left[\log \frac{\mathrm{d}\vec{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{LV}}[\vec{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{Var}_{\vec{\boldsymbol{\pi}}}\left[\log \frac{\mathrm{d}\vec{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{TB}}[\vec{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{E}_{\vec{\boldsymbol{\pi}}}\left[\left(\log \frac{\mathrm{d}\vec{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)} - k\right)^2\right]$$

Other choices exist, including sub-TB, DB, etc…

Diffusion Neural samplers - idea 1.1

Match $q\left(X_{0:t_N}\right)$ with $\tilde{p}\left(X_{0:t_N}\right)$: → $\vec{\mathbf{Q}}(X), \overleftarrow{\mathbf{P}}(X)$

$$\log \frac{\mathrm{d}\vec{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}?$$

$$D_{\mathrm{KL}}[\vec{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{E}_{\vec{\mathbf{Q}}}\left[\log \frac{\mathrm{d}\vec{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{LV}}[\vec{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{Var}_{\vec{\pi}}\left[\log \frac{\mathrm{d}\vec{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{TB}}[\vec{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{E}_{\vec{\pi}}\left[\left(\log \frac{\mathrm{d}\vec{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)} - k\right)^2\right]$$

Other choices exist, including sub-TB, DB, etc...

Diffusion Neural samplers - idea 1.1

Match $q(X_{0:t_N})$ with $\tilde{p}(X_{0:t_N})$: → $\vec{\mathbf{Q}}(X), \overleftarrow{\mathbf{P}}(X)$

$$\log \frac{\mathrm{d}\vec{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}?$$

$$D_{\mathrm{KL}}[\vec{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathbb{E}_{\vec{\mathbf{Q}}}\left[\log \frac{\mathrm{d}\vec{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{LV}}[\vec{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{Var}_{\vec{\mathbf{Q}}}\left[\log \frac{\mathrm{d}\vec{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

We can calculate this by Girsanov theorem
when two paths are **in the same direction**

$$D_{\mathrm{TB}}[\vec{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathbb{E}_{\vec{\pi}}\left[\left(\log \frac{\mathrm{d}\vec{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)} - k\right)^2\right]$$

Other choices exist, including sub-TB, DB, etc…

Diffusion Neural samplers - idea 1.1

Match $q\left(X_{0:t_N}\right)$ with $\tilde{p}\left(X_{0:t_N}\right)$: ➡ $\overrightarrow{\mathbf{Q}}(X), \overleftarrow{\mathbf{P}}(X)$

$$\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)} ?$$

$$= \log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overrightarrow{\mathbf{P}_r}(X)} + \log \frac{\mathrm{d}\overrightarrow{\mathbf{P}_r}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$$

$$D_{\mathrm{KL}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{E}_{\overrightarrow{\mathbf{Q}}}\left[\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{LV}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{Var}_{\overrightarrow{\pi}}\left[\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{TB}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{E}_{\overrightarrow{\pi}}\left[\left(\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)} - k\right)^2\right]$$

Other choices exist, including sub-TB, DB, etc...

Diffusion Neural samplers - idea 1.1

Match $q(X_{0:t_N})$ with $\tilde{p}(X_{0:t_N})$: → $\overrightarrow{\mathbf{Q}}(X), \overleftarrow{\mathbf{P}}(X)$

$$\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}?$$

$$= \log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overrightarrow{\mathbf{P}}_r(X)} + \log \frac{\mathrm{d}\overrightarrow{\mathbf{P}}_r(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$$

$$= \log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overrightarrow{\mathbf{P}}_r(X)} + \log \frac{\mathrm{d}\overleftarrow{\mathbf{P}}_r(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$$

Other choices exist, including sub-TB, DB, etc...

Diffusion Neural samplers - idea 1.1

Match $q\left(X_{0:t_N}\right)$ with $\tilde{p}\left(X_{0:t_N}\right)$: ➡ $\overrightarrow{\mathbf{Q}}(X), \overleftarrow{\mathbf{P}}(X)$

$$\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)} ?$$

We can choose any $\mathbf{P}_r$

$$= \log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overrightarrow{\mathbf{P}_r}(X)} + \log \frac{\mathrm{d}\overrightarrow{\mathbf{P}_r}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$$

$$= \log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overrightarrow{\mathbf{P}_r}(X)} + \log \frac{\mathrm{d}\overleftarrow{\mathbf{P}_r}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$$

Other choices exist, including sub-TB, DB, etc...

Diffusion Neural samplers - idea 1.1

Match $q(X_{0:t_N})$ with $\tilde{p}(X_{0:t_N})$: $\Rightarrow$ $\overrightarrow{\mathbf{Q}}(X), \overleftarrow{\mathbf{P}}(X)$

$$\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)} ?$$

We can choose any $\mathbf{P}_r$

$$= \log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overrightarrow{\mathbf{P}}_r(X)} + \log \frac{\mathrm{d}\overrightarrow{\mathbf{P}}_r(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$$

Choose it to have known $\overrightarrow{\mathbf{P}}_r$ and $\overleftarrow{\mathbf{P}}_r$

$$= \log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overrightarrow{\mathbf{P}}_r(X)} + \log \frac{\mathrm{d}\overleftarrow{\mathbf{P}}_r(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$$

Other choices exist, including sub-TB, DB, etc...

Diffusion Neural samplers - idea 1.1

Want a sample process (prior to target),

To be the **time-reversal,**

of a simple target process (target to prior)

How to achieve this?

matching forward and backward processes

Diffusion Neural samplers - idea 1.2

Want a sample process (prior to target),

To be the **time-reversal,**

of a simple target process (target to prior)

Any other choices to achieve this? YES!

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

For simplicity, we consider $g = 0$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Recall in diffusion models,

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Recall in diffusion models, we learn

$$f_\theta(X_t, t) = 2\sigma^2 \nabla \log p_{T-t}(X_t)$$

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Recall in diffusion models, we learn

$$f_\theta(X_t, t) = 2\sigma^2 \nabla \log p_{T-t}(X_t)$$

What is this term?

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Recall in diffusion models, we learn

$$f_\theta(X_t, t) = 2\sigma^2 \nabla \log p_{T-t}(X_t)$$

What is this term?

The "score" at $T - t$

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Recall in diffusion models, we learn

$$f_\theta(X_t, t) = 2\sigma^2 \nabla \log p_{T-t}(X_t)$$

What is this term?

The "score" at $T - t$

Recall $X_t \sim Y_{T-t}$

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Recall in diffusion models, we learn

$$f_\theta(X_t, t) = 2\sigma^2 \nabla \log p_{T-t}(X_t)$$

What is this term?

The "score" at $T - t$

Recall $X_t \sim Y_{T-t}$

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

The "score" at t

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}$$

At time t, $p_t(Y_t) = \int p_{\mathrm{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0$

Diffusion Neural samplers - idea 1.2

$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$

At time t, $p_t(Y_t) = \int p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I)\mathrm{d}Y_0$

We want to have a network to regress its score

Diffusion Neural samplers - idea 1.2

$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}$

At time t, $p_t(Y_t) = \int p_{\mathrm{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0$

We want to have a network to regress its score

With data $Y_0 \sim p_{\mathrm{target}}$: **denoising score matching**

Diffusion Neural samplers - idea 1.2

$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$

At time t, $p_t(Y_t) = \int p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0$

We want to have a network to regress its score

With data $Y_0 \sim p_{\text{target}}$: **denoising score matching**

What if without data?

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$$

$$\nabla \log p_t(Y_t) = \nabla \log \overbrace{\int p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0}^{\text{Gaussian convolution}}$$

Diffusion Neural samplers - idea 1.2

$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$

$$\nabla \log p_t(Y_t) = \nabla \log \overbrace{\int p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0}^{\text{Gaussian convolution}}$$

$$= \nabla \left(p_{\text{target}} * N(\cdot\, | 0, v_t I)\right)(Y_t) / \, p_t(Y_t)$$

Diffusion Neural samplers - idea 1.2

$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$

Gaussian convolution

$$\nabla \log p_t(Y_t) = \nabla \log \overbrace{\int p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0}$$

$$= \nabla \left(p_{\text{target}} * N(\cdot\, | 0, v_t I)\right)(Y_t) / \, p_t(Y_t)$$

Gradient of Conv = Conv of gradient

$$= \left(\nabla p_{\text{target}} * N(\cdot\, | 0, v_t I)\right)(Y_t) / \, p_t(Y_t)$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$$

Gaussian convolution

$$\nabla \log p_t(Y_t) = \nabla \log \int p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0$$

$$= \nabla \left(p_{\text{target}} * N(\cdot\,|0, v_t I)\right)(Y_t) / p_t(Y_t)$$

Gradient of Conv = Conv of gradient

$$= \left(\nabla p_{\text{target}} * N(\cdot\,|0, v_t I)\right)(Y_t) / p_t(Y_t)$$

$$= \int \nabla p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0 \,/ p_t(Y_t)$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}$$

$$\nabla \log p_t(Y_t)$$

$$= \int \nabla p_{\mathrm{target}}(Y_0) N(Y_t | Y_0, v_t I)\mathrm{d}Y_0 \; / p_t(Y_t)$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$$

$$\nabla \log p_t(Y_t)$$

$$= \int \nabla p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I)\mathrm{d}Y_0 \; / p_t(Y_t)$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$$

$$\nabla \log p_t(Y_t)$$

$$= \int \boxed{\nabla p_{\text{target}}(Y_0)} N(Y_t | Y_0, v_t I) \mathrm{d}Y_0 \; / p_t(Y_t)$$

$$\boxed{p_{\text{target}}(Y_0) \nabla \log p_{\text{target}}(Y_0)}$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}$$

$$\nabla \log p_t(Y_t)$$

$$= \int \nabla p_{\mathrm{target}}(Y_0) N(Y_t | Y_0, v_t I)\mathrm{d}Y_0 \; / p_t(Y_t)$$

$$= \int p_{\mathrm{target}}(Y_0) \nabla \log p_{\mathrm{target}}(Y_0) N(Y_t | Y_0, v_t I)\mathrm{d}Y_0 \; / p_t(Y_t)$$

Diffusion Neural samplers - idea 1.2

$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}$

$$\nabla \log p_t(Y_t)$$

$$= \int \nabla p_{\mathrm{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0 \,/ p_t(Y_t)$$

$$= \int p_{\mathrm{target}}(Y_0) \nabla \log p_{\mathrm{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0 \,/ p_t(Y_t)$$

Diffusion Neural samplers - idea 1.2

$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$

$$\nabla \log p_t(Y_t)$$

$$= \int \nabla p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0 \; / p_t(Y_t)$$

$$= \int p_{\text{target}}(Y_0) \nabla \log p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0 \; / p_t(Y_t)$$

$$= \int p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I) \; / p_t(Y_t) \nabla \log p_{\text{target}}(Y_0) \mathrm{d}Y_0$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$$

$$\nabla \log p_t(Y_t)$$

$$= \int \nabla p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0 \; / p_t(Y_t)$$

$$= \int p_{\text{target}}(Y_0) \nabla \log p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0 \; / p_t(Y_t)$$

$$= \int \boxed{p_{\text{target}}(Y_0) N(Y_t | Y_0, v_t I) \; / p_t(Y_t)} \nabla \log p_{\text{target}}(Y_0) \mathrm{d}Y_0$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}$$

$$\nabla \log p_t(Y_t)$$

$$= \int \nabla p_{\mathrm{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0 \; / p_t(Y_t)$$

$$= \int p_{\mathrm{target}}(Y_0) \nabla \log p_{\mathrm{target}}(Y_0) N(Y_t | Y_0, v_t I) \mathrm{d}Y_0 \; / p_t(Y_t)$$

$$= \int \boxed{p_{\mathrm{target}}(Y_0) N(Y_t | Y_0, v_t I) \; / p_t(Y_t)} \nabla \log p_{\mathrm{target}}(Y_0) \mathrm{d}Y_0$$

Bayes' Rule! $\boxed{p(Y_0 | Y_t)}$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$$

$$\nabla\log p_t(Y_t)$$

$$= \int \nabla p_{\text{target}}(Y_0) N(Y_t|Y_0, v_t I)\mathrm{d}Y_0 \;/p_t(Y_t)$$

$$= \int p_{\text{target}}(Y_0)\nabla\log p_{\text{target}}(Y_0) N(Y_t|Y_0, v_t I)\mathrm{d}Y_0 \;/p_t(Y_t)$$

$$= \int p_{\text{target}}(Y_0) N(Y_t|Y_0, v_t I) \;/p_t(Y_t)\nabla\log p_{\text{target}}(Y_0)\mathrm{d}Y_0$$

$$= \int p(Y_0|Y_t)\nabla\log p_{\text{target}}(Y_0)\mathrm{d}Y_0$$

Diffusion Neural samplers - idea 1.2

$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$

$$\nabla \log p_t(Y_t) = \int p(Y_0|Y_t) \nabla \log p_{\text{target}}(Y_0) \mathrm{d}Y_0$$

Target score identity (TSI)

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$$

$$\nabla \log p_t(Y_t) = \int p(Y_0|Y_t)\nabla \log p_{\text{target}}\,(Y_0)\mathrm{d}Y_0$$

Target score identity (TSI)

But we still do not know how to sample from $p(Y_0|Y_t)$

Diffusion Neural samplers - idea 1.2

$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$

$$\nabla \log p_t(Y_t) = \int p(Y_0|Y_t) \nabla \log p_{\text{target}}\,(Y_0)\mathrm{d}Y_0$$

Target score identity (TSI)

But we still do not know how to sample from $p(Y_0|Y_t)$

$$\nabla \log p_t(Y_t) = \int q(Y_0|Y_t) \frac{p(Y_0|Y_t)}{q(Y_0|Y_t)} \nabla \log p_{\text{target}}\,(Y_0)\mathrm{d}Y_0$$

Diffusion Neural samplers - idea 1.2

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$$

$$\nabla \log p_t(Y_t) = \int p(Y_0|Y_t) \nabla \log p_{\text{target}}\,(Y_0)\mathrm{d}Y_0$$

Target score identity (TSI)

But we still do not know how to sample from $p(Y_0|Y_t)$

$$\nabla \log p_t(Y_t) = \int q(Y_0|Y_t) \frac{p(Y_0|Y_t)}{q(Y_0|Y_t)} \nabla \log p_{\text{target}}\,(Y_0)\mathrm{d}Y_0$$

Importance Sampling using q

Diffusion Neural samplers - idea 1.2

$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}$

Want a sample process (prior to target),

$$\nabla \log p_t(Y_t) = \int p(Y_0|Y_t) \nabla \log p_{\text{target}}(Y_0) \mathrm{d}Y_0$$

To be the **time-reversal,**

Target score identity

of a simple target process (target to prior)

But we still do not know how to sample from $p(Y_0|Y_t)$

$$\nabla \log p_t(Y_t) = \int q(Y_0|Y_t) \frac{p(Y_0|Y_t)}{q(Y_0|Y_t)} \nabla \log p_{\text{target}}(Y_0) \mathrm{d}Y_0$$

Estimate score by TSI+IS, and regress it with a score net

Importance Sampling using q

Diffusion Neural samplers - idea 1.3

$$dY_t = \sigma\sqrt{2}dW_t, Y_0 \sim p_{\text{target}}$$

Want a sample process (prior to target),

$$\nabla \log p_t(Y_t) = \int p(Y_0|Y_t) \nabla \log p_{\text{target}}(Y_0) dY_0$$

To be the **time-reversal,**

Target score identity

of a simple target process (target to prior)

But we still do not know how to sample from $p(Y_0|Y_t)$

Any other choices to achieve this? YEEEES!

$$\nabla \log p_t(Y_t) = \int q(Y_0|Y_t) \frac{p(Y_0|Y_t)}{q(Y_0|Y_t)} \nabla \log p_{\text{target}}(Y_0) dY_0$$

Importance Sampling using q

Diffusion Neural samplers - idea 1.3

$$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Recall in diffusion models, we learn

$$f_\theta(X_t, t) = 2\sigma^2 \nabla \log p_{T-t}(X_t)$$

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

Diffusion Neural samplers - idea 1.3

$$\mathrm{d}X_t = 2\sigma^2 \nabla \log p_{T-t}(X_t)\,\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

$$\mathrm{d}Y_t = \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}},$$

Diffusion Neural samplers - idea 1.3

$$\mathrm{d}X_t = 2\sigma^2 \nabla \log p_{T-t}(X_t)\,\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Diffusion Neural samplers - idea 1.3

$$\mathrm{d}X_t = 2\sigma^2 \nabla \log p_{T-t}(X_t)\,\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

We want the marginal density of this SDE at $T - t$, to be $p_{T-t}(X_t)$

Diffusion Neural samplers - idea 1.3

$$\mathrm{d}X_t = 2\sigma^2 \nabla \log p_{T-t}(X_t)\,\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

We want the marginal density of this SDE at $T - t$, to be $p_{T-t}(X_t)$

What connects an SDE with its marginal density?

Diffusion Neural samplers - idea 1.3

$$\mathrm{d}X_t = 2\sigma^2 \nabla \log p_{T-t}(X_t)\,\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

We want the marginal density of this SDE at $T - t$, to be $p_{T-t}(X_t)$

What connects an SDE with its marginal density?

Fokker-Planck equation!

Diffusion Neural samplers - idea 1.3

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Fokker-Planck equation (in log space)

$$\partial_t \log p_t + \nabla \cdot f + \nabla \log p_t \cdot f - \sigma^2 \left|\left|\nabla \log p_t\right|\right|^2 - \sigma^2 \Delta \log p_t = 0$$

Diffusion Neural samplers - idea 1.3

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Fokker-Planck equation (in log space)

$$\partial_t \log p_t + \nabla \cdot f + \nabla \log p_t \cdot f - \sigma^2 \left|\left|\nabla \log p_t\right|\right|^2 - \sigma^2 \Delta \log p_t = 0$$

Do not worry on this formula

Let's focus on the high-level idea

Diffusion Neural samplers - idea 1.3

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Fokker-Planck equation (in log space)

$$\partial_t \log p_t + \nabla \cdot f + \nabla \log p_t \cdot f - \sigma^2 ||\nabla \log p_t||^2 - \sigma^2 \Delta \log p_t = 0$$

f **only contains** σ **and score of marginal:** $\nabla \log p_t$

Diffusion Neural samplers - idea 1.3

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Fokker-Planck equation (in log space)

$$\partial_t \log p_t + \nabla \cdot f + \nabla \log p_t \cdot f - \sigma^2 \left|\left|\nabla \log p_t\right|\right|^2 - \sigma^2 \Delta \log p_t = 0$$

LFS will have only one unknown term $\log p_t$

Diffusion Neural samplers - idea 1.3

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Fokker-Planck equation (in log space)

$$\partial_t \log p_t + \nabla\cdot f + \nabla \log p_t \cdot f - \sigma^2 ||\nabla \log p_t||^2 - \sigma^2 \Delta \log p_t = 0$$

LFS will have only one unknown term $\log p_t$

We can parameter network for $\log p_t$, and learn it by $\min ||\mathrm{LFS}||^2$

Diffusion Neural samplers - idea 1.3

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Want a sample process (prior to target),

Fokker-Planck equation (in log space)

To be the **time-reversal,**

$$\partial_t \log p_t + \nabla\cdot f + \nabla \log p_t \cdot f - \sigma^2 \left|\left|\nabla \log p_t\right|\right|^2 - \sigma^2 \Delta \log p_t = 0$$

of a simple target process (target to prior)

LFS will have only one unknown term $\log p_t$

We can parameter network for $\log p_t$, and learn it by $\min\left|\left|\mathrm{LFS}\right|\right|^2$

matching the PDE induced by SDE

Diffusion Neural samplers - idea 1

Want a sample process (prior to target),

To be the **time-reversal,**

of a simple target process (target to prior)

1.1 align forward with backward

1.2 align the marginal to the desired marginal by

1.2.1 score matching

1.2.2 satisfy PDE

Diffusion Neural samplers - idea 1

This includes

(1) DDS (denoising diffusion sampler)

(2) PIS (path integral sampler)

(3) DIS (diffusion time-reversal sampler)

(4) GFlowNet (generative flow network)

aligning forward with backward

(5) iDEM (iterated denoising energy matching)

(6) RDMC (reversal diffusion monte carlo)

score matching/estimation with IS

(7) PINN (physics-informed neural networks) sampler

satisfying PDE

…

Diffusion Neural samplers - idea 2

Diffusion Neural samplers - idea 2

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

Diffusion Neural samplers - idea 2

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

We can define a sequence of interpolants π_t :

$$\pi_0 = p_{\mathrm{prior}}, \pi_T = p_{\mathrm{target}}$$

Diffusion Neural samplers - idea 2

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

We can define a sequence of interpolants π_t :

$$\pi_0 = p_{\mathrm{prior}}, \pi_T = p_{\mathrm{target}}$$

We want the marginal of X_t to be π_t.

Diffusion Neural samplers - idea 2

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

We can define a sequence of interpolants π_t :

$$\pi_0 = p_{\mathrm{prior}}, \pi_T = p_{\mathrm{target}}$$

We want the marginal of X_t to be π_t.

One example for π_t: $\pi_t \propto p_{\mathrm{prior}}^{\beta_t} p_{\mathrm{target}}^{1-\beta_t}$

Diffusion Neural samplers - idea 2

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}}$, we want $X_T \sim p_{\text{target}}$.

We can define a sequence of interpolants π_t :

$$\pi_0 = p_{\text{prior}}, \pi_T = p_{\text{target}}$$

We want the marginal of X_t to be π_t.

Diffusion Neural samplers - idea 2

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}}$, we want $X_T \sim p_{\text{target}}$.

Want a sample process (prior to target),

We can define a sequence of interpolants π_t :

whose marginal density at every time step,

$$\pi_0 = p_{\text{prior}}, \pi_T = p_{\text{target}}$$

aligns with known interpolants between prior and target

We want the marginal of X_t to be π_t.

Diffusion Neural samplers - idea 2

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

Want a sample process (prior to target),

We can define a sequence of interpolants π_t :

whose marginal density at every time step,

$$\pi_0 = p_{\mathrm{prior}}, \pi_T = p_{\mathrm{target}}$$

aligns with known interpolants between prior and target

We want the marginal of X_t to be π_t.

How to achieve this?

Diffusion Neural samplers - idea 2

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}}$, we want $X_T \sim p_{\text{target}}$.

Want a sample process (prior to target),

We can define a sequence of interpolants π_t :

whose marginal density at every time step,

$$\pi_0 = p_{\text{prior}}, \pi_T = p_{\text{target}}$$

aligns with known interpolants between prior and target

We want the marginal of X_t to be π_t.

How to achieve this?

Satisfy the PDE!

Diffusion Neural samplers - idea 2.1

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Fokker-Planck equation (in log space)

$$\partial_t \log p_t + \nabla \cdot f + \nabla \log p_t \cdot f - \sigma^2 \left|\left|\nabla \log p_t\right|\right|^2 - \sigma^2 \Delta \log p_t = 0$$

Diffusion Neural samplers - idea 2.1

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Fokker-Planck equation (in log space)

$$\partial_t \cancel{\log p_t} + \nabla \cdot f + \nabla \cancel{\log p_t} \cdot f - \sigma^2 \left|\left|\nabla \cancel{\log p_t}\right|\right|^2 - \sigma^2 \Delta \cancel{\log p_t} = 0$$

$\log \pi_t$ $\log \pi_t$ $\log \pi_t$ $\log \pi_t$

Diffusion Neural samplers - idea 2.1

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Fokker-Planck equation (in log space)

$$\partial_t \log p_t + \nabla \cdot f + \nabla \log p_t \cdot f - \sigma^2 \left|\left|\nabla \log p_t\right|\right|^2 - \sigma^2 \Delta \log p_t = 0$$

$\log \pi_t$ $\log \pi_t$ $\log \pi_t$ $\log \pi_t$

For example, $\pi_t = p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t} / Z_{\pi_t}$

Diffusion Neural samplers - idea 2.1

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Fokker-Planck equation (in log space)

$$\partial_t \log p_t + \nabla \cdot f + \nabla \log p_t \cdot f - \sigma^2 \left|\left|\nabla \log p_t\right|\right|^2 - \sigma^2 \Delta \log p_t = 0$$

$\partial_t \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t} - \partial_t Z_{\pi_t}$ $\quad \log \pi_t \quad \log \pi_t \quad \log \pi_t$

For example, $\pi_t = p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t} / Z_{\pi_t}$

Diffusion Neural samplers - idea 2.1

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Fokker-Planck equation (in log space)

$$\cancel{\partial_t \log p_t} + \nabla \cdot f + \cancel{\nabla \log p_t} \cdot f - \sigma^2 \left|\left|\cancel{\nabla \log p_t}\right|\right|^2 - \sigma^2 \Delta \cancel{\log p_t} = 0$$

$\partial_t \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t} - \partial_t Z_{\pi_t}$ $\quad \nabla \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t}$ $\quad \log \pi_t$

For example, $\pi_t = p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t} / Z_{\pi_t}$

Diffusion Neural samplers - idea 2.1

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Fokker-Planck equation (in log space)

$$\partial_t \log p_t + \nabla \cdot f + \nabla \log p_t \cdot f - \sigma^2 \left|\left|\nabla \log p_t\right|\right|^2 - \sigma^2 \Delta \log p_t = 0$$

$\partial_t \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t} - \partial_t Z_{\pi_t}$ $\nabla \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t}$ $\mathrm{tr}\left(\nabla\nabla \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t}\right)$

For example, $\pi_t = p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t} / Z_{\pi_t}$

Diffusion Neural samplers - idea 2.1

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Fokker-Planck equation (in log space)

$$\cancel{\partial_t \log p_t} + \nabla \cdot f + \cancel{\nabla \log p_t} \cdot f - \sigma^2 \left|\left|\cancel{\nabla \log p_t}\right|\right|^2 - \sigma^2 \cancel{\Delta \log p_t} = 0$$

$\partial_t \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t} - \partial_t Z_{\pi_t}$ $\quad \nabla \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t}$ $\quad \mathrm{tr}\left(\nabla\nabla \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t}\right)$

Again, do not worry on this formula

Let's focus on the high-level idea

Diffusion Neural samplers - idea 2.1

$$\mathrm{d}X_t = f(X_t)\mathrm{dt} + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Fokker-Planck equation (in log space)

$$\partial_t \log p_t + \nabla \cdot f + \nabla \log p_t \cdot f - \sigma^2 \left|\left|\nabla \log p_t\right|\right|^2 - \sigma^2 \Delta \log p_t = 0$$

$\partial_t \log p_{\mathrm{prior}}^{\beta_t} p_{\mathrm{target}}^{1-\beta_t} - \partial_t Z_{\pi_t}$ $\quad\nabla \log p_{\mathrm{prior}}^{\beta_t} p_{\mathrm{target}}^{1-\beta_t}$ $\quad\mathrm{tr}\left(\nabla\nabla \log p_{\mathrm{prior}}^{\beta_t} p_{\mathrm{target}}^{1-\beta_t}\right)$

The LHS only has **2 unknown terms**: scalar func $Z_{\pi_t}(t)$ and vector func $f(X,t)$

We can parameter network for $Z_{\pi_t}(t), f(X,t)$, and learn it by $\min\left|\left|\mathrm{LFS}\right|\right|^2$

Diffusion Neural samplers - idea 2.1

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Want a sample process (prior to target),

Fokker-Planck equation (in log space)

whose marginal density at every time step,

$$\partial_t \log p_t + \nabla \cdot f + \nabla \log p_t \cdot f - \sigma^2 ||\nabla \log p_t||^2 - \sigma^2 \Delta \log p_t = 0$$

$\partial_t \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t} - \partial_t Z_{\pi_t}$ $\nabla \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t}$ $\left(\nabla \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t}\right)$

aligns with known interpolants between prior and target

How to achieve this?

The LHS only has **2 unknown terms**: scalar func $Z_{\pi_t}(t)$ and vector func $f(X,t)$

Satisfy the PDE!

We can parameter network for $Z_\theta(t), f_\theta(X,t)$, and learn it by $\min ||\text{LFS}||^2$

Diffusion Neural samplers - idea 2.1

$$\mathrm{d}X_t = f(X_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Want a sample process (prior to target),

Fokker-Planck equation (in log space)

whose marginal density at every time step,

$$\partial_t \log p_t + \nabla \cdot f + \nabla \log p_t \cdot f - \sigma^2 \left|\left|\nabla \log p_t\right|\right|^2 - \sigma^2 \Delta \log p_t = 0$$

$$\partial_t \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t} - \partial_t Z_{\pi_t} \quad \nabla \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t} \quad \sigma^2 \left(\nabla \cdot \nabla \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t}\right)$$

aligns with known interpolants between prior and target

Any other ways? YES!

The LHS only has **2 unknown terms**: scalar func $Z_{\pi_t}(t)$ and vector func $f(X,t)$

We can parameter network for $Z_{\pi_t}(t), f(X,t)$, and learn it by $\min\left|\left|\mathrm{LFS}\right|\right|^2$

Diffusion Neural samplers - idea 2.2

$dX_t = f(X_t)dt + \sigma\sqrt{2}dW_t, X_0 \sim p_{prior},$

Want a sample process (prior to target),

Fokker-Planck equation (in log space)

whose marginal density at every time step,

$$\partial_t \log p_t + \nabla \cdot f + \nabla \log p_t \cdot f - \sigma^2 ||\nabla \log p_t||^2 - \sigma^2 \Delta \log p_t = 0$$

aligns with known interpolants between prior and target

$\partial_t \log p_{prior}^{\beta_t} p_{target}^{1-\beta_t} - \partial_t Z_{\pi_t}$ … $\nabla \log p_{prior}^{\beta_t} p_{target}^{1-\beta_t}$ … $\left(\nabla \log p_{prior}^{\beta_t} p_{target}^{1-\beta_t}\right)$

Any other ways? YES!

The LHS only has **2 unknown terms**: scalar func $Z_{\pi_t}(t)$ and vector func $f(X,t)$

We can still match some forward and backward process!

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

If the marginal at diffusion time t is π_t

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

If the marginal at diffusion time t is π_t

its **time-reversal** is given by

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

If the marginal at diffusion time t is π_t

its **time-reversal** is given by

$$\mathrm{d}Y_t = -f(Y_t, T - t)\mathrm{d}t + 2\sigma^2\nabla\log\pi_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,\ Y_0 \sim \pi_T = p_{\mathrm{target}},$$

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

If the marginal at diffusion time t is π_t

its **time-reversal** is given by

$$\mathrm{d}Y_t = -f(Y_t, T-t)\mathrm{d}t + 2\sigma^2\nabla\log\pi_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,\ Y_0 \sim \pi_T = p_{\mathrm{target}},$$

"Nelson's Condition"

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

If the marginal at diffusion time t is π_t

its **time-reversal** is given by

$$\mathrm{d}Y_t = -f(Y_t, T-t)\mathrm{d}t + 2\sigma^2\nabla\log\pi_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,\ Y_0 \sim \pi_T = p_{\mathrm{target}},$$

“Nelson’s Condition” is an iff condition

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

If its **time-reversal** is given by

$$\mathrm{d}Y_t = -f(Y_t, T - t)\mathrm{d}t + 2\sigma^2\nabla\log \pi_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,\ Y_0 \sim \pi_T = p_{\mathrm{target}},$$

then the marginal for at X_t diffusion time t is π_t

"Nelson's Condition" is an iff condition

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

If its **time-reversal** is given by

$$\mathrm{d}Y_t = -f(Y_t, T-t)\mathrm{d}t + 2\sigma^2 \nabla \log \pi_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,\ Y_0 \sim \pi_T = p_{\text{target}},$$

then the marginal for at X_t diffusion time t is π_t

“Nelson’s Condition” is an iff condition

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

If its **time-reversal** is given by

known term

$$\mathrm{d}Y_t = -f(Y_t, T-t)\mathrm{d}t + 2\sigma^2\nabla\log\pi_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,\ Y_0 \sim \pi_T = p_{\mathrm{target}},$$

then the marginal for at X_t diffusion time t is π_t

“Nelson’s Condition” is an iff condition

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

Time-dependent network

If its **time-reversal** is given by

The same network **known term**

$$\mathrm{d}Y_t = -f(Y_t, T-t)\mathrm{d}t + 2\sigma^2\nabla\log\pi_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,\ Y_0 \sim \pi_T = p_{\text{target}},$$

then the marginal for at X_t diffusion time t is π_t

“Nelson’s Condition” is an iff condition

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

If its **time-reversal** is given by

$$\mathrm{d}Y_t = -f(Y_t, T-t)\mathrm{d}t + 2\sigma^2\nabla\log\pi_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,\ Y_0 \sim \pi_T = p_{\mathrm{target}},$$

then the marginal for at X_t diffusion time t is π_t

"Nelson's Condition" is an iff condition

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

If its **time-reversal** is given by

$$\mathrm{d}Y_t = -f(Y_t, T-t)\mathrm{d}t + 2\sigma^2\nabla\log\pi_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,\ Y_0 \sim \pi_T = p_{\text{target}},$$

then the marginal for at X_t diffusion time t is π_t

“Nelson’s Condition” is an iff condition

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

$$p_{\text{prior}}(X_0)N\left(X_{t_1}\middle|X_0\right)N\left(X_{t_2}\middle|X_{t_1}\right)\ldots N(X_{t_N}|X_{t_{N-1}}) \quad := q(X_{0:t_N})$$

$$\mathrm{d}Y_t = -f(Y_t, T-t)\mathrm{d}t + 2\sigma^2\nabla\log\pi_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,\ Y_0 \sim \pi_T = p_{\text{target}},$$

$$p_{\text{target}}(Y_0)N\left(Y_{t_1}\middle|Y_0\right)N\left(Y_{t_2}\middle|Y_{t_1}\right)\ldots N(Y_{t_N}|Y_{t_{N-1}}) \quad := p(X_{0:t_N})$$

"Nelson's Condition" is an iff condition

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

$$p_{\text{prior}}(X_0)N\left(X_{t_1}\middle|X_0\right)N\left(X_{t_2}\middle|X_{t_1}\right)\ldots N(X_{t_N}|X_{t_{N-1}}) \quad := q(X_{0:t_N})$$

$$\mathrm{d}Y_t = -f(Y_t, T-t)\mathrm{d}t + 2\sigma^2\nabla\log\pi_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim \pi_T = p_{\text{target}},$$

$$p_{\text{target}}(Y_0)N\left(Y_{t_1}\middle|Y_0\right)N\left(Y_{t_2}\middle|Y_{t_1}\right)\ldots N(Y_{t_N}|Y_{t_{N-1}}) \quad := p(X_{0:t_N})$$

“Nelson’s Condition” is an iff condition

Diffusion Neural samplers - idea 2.2

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

If its time-reversal is given by

$$p_{\text{prior}}(X_0)N\left(X_{t_1}\middle|X_0\right)N\left(X_{t_2}\middle|X_{t_1}\right)\dots N(X_{t_N}|X_{t_{N-1}}) \quad := q(X_{0:t_N})$$

$$\mathrm{d}Y_t = -f(Y_t, T-t)\mathrm{d}t + 2\sigma^2\nabla\log\pi_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t,\ Y_0 \sim \pi_T = p_{\text{target}},$$

then the marginal of X_t at diffusion time is π_t

$$\tilde{p}_{\text{target}}(Y_0)N\left(Y_{t_1}\middle|Y_0\right)N\left(Y_{t_2}\middle|Y_{t_1}\right)\dots N(Y_{t_N}|Y_{t_{N-1}}) \quad := \tilde{p}(X_{0:t_N})$$

“Nelson’s Condition” is an iff condition

Diffusion Neural samplers - idea 2.2

Match $q\left(X_{0:t_N}\right)$ with $\tilde{p}\left(X_{0:t_N}\right)$:

Diffusion Neural samplers - idea 2.2

Match $q(X_{0:t_N})$ with $\tilde{p}(X_{0:t_N})$:

💡 We can use all objectives in the previous slide (idea 1.1)

Diffusion Neural samplers - idea 2.2

Match $q\left(X_{0:t_N}\right)$ with $\tilde{p}\left(X_{0:t_N}\right)$:

$$D_{\mathrm{KL}}[q\left(X_{0:t_N}\right)||\tilde{p}\left(X_{0:t_N}\right)] = \mathrm{E}_q\left[\log \frac{q\left(X_{0:t_N}\right)}{\tilde{p}\left(X_{0:t_N}\right)}\right]$$

$$D_{\mathrm{LV}}[q\left(X_{0:t_N}\right)||\tilde{p}\left(X_{0:t_N}\right)] = \mathrm{Var}_\pi\left[\log \frac{q\left(X_{0:t_N}\right)}{\tilde{p}\left(X_{0:t_N}\right)}\right]$$

$$D_{\mathrm{TB}}[q\left(X_{0:t_N}\right)||\tilde{p}\left(X_{0:t_N}\right)] = \mathrm{E}_\pi\left[\left(\log \frac{q\left(X_{0:t_N}\right)}{\tilde{p}\left(X_{0:t_N}\right)} - k\right)^2\right]$$

Other choices exist, including sub-TB, DB, etc…

Diffusion Neural samplers - idea 2.2

Match $q(X_{0:t_N})$ with $\tilde{p}(X_{0:t_N})$:

Want a sample process (prior to target),

$$D_{\mathrm{KL}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{E}_q\left[\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})}\right]$$

whose marginal density at every time step,

$$D_{\mathrm{LV}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{Var}_\pi\left[\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})}\right]$$

aligns with known interpolants between prior and target

$$D_{\mathrm{TB}}[q(X_{0:t_N})||\tilde{p}(X_{0:t_N})] = \mathrm{E}_\pi\left[\left(\log\frac{q(X_{0:t_N})}{\tilde{p}(X_{0:t_N})} - k\right)^2\right]$$

match forward and backward process!

Other choices exist, including sub-TB, DB, etc...

Diffusion Neural samplers - idea 2

Want a sample process (prior to target),

whose marginal density at every time step,

aligns with known interpolants between prior and target

1.1 align the marginal to the desired marginal by satisfying PDE

1.2 align forward with backward

Diffusion Neural samplers - idea 2

This includes

(1) NETS (non-equilibrium transport sampler)

(2) PINN (physics-informed neural networks) sampler — **satisfying PDE**

(3) LFIS (Liouville Flow Importance Sampler)

(4) CMCD (Controlled Monte Carlo Diffusions) — **aligning forward with backward**

…

Diffusion Neural samplers - idea 3

$\mathrm{d}X_t = f_\theta(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

Diffusion Neural samplers - idea 3

$\mathrm{d}X_t = \underbrace{f_\theta(X_t, t)}_{\text{What if we do not train it?}}\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$, we want $X_T \sim p_{\mathrm{target}}$.

Diffusion Neural samplers - idea 3

$$\mathrm{d}X_t = \underbrace{f(X_t, t)}_{\text{What if we do not train it?}}\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}},$$

Diffusion Neural samplers - idea 3

$$\mathrm{d}X_t = f\underbrace{(X_t, t)}_{\text{What if we do not train it?}}\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}, X_T \nsim p_{\mathrm{target}}$$

Diffusion Neural samplers - idea 3

$$\mathrm{d}X_t = \underbrace{f(X_t, t)}_{\text{What if we do not train it?}}\mathrm{d}t + \sigma\sqrt{2}\,\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}, \boxed{X_T \nsim p_{\mathrm{target}}}$$

What if we do not train it?

How to rescue?

Diffusion Neural samplers - idea 3

$$\mathrm{d}X_t = \underbrace{f(X_t, t)}_{\text{What if we do not train it?}}\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}}, \boxed{X_T \nsim p_{\text{target}}}$$

How to rescue?

Importance Sampling

Diffusion Neural samplers - idea 3

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$$

Diffusion Neural samplers - idea 3

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}, \quad \Rightarrow \quad \overrightarrow{\mathbf{Q}}(X)$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}, \quad \Rightarrow \quad \overleftarrow{\mathbf{P}}(X)$$

Diffusion Neural samplers - idea 3

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}, \quad \Rightarrow \quad \overrightarrow{\mathbf{Q}}(X)$$

$$\mathrm{d}Y_t = g(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}, \quad \Rightarrow \quad \overleftarrow{\mathbf{P}}(X)$$

Importance weight: $\dfrac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$

Diffusion Neural samplers - idea 3

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}}, \quad \Rightarrow \quad \overrightarrow{\mathbf{Q}}(X)$$

$$\mathrm{d}Y_t = \boxed{g_\theta(Y_t, t)}\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}, \quad \Rightarrow \quad \overleftarrow{\mathbf{P}}(X)$$

Also possible to learn it

Importance weight: $\dfrac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$

Diffusion Neural samplers - idea 3

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}}, \Rightarrow \overrightarrow{\mathbf{Q}}(X)$$

align

$$\mathrm{d}Y_t = g_\theta(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}, \Rightarrow \overleftarrow{\mathbf{P}}(X)$$

Also possible to learn it

Small variance

Importance weight: $\dfrac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$

Diffusion Neural samplers - idea 3

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}}, \quad \Rightarrow \quad \overrightarrow{\mathbf{Q}}(X)$$

align

$$\mathrm{d}Y_t = g_\theta(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}, \quad \Rightarrow \quad \overleftarrow{\mathbf{P}}(X)$$

Also possible to learn it

Small variance

Importance weight: $\dfrac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$

Diffusion Neural samplers - idea 3

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}}, \Rightarrow \overrightarrow{\mathbf{Q}}(X)$$

Predefine a sample process (prior to target),

$$\mathrm{d}Y_t = g_\theta(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}}, \Rightarrow \overleftarrow{\mathbf{P}}(X)$$

Also possible to learn it

Small variance

Importance weight: $\dfrac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$

Diffusion Neural samplers - idea 3

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}, \Rightarrow \overrightarrow{\mathbf{Q}}(X)$$

Predefine a sample process (prior to target),

define or train a backward process (target to prior),

$$\mathrm{d}Y_t = g_\theta(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\mathrm{target}}, \Rightarrow \overleftarrow{\mathbf{P}}(X)$$

Also possible to learn it

Small variance

Importance weight: $\dfrac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$

Diffusion Neural samplers - idea 3

$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$ ➡ $\overrightarrow{\mathbf{Q}}(X)$

Predefine a sample process (prior to target),

define or train a backward process (target to prior),

$\mathrm{d}Y_t = g_\theta(Y_t, t)\mathrm{d}t + \sigma\sqrt{2}\mathrm{d}W_t, Y_0 \sim p_{\text{target}},$ ➡ $\overleftarrow{\mathbf{P}}(X)$

Also possible to learn it

perform importance sampling

Small variance

Importance weight: $\frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}$

Diffusion Neural samplers - idea 3

This includes

(1) AIS (Annealed Importance Sampling)

(2) MCD (Monte Carlo Diffusions)

(3) LDVI (Langevin Diffusion Variational Inference)

…

Fixed target and proposal

Fixed proposal, learned target

Diffusion Neural samplers

Overall framework:
1. Time-reversal sampler
2. Escorted transport sampler
3. Annealed variance reduction sampler

Objectives:

👉 Write down backward and forward, align them (path measure alignment)

👉 Write down the marginal, align it with the sampling process (marginal alignment)

Diffusion Neural samplers

	Time-reversal sampler	Escorted transport sampler	Annealed Variance Reduction Sampler
Path measure alignment	DDS, DIS, PIS, GFN	CMCD, SLCD	MCD
Marginal alignment	iDEM, RDMC, PINN-sampler	NETS, PINN-sampler, LFIS	

Diffusion Neural samplers - Desiderata

Let's look at the loss again, for example:

Diffusion Neural samplers - Desiderata

Let's look at the loss again, for example:

$$D_{\mathrm{KL}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{E}_{\overrightarrow{\mathbf{Q}}}\left[\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{LV}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{Var}_{\overrightarrow{\boldsymbol{\pi}}}\left[\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{TB}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{E}_{\overrightarrow{\boldsymbol{\pi}}}\left[\left(\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)} - k\right)^2\right]$$

Diffusion Neural samplers - Desiderata

Let's look at the loss again, for example:

$$D_{\mathrm{KL}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathbb{E}_{\overrightarrow{\mathbf{Q}}}\left[\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{LV}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{Var}_{\overrightarrow{\boldsymbol{\pi}}}\left[\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{TB}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathbb{E}_{\overrightarrow{\boldsymbol{\pi}}}\left[\left(\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)} - k\right)^2\right]$$

Diffusion Neural samplers - Desiderata

Let's look at the loss again, for example:

$$D_{\mathrm{KL}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathbb{E}_{\overrightarrow{\mathbf{Q}}}\left[\log\frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{LV}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{Var}_{\overrightarrow{\boldsymbol{\pi}}}\left[\log\frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{TB}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathbb{E}_{\overrightarrow{\boldsymbol{\pi}}}\left[\left(\log\frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)} - k\right)^2\right]$$

😔 need to simulate the trajectory – expensive!

Diffusion Neural samplers - Desiderata

Let's look at the loss again, for example:

$$D_{\mathrm{KL}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathbb{E}_{\overrightarrow{\mathbf{Q}}}\left[\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{LV}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathrm{Var}_{\overrightarrow{\boldsymbol{\pi}}}\left[\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)}\right]$$

$$D_{\mathrm{TB}}[\overrightarrow{\mathbf{Q}}||\overleftarrow{\mathbf{P}}] = \mathbb{E}_{\overrightarrow{\boldsymbol{\pi}}}\left[\left(\log \frac{\mathrm{d}\overrightarrow{\mathbf{Q}}(X)}{\mathrm{d}\overleftarrow{\mathbf{P}}(X)} - k\right)^2\right]$$

😔 need to simulate the trajectory – expensive!

Any ways for "simulation-free" training?

Simulation-free training of Diffusion Neural samplers

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling $X_t = F_\theta(Z, t), \quad Z \sim p_{\text{base}}$

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling $X_t = F_\theta(Z, t), \quad Z \sim p_{\text{base}}$

The second way of sampling

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling $X_t = F_\theta(Z, t), \quad Z \sim p_{\text{base}}$

The second way of sampling $X_0 = F_\theta(Z, 0), Z \sim p_{\text{base}}$

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling $X_t = F_\theta(Z, t), \quad Z \sim p_{\mathrm{base}}$

The second way of sampling

$$X_0 = F_\theta(Z, 0), Z \sim p_{\mathrm{base}}$$
$$\mathrm{d}X_t = \partial_t F_\theta(Z, t)\mathrm{d}t$$

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling $$X_t = F_\theta(Z, t), \quad Z \sim p_{\mathrm{base}}$$

The second way of sampling
$$X_0 = F_\theta(Z, 0), Z \sim p_{\mathrm{base}}$$
$$\mathrm{d}X_t = \partial_t F_\theta(\underbrace{Z,}_{Z = F_\theta^{-1}(X_t, t)} t)\mathrm{d}t$$

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling $X_t = F_\theta(Z, t), \quad Z \sim p_{\text{base}}$

The second way of sampling

$$X_0 = F_\theta(Z, 0), Z \sim p_{\text{base}}$$
$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t, t), t\big)\mathrm{d}t$$

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling $X_t = F_\theta(Z, t), \quad Z \sim p_{\text{base}}$

The second way of sampling

$$X_0 = F_\theta(Z, 0), Z \sim p_{\text{base}}$$
$$\mathrm{d}X_t = \underbrace{\partial_t F_\theta\left(F_\theta^{-1}(X_t, t), t\right)}_{\text{Standard form of ODE}}\mathrm{d}t$$

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling $X_t = F_\theta(Z, t), \quad Z \sim p_{\text{base}}$

The second way of sampling

$$X_0 = F_\theta(Z, 0), Z \sim p_{\text{base}}$$
$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t, t), t\big)\mathrm{d}t$$

$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t, t), t\big)\mathrm{d}t \qquad + \sigma_t\sqrt{2}\mathrm{d}W_t$$

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling $$X_t = F_\theta(Z, t), \quad Z \sim p_{\text{base}}$$

The second way of sampling
$$X_0 = F_\theta(Z, 0), Z \sim p_{\text{base}}$$
$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t, t), t\big)\mathrm{d}t$$
$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t, t), t\big)\mathrm{d}t + \sigma_t^2 \nabla \log q_\theta(X_t, t)\mathrm{d}t + \sigma_t \sqrt{2}\mathrm{d}W_t$$

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling $X_t = F_\theta(Z, t), \quad Z \sim p_{\mathrm{base}}$

The second way of sampling $X_0 = F_\theta(Z, 0), Z \sim p_{\mathrm{base}}$

$$\mathrm{d}X_t = \partial_t F_\theta\left(F_\theta^{-1}(X_t, t), t\right)\mathrm{d}t$$

Easily obtained by NF

$$\mathrm{d}X_t = \partial_t F_\theta\left(F_\theta^{-1}(X_t, t), t\right)\mathrm{d}t + \sigma_t^2 \nabla \log q_\theta(X_t, t)\mathrm{d}t + \sigma_t \sqrt{2}\mathrm{d}W_t$$

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling $X_t = F_\theta(Z, t), \quad Z \sim p_{\text{base}}$ **directly sample from time t**

The second way of sampling

$$X_0 = F_\theta(Z, 0), Z \sim p_{\text{base}}$$

$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t, t), t\big)\mathrm{d}t$$

$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t, t), t\big)\mathrm{d}t + \sigma_t^2 \nabla \log q_\theta(X_t, t)\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t$$

Simulation-free training of Diffusion Neural samplers

avoid simulating the trajectory (entirely) during training.

💡 using a time-dependent normalizing flow

Define $F_\theta(\cdot, t)$ as an invertible function

The first way of sampling $X_t = F_\theta(Z, t), \quad Z \sim p_{\text{base}}$

directly sample from time t

The second way of sampling

$$X_0 = F_\theta(Z, 0), Z \sim p_{\text{base}}$$

$$\mathrm{d}X_t = \partial_t F_\theta\left(F_\theta^{-1}(X_t, t), t\right)\mathrm{d}t$$

$$\mathrm{d}X_t = \partial_t F_\theta\left(F_\theta^{-1}(X_t, t), t\right)\mathrm{d}t + \sigma_t^2 \nabla \log q_\theta(X_t, t)\mathrm{d}t + \sigma_t \sqrt{2}\mathrm{d}W_t$$

Calculate the same loss as other diffusion samplers

Simulation-free training of Diffusion Neural samplers

$$\mathrm{d}X_t = \partial_t F_\theta\left(F_\theta^{-1}(X_t, t), t\right)\mathrm{d}t + \sigma_t^2 \nabla \log q_\theta(X_t, t)\mathrm{d}t + \sigma_t \sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$$

Simulation-free training of Diffusion Neural samplers

$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t, t), t\big)\mathrm{d}t + \sigma_t^2 \nabla \log q_\theta(X_t, t)\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$$

Align

$$\mathrm{d}X_t = \underbrace{g(X_t)}_{\text{a simple function, e.g., 0}}\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t^-, X_T \sim p_{\mathrm{target}}$$

Simulation-free training of Diffusion Neural samplers

Align

$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t, t), t\big)\mathrm{d}t + \sigma_t^2 \nabla \log q_\theta(X_t, t)\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$$

time-reversal

$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t, t), t\big)\mathrm{d}t - \sigma_t^2 \nabla \log q_\theta(X_t, t)\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t^-, X_T \sim q_\theta(\cdot, T)$$

$$\mathrm{d}X_t = \underbrace{g(X_t)}_{\text{a simple function, e.g., 0}}\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t^-, X_T \sim p_{\mathrm{target}}$$

Simulation-free training of Diffusion Neural samplers

$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t, t), t\big)\mathrm{d}t + \sigma_t^2 \nabla \log q_\theta(X_t, t)\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}}$$

time-reversal

Align

$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t, t), t\big)\mathrm{d}t - \sigma_t^2 \nabla \log q_\theta(X_t, t)\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t^-, X_T \sim q_\theta(\cdot, T)$$

$$\mathrm{d}X_t = \underbrace{g(X_t)}_{\text{a simple function, e.g., 0}}\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t^-, X_T \sim p_{\text{target}}$$

Simulation-free training of Diffusion Neural samplers

$$\mathrm{d}X_t = \partial_t F_\theta\left(F_\theta^{-1}(X_t, t), t\right)\mathrm{d}t + \sigma_t^2 \nabla \log q_\theta(X_t, t)\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$$

time-reversal

Align

$$\mathrm{d}X_t = \partial_t F_\theta\left(F_\theta^{-1}(X_t, t), t\right)\mathrm{d}t - \sigma_t^2 \nabla \log q_\theta(X_t, t)\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t^-, X_T \sim q_\theta(\cdot, T)$$

$$\mathrm{d}X_t = \underbrace{g(X_t)}_{\text{a simple function, e.g., 0}}\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t^-, X_T \sim p_{\mathrm{target}}$$

✅ **same direction – Girsanov Theorem applicable**

Simulation-free training of Diffusion Neural samplers

$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t,t),t\big)\mathrm{d}t + \sigma_t^2 \nabla \log q_\theta(X_t,t)\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\mathrm{prior}}$$

time-reversal

Align

$$\mathrm{d}X_t = \partial_t F_\theta\big(F_\theta^{-1}(X_t,t),t\big)\mathrm{d}t - \sigma_t^2 \nabla \log q_\theta(X_t,t)\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t^-, X_T \sim q_\theta(\cdot,T)$$

$$\mathrm{d}X_t = \underbrace{g(X_t)}_{\text{a simple function, e.g., 0}}\mathrm{d}t + \sigma_t\sqrt{2}\mathrm{d}W_t^-, X_T \sim p_{\mathrm{target}}$$

✅ **same direction – Girsanov Theorem applicable**

✅ **simulation-free evaluation – can always obtain sample by 1-step** $X_t = F_\theta(Z,t), Z \sim p_{\mathrm{base}}$

Simulation-free training of Diffusion Neural samplers

😄 Great! How does it perform?

Simulation-free training of Diffusion Neural samplers

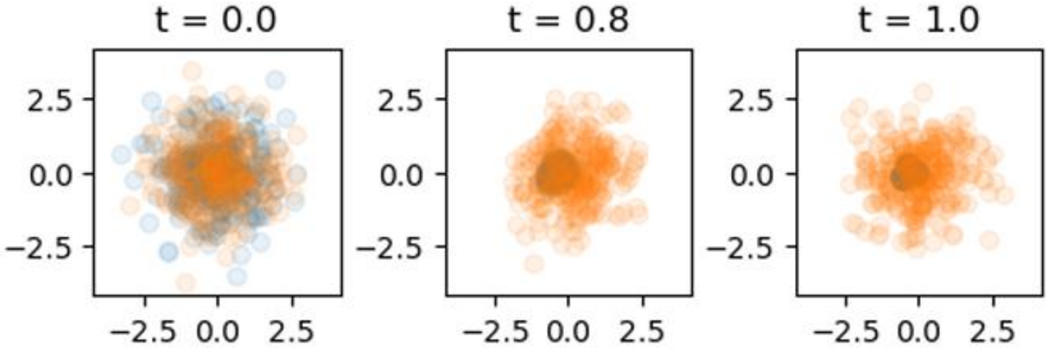
😄 Great! How does it perform?

😢 unfortunately…

Simulation-free training of Diffusion Neural samplers

😄 Great! How does it perform?

😢 unfortunately…

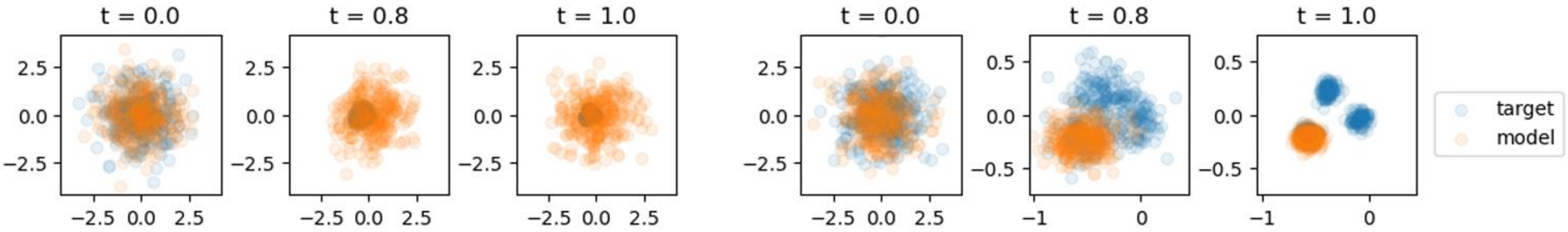


initialization

Simulation-free training of Diffusion Neural samplers

😄 Great! How does it perform?

😢 unfortunately…

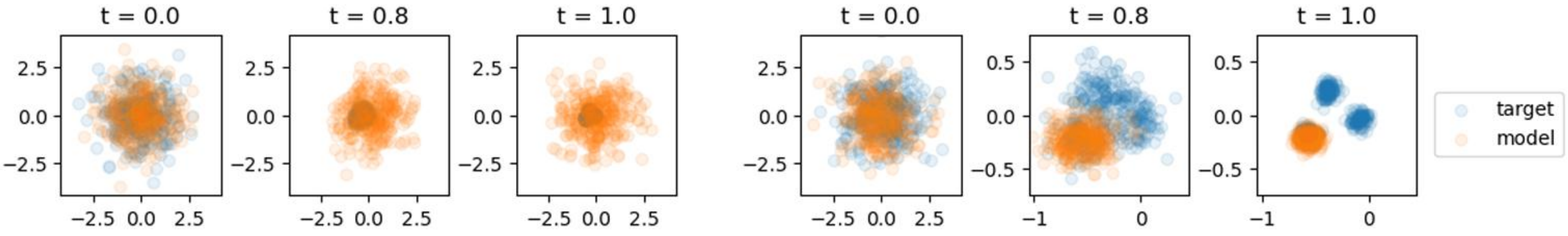


initialization

after training

Simulation-free training of Diffusion Neural samplers

Why?



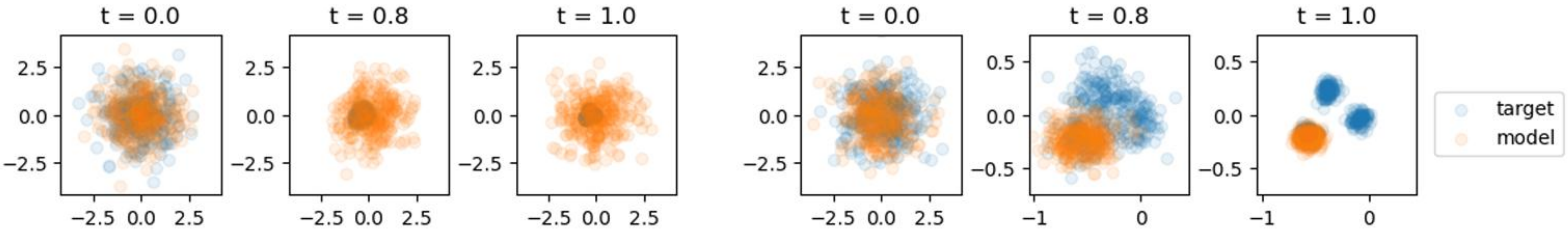
initialization

after training

Simulation-free training of Diffusion Neural samplers

Why?

Objective? same as DDS



initialization

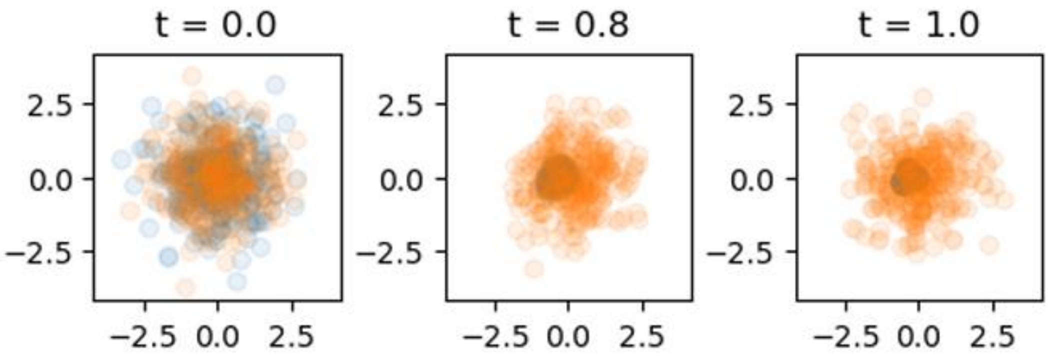
after training

Simulation-free training of Diffusion Neural samplers

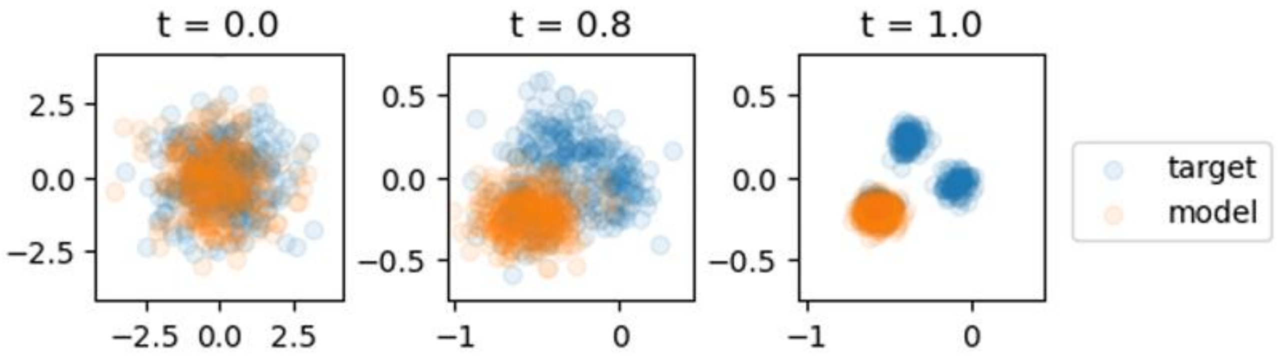
Why?

Objective? 🫤 same as DDS

Capacity? 🫤 target is so simple



initialization



after training

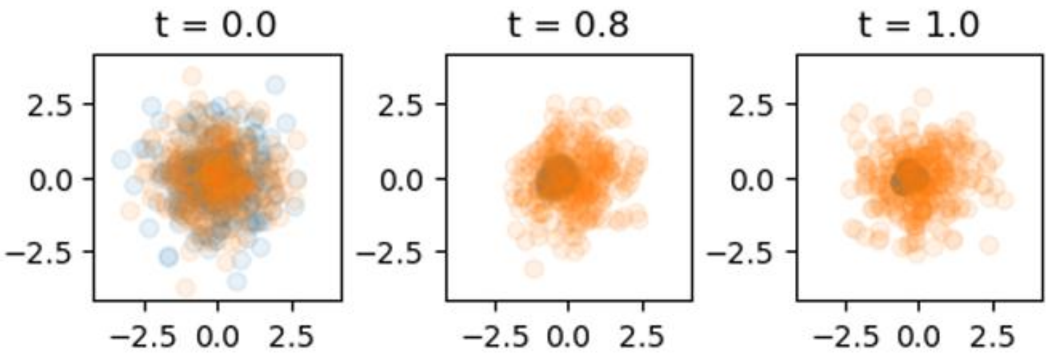
Simulation-free training of Diffusion Neural samplers

Why?

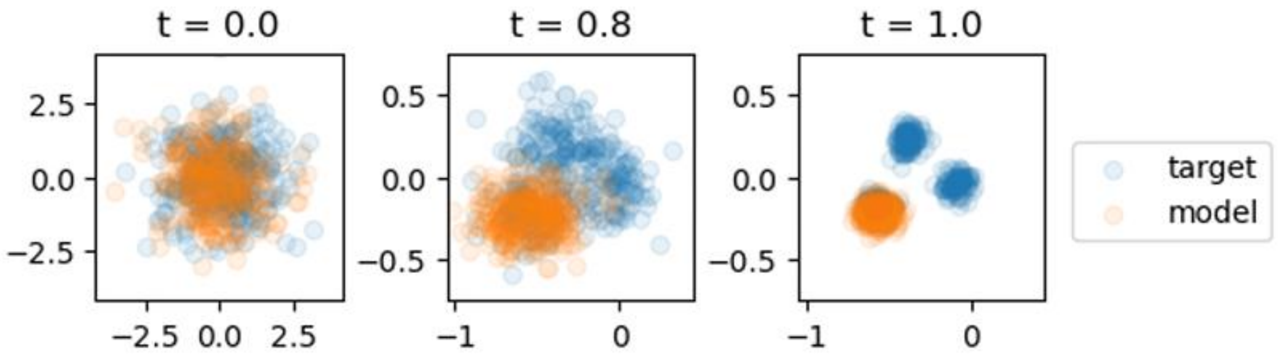
Objective? 🫤 same as DDS

Capacity? 🫤 target is so simple

Network parameterization? 🤔 might be the reason



initialization



after training

Langevin Preconditioning

Langevin Preconditioning

a. DDS/PIS/DDS/GFN...

$$f_\theta(\cdot, t) = \mathrm{NN}_{1,\theta}(\cdot, t) + \mathrm{NN}_{2,\theta}(t) \circ \nabla \log p_{\mathrm{target}}(\cdot)$$

Langevin Preconditioning

a. DDS/PIS/DDS/GFN...

$$f_\theta(\cdot, t) = \mathrm{NN}_{1,\theta}(\cdot, t) + \mathrm{NN}_{2,\theta}(t) \circ \boxed{\nabla \log p_{\text{target}}(\cdot)}$$

b. CMCD/NETS

$$dX_t = \left(f_\theta(X_t, t) + \boxed{\sigma_t^2 \nabla \log \pi_t(X_t)\right)dt} + \sqrt{2}\,\sigma_t dW_t \qquad \overrightarrow{\mathbf{Q}_{\boldsymbol{\theta}}}(X)$$

$$dX_t = \left(f_\theta(X_t, t) - \sigma_t^2 \nabla \log \pi_t(X_t)\right)dt + \sqrt{2}\,\sigma_t dW_t^- \qquad \overleftarrow{\mathbf{P}_\theta}(X)$$

Langevin Preconditioning

a. DDS/PIS/DDS/GFN...

$$f_\theta(\cdot, t) = \mathrm{NN}_{1,\theta}(\cdot, t) + \mathrm{NN}_{2,\theta}(t) \circ \boxed{\nabla \log p_{\text{target}}(\cdot)}$$

b. CMCD/NETS

$$dX_t = \left(f_\theta(X_t, t) + \boxed{\sigma_t^2 \nabla \log \pi_t(X_t)\right) dt} + \sqrt{2}\, \sigma_t dW_t \qquad \overrightarrow{\mathbf{Q}_{\boldsymbol{\theta}}}(X)$$

$$dX_t = \left(f_\theta(X_t, t) - \sigma_t^2 \nabla \log \pi_t(X_t)\right) dt + \sqrt{2}\, \sigma_t dW_t^- \qquad \overleftarrow{\mathbf{P}_\theta}(X)$$

Langevin Preconditioning

a. DDS/PIS/DDS/GFN...

$$f_\theta(\cdot, t) = \mathrm{NN}_{1,\theta}(\cdot, t) + \mathrm{NN}_{2,\theta}(t) \circ \nabla \log p_{\mathrm{target}}(\cdot)$$

When we do simulation, we are secretly running a Langevin!

b. CMCD/NETS

$$dX_t = (f_\theta(X_t, t) + \sigma_t^2 \nabla \log \pi_t(X_t))dt + \sqrt{2}\,\sigma_t dW_t \qquad \overrightarrow{\mathbf{Q}_\theta}(X)$$

$$dX_t = (f_\theta(X_t, t) - \sigma_t^2 \nabla \log \pi_t(X_t))dt + \sqrt{2}\,\sigma_t dW_t^- \qquad \overleftarrow{\mathbf{P}_\theta}(X)$$

Langevin Preconditioning

a. DDS/PIS/DDS/GFN...

$$f_\theta(\cdot, t) = \mathrm{NN}_{1,\theta}(\cdot, t) + \mathrm{NN}_{2,\theta}(t) \circ \nabla \log p_{\mathrm{target}}(\cdot)$$

When we do simulation, we are secretly running a Langevin!

b. CMCD/NETS

What if we remove this Langevin?

$$dX_t = \left(f_\theta(X_t, t) + \sigma_t^2 \nabla \log \pi_t(X_t)\right)dt + \sqrt{2}\,\sigma_t dW_t \qquad \overrightarrow{\mathbf{Q}}_\theta(X)$$

$$dX_t = \left(f_\theta(X_t, t) - \sigma_t^2 \nabla \log \pi_t(X_t)\right)dt + \sqrt{2}\,\sigma_t dW_t^- \qquad \overleftarrow{\mathbf{P}}_\theta(X)$$

Langevin Preconditioning

a. DDS/PIS/DDS/GFN...

$$f_\theta(\cdot,t) = \mathrm{NN}_{1,\theta}(\cdot,t) + \mathrm{NN}_{2,\theta}(t) \circ \nabla \log p_{\mathrm{target}}(\cdot)$$

b. CMCD

$$dX_t = \ (f_\theta(X_t,t) + \sigma_t^2 \nabla \log \pi_t(X_t))dt + \sqrt{2}\,\sigma_t dW_t \qquad \overrightarrow{\mathbf{Q}_{\boldsymbol{\theta}}}(X)$$

$$dX_t = \ (f_\theta(X_t,t) - \sigma_t^2 \nabla \log \pi_t(X_t))dt + \sqrt{2}\,\sigma_t dW_t^- \qquad \overleftarrow{\mathbf{P}_{\theta}}(X)$$

c. PINN (NETS)

$$dX_t = \ (f_\theta(X_t,t) + \sigma_t^2 \nabla \log \pi_t(X_t))dt + \sqrt{2}\,\sigma_t dW_t \qquad \overrightarrow{\mathbf{Q}_{\boldsymbol{\theta}}}(X)$$

Langevin Preconditioning

a. DDS/PIS/DDS/GFN...

$$f_\theta(\cdot, t) = \mathrm{NN}_{1,\theta}(\cdot, t) + \mathrm{NN}_{2,\theta}(t) \circ \nabla \log p_{\mathrm{target}}(\cdot)$$

b. CMCD

$$dX_t = \left(f_\theta(X_t, t) + \sigma_t^2 \nabla \log \pi_t(X_t)\right)dt + \sqrt{2}\,\sigma_t dW_t \qquad \overrightarrow{\mathbf{Q}_{\boldsymbol{\theta}}}(X)$$

$$dX_t = \left(f_\theta(X_t, t) - 2\sigma_t^2 \nabla \log \pi_t(X_t)\right)dt + \sqrt{2}\,\sigma_t dW_t^- \qquad \overleftarrow{\mathbf{P}_\theta}(X)$$

c. PINN (NETS)

$$dX_t = \left(f_\theta(X_t, t) + \sigma_t^2 \nabla \log \pi_t(X_t)\right)dt + \sqrt{2}\,\sigma_t dW_t \qquad \overrightarrow{\mathbf{Q}_{\boldsymbol{\theta}}}(X)$$

Langevin Preconditioning

a. DDS/PIS/DDS/GFN...

$$f_\theta(\cdot, t) = \mathrm{NN}_{1,\theta}(\cdot, t) + \mathrm{NN}_{2,\theta}(t) \circ \nabla \log p_{\mathrm{target}}(\cdot)$$

b. CMCD

$$dX_t = (f_\theta(X_t, t) + \sigma_t^2 \nabla \log \pi_t(X_t))dt + \sqrt{2}\, \sigma_t dW_t \qquad \overrightarrow{\mathbf{Q}_{\boldsymbol{\theta}}}(X)$$

$$dX_t = (f_\theta(X_t, t) - 2\sigma_t^2 \nabla \log \pi_t(X_t))dt + \sqrt{2}\, \sigma_t dW_t^- \qquad \overleftarrow{\mathbf{P}_{\theta}}(X)$$

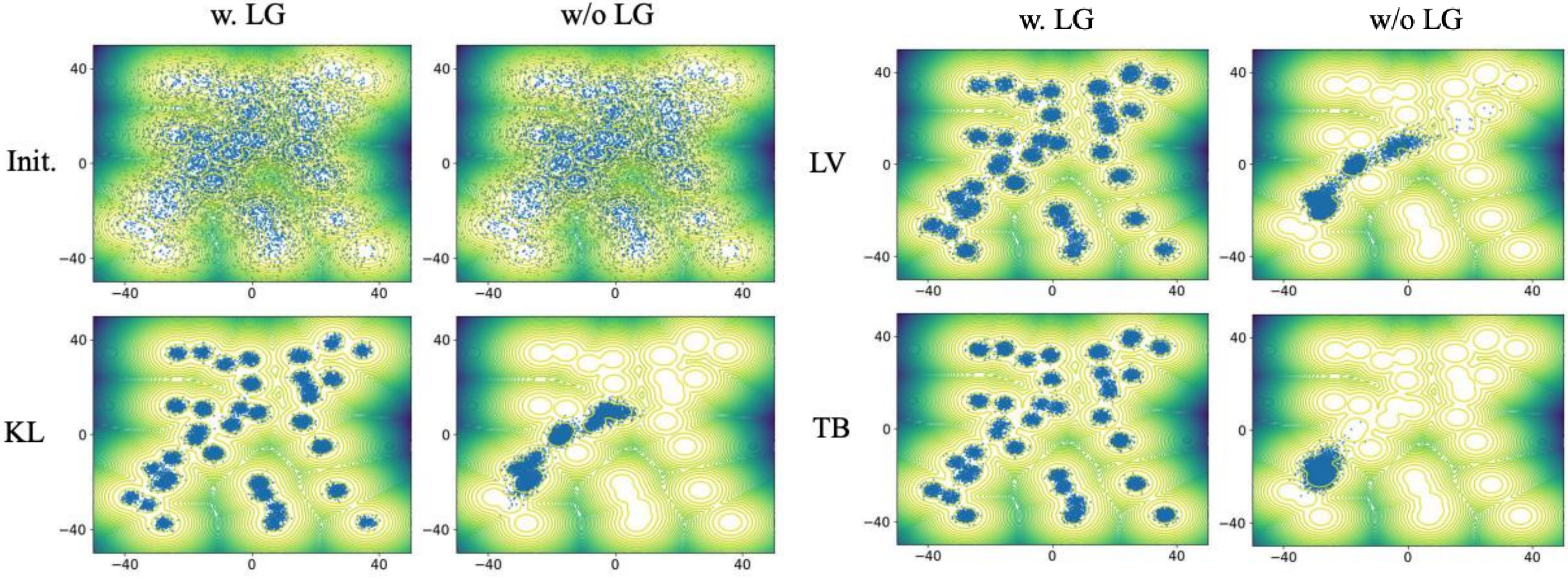
c. PINN (NETS)

$$dX_t = (f_\theta(X_t, t) + \sigma_t^2 \nabla \log \pi_t(X_t))dt + \sqrt{2}\, \sigma_t dW_t \qquad \overrightarrow{\mathbf{Q}_{\boldsymbol{\theta}}}(X)$$

When we do simulation with $\mathbf{Q}_{\boldsymbol{\theta}}$, we do not have the secret Langevin anymore

Empirical Results

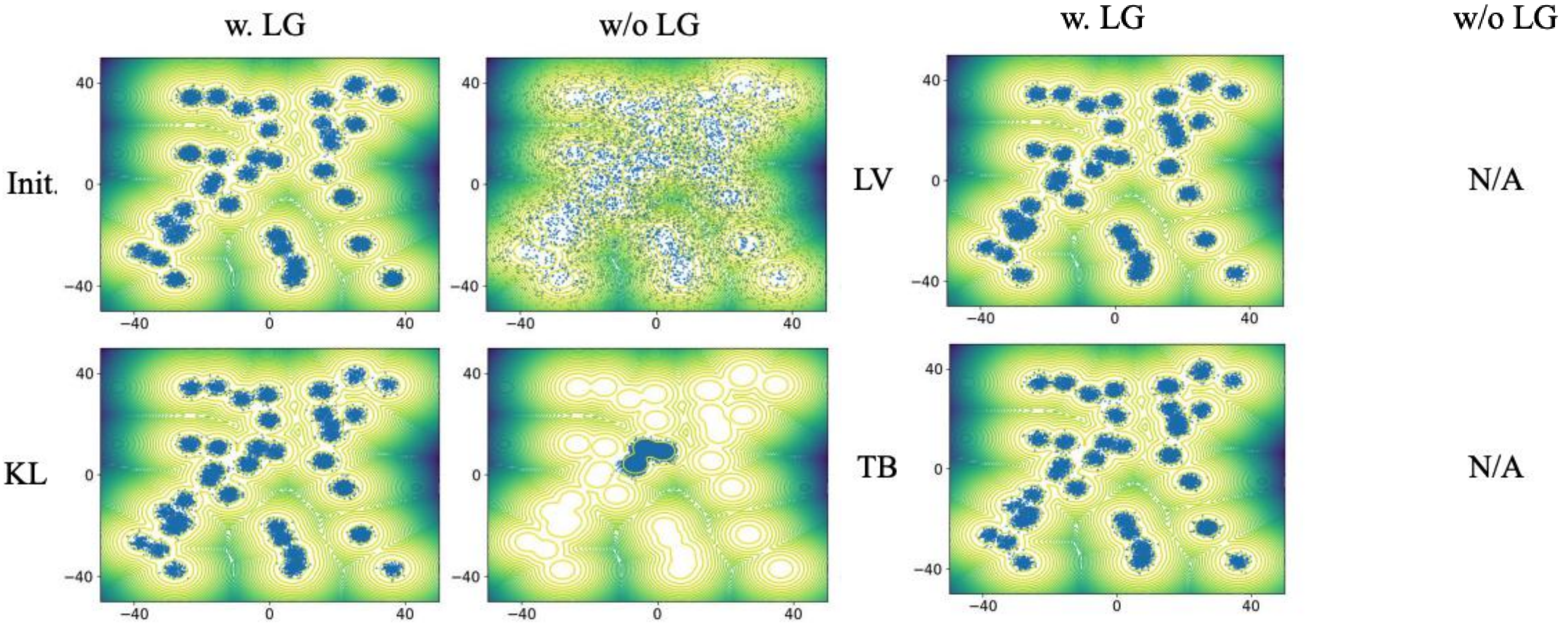
a. Langevin precondition is necessary to prevent mode collapse



DDS

Empirical Results

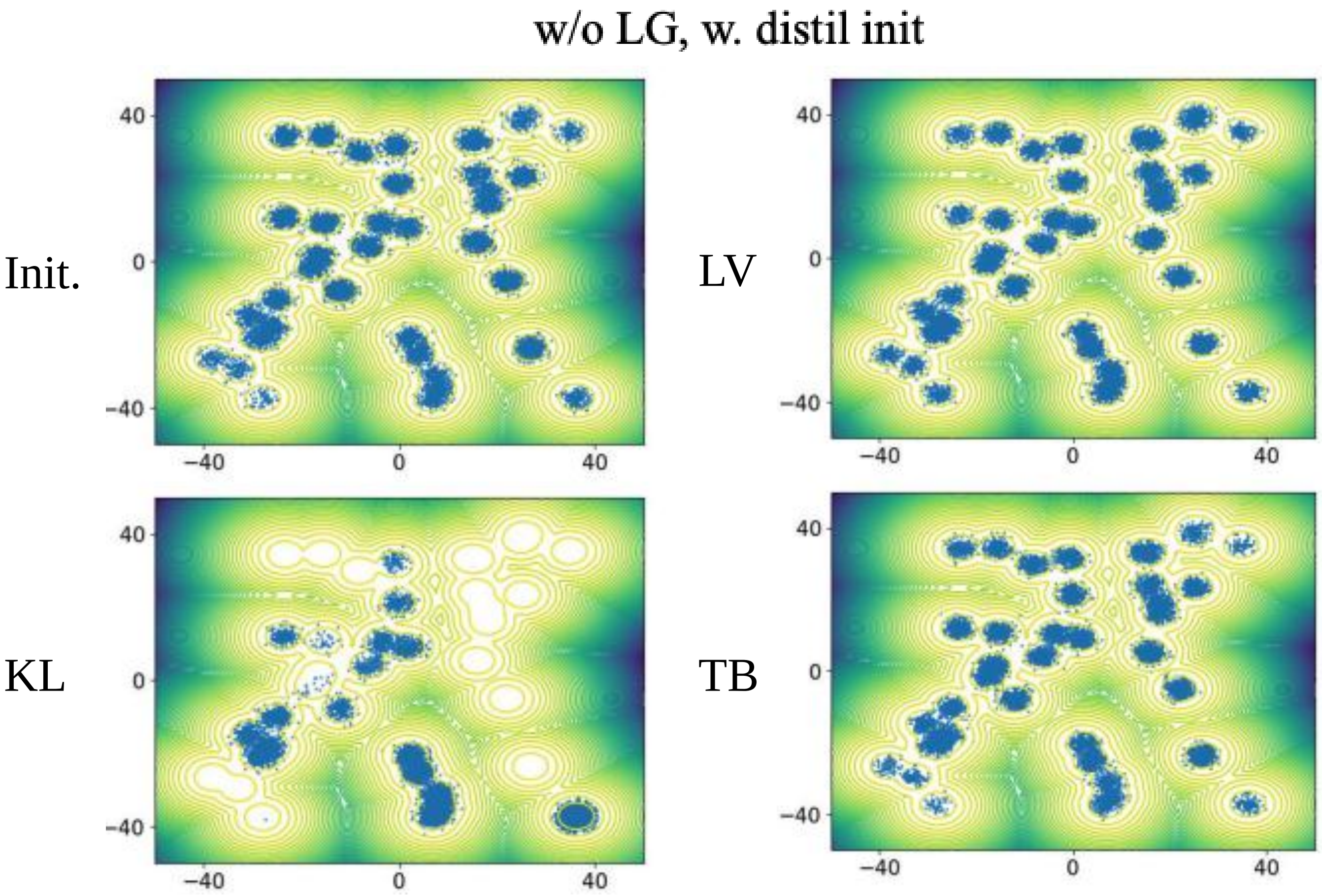
a. Langevin precondition is necessary to prevent mode collapse



CMCD

Empirical Results

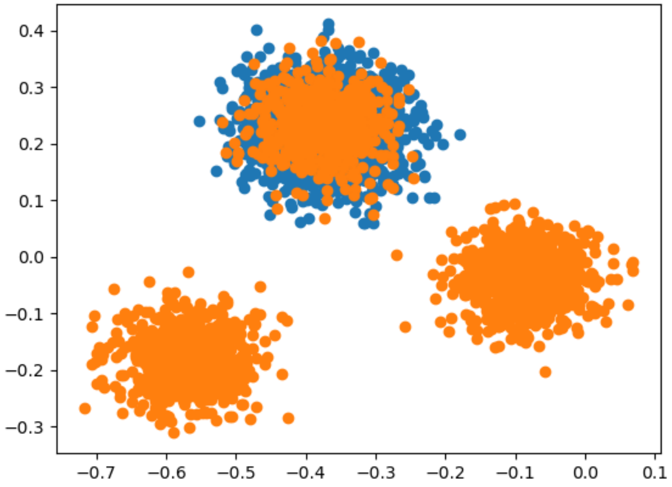
b. Mode collapse can happen even starting with “perfect” initialization.



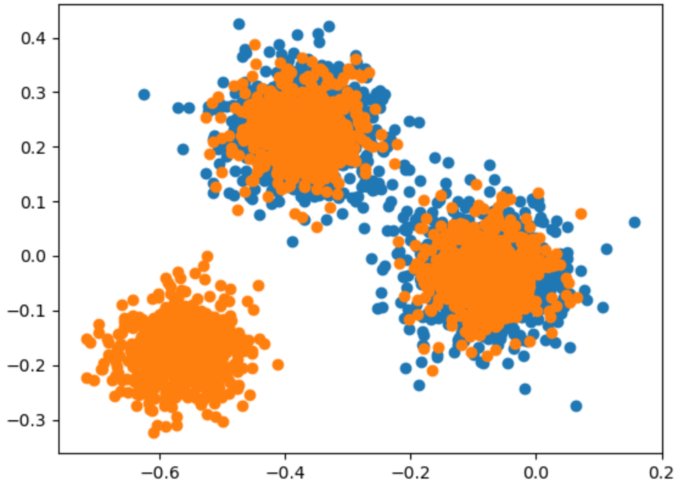
Empirical Results

DDS w/o Langevin for GMM-3:

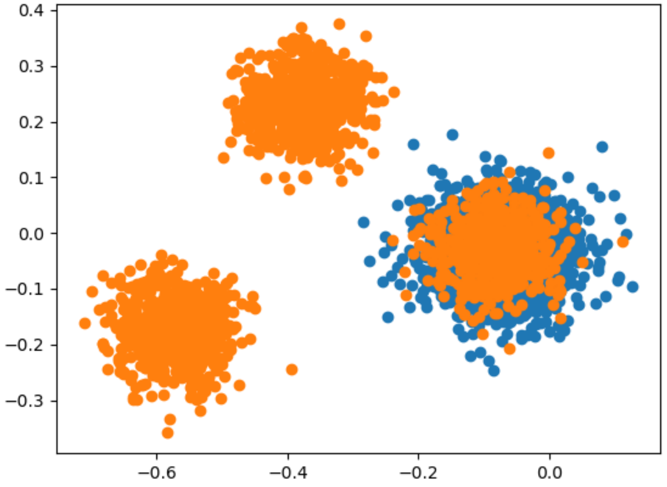
KL



LV



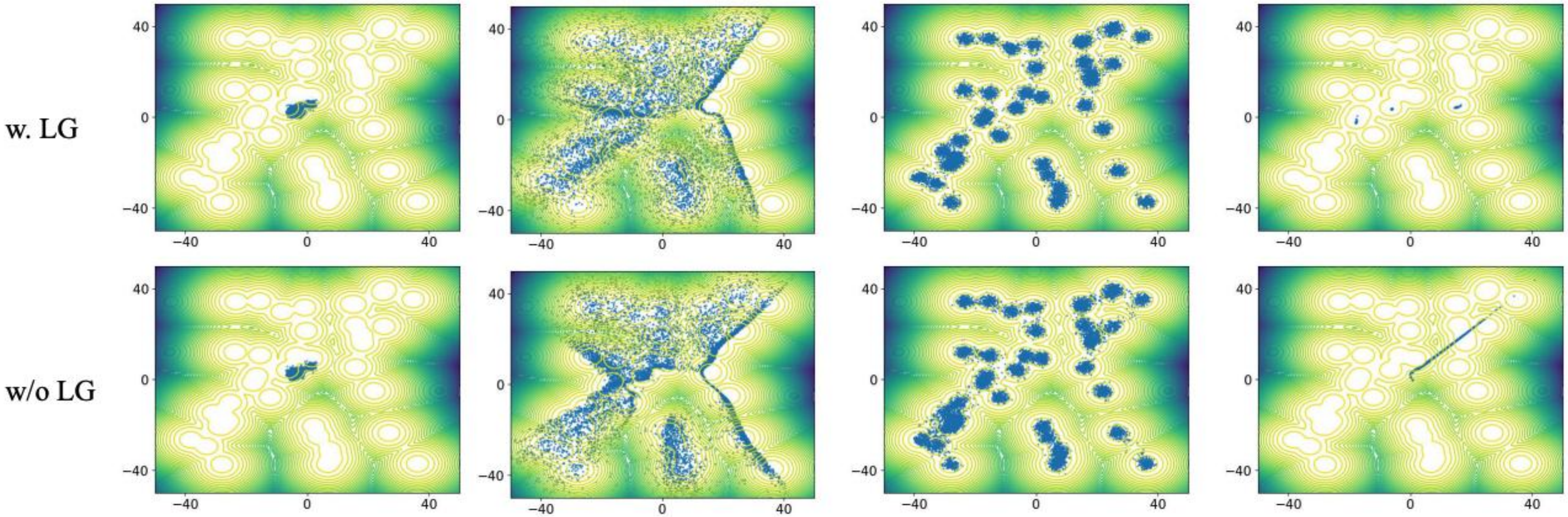
TB



Empirical Results

c. PINN objective is different

1. **Sensitive to interpolant**
2. **Sensitive to prior size**
3. **Robust to Langevin**



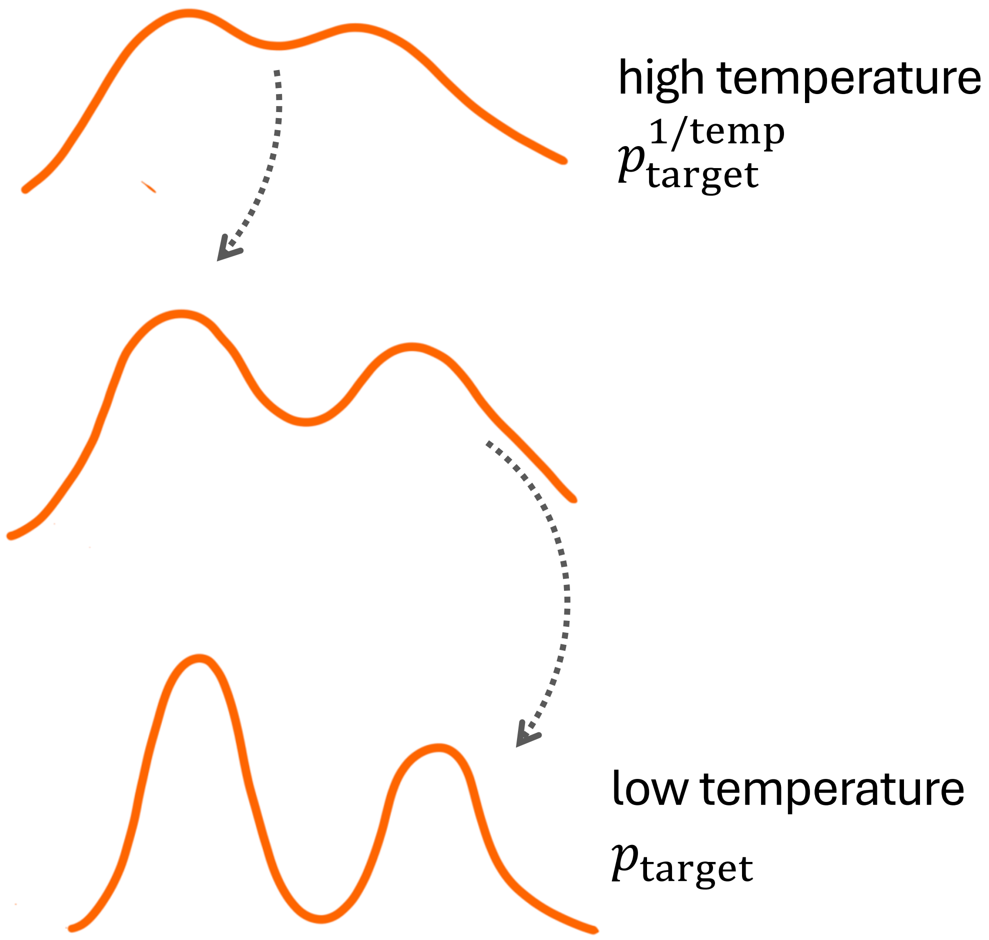
geom interpolate, prior $N(0, 2\boldsymbol{I})$ | geom interpolate, prior $N(0, 30^2\boldsymbol{I})$ | mode interpolate, prior $N(0, 2\boldsymbol{I})$ | mode interpolate, prior $N(0, 30^2\boldsymbol{I})$

Sample Efficiency?

If neural samplers need to run Langevin secretly,

Why not **directly run MCMC to collect data?**

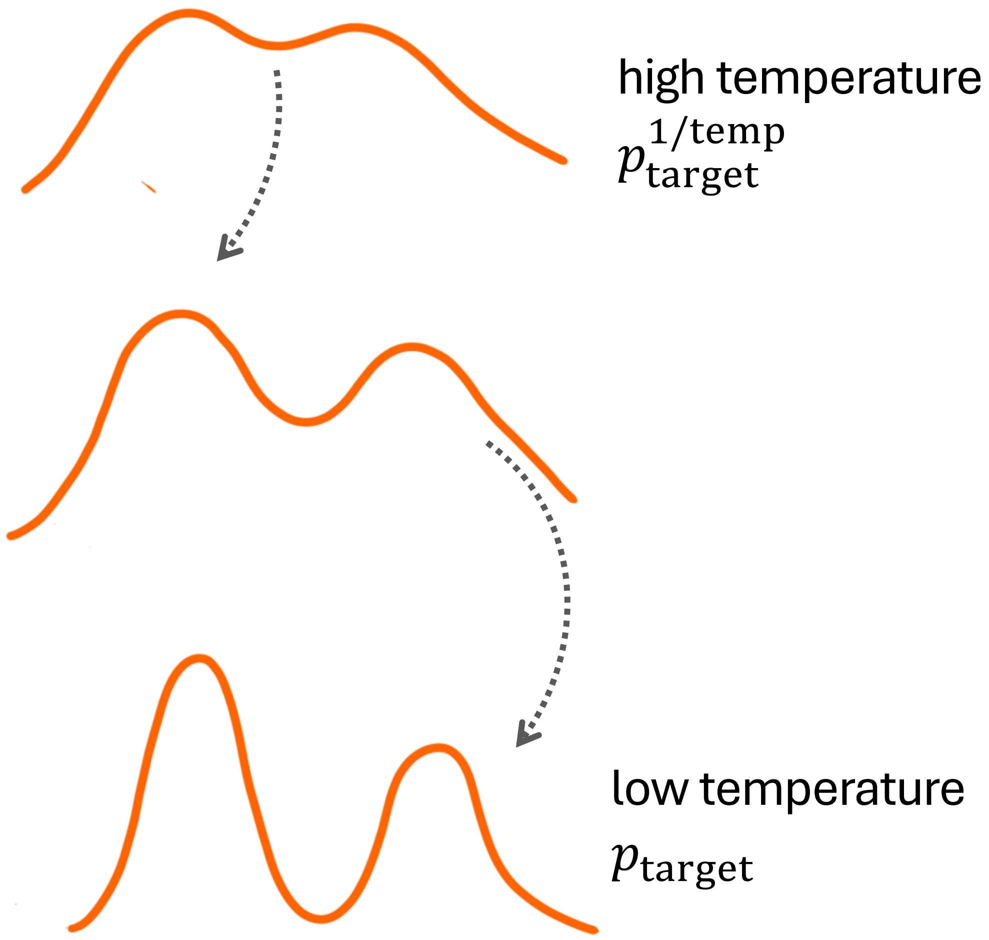
Parallel Tempering/Replica Exchange



Parallel Tempering/Replica Exchange

🥲 SOTA MCMC in MD simulation

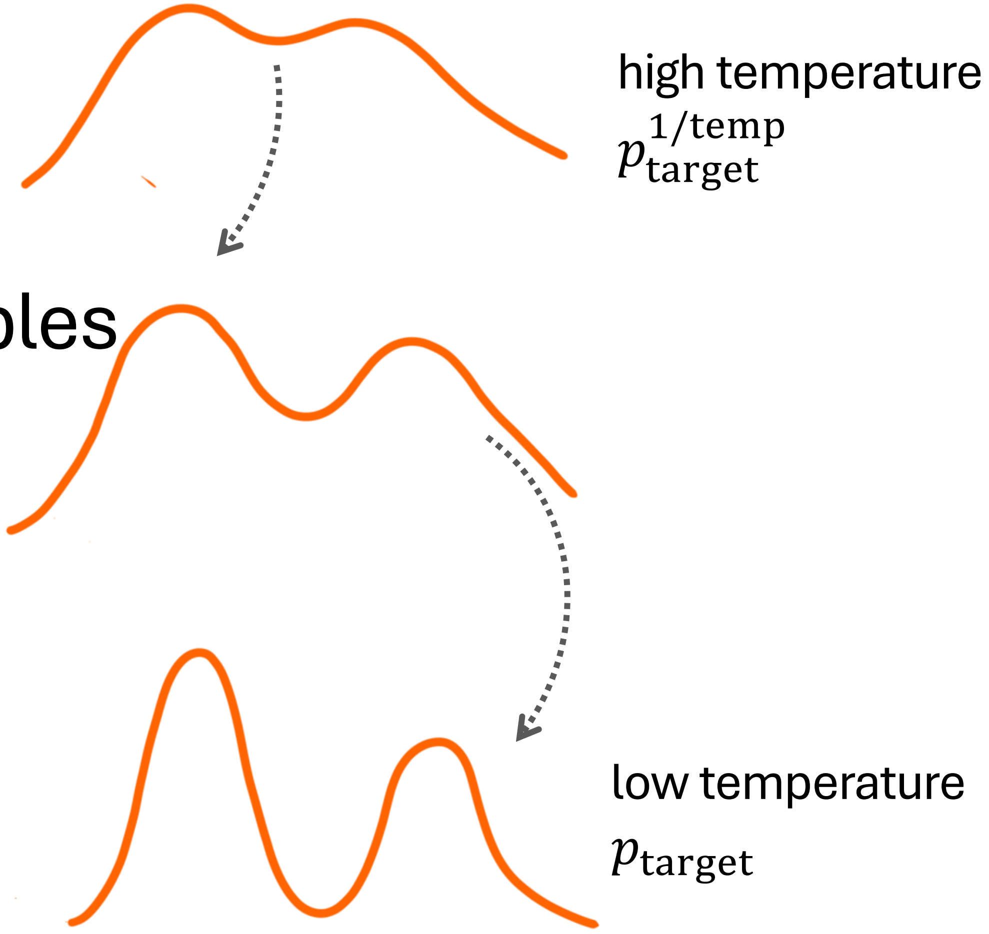
🥲 Highly parallel



Parallel Tempering/Replica Exchange

😔 Correlated samples

😔 Need more simulation for new samples

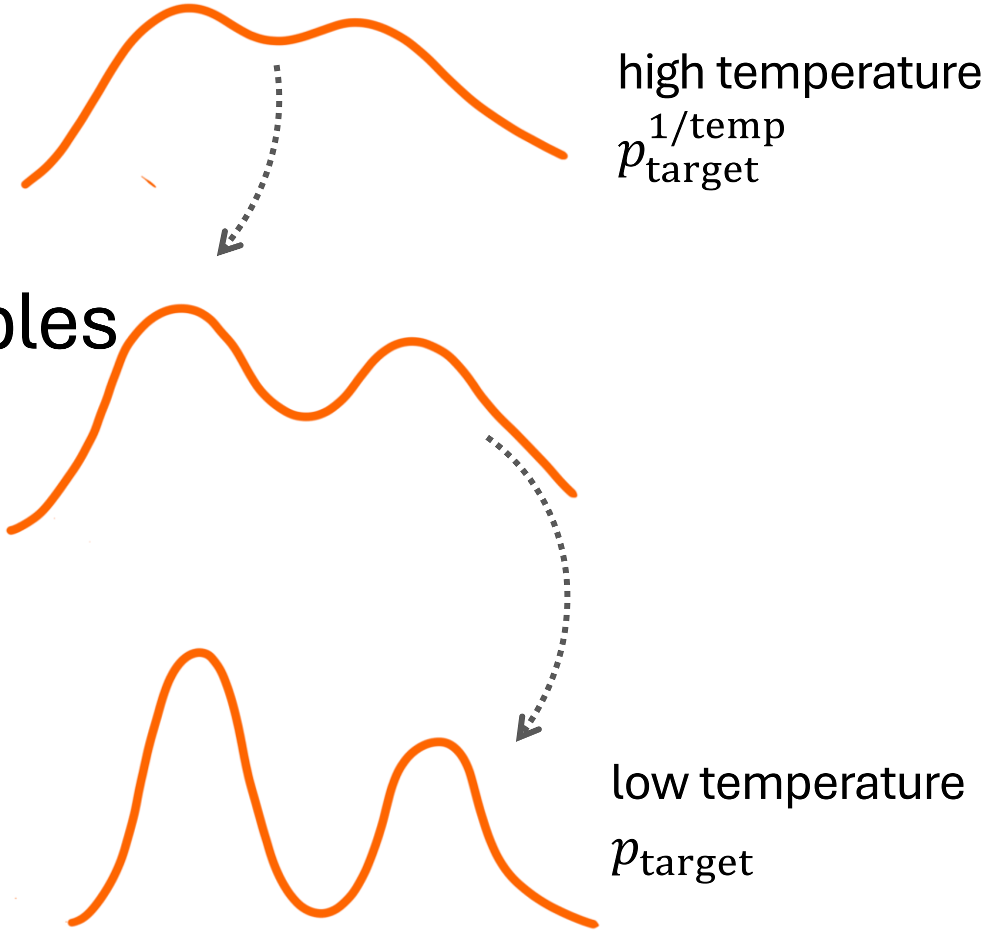


Parallel Tempering/Replica Exchange

😔 Correlated samples

😔 Need more simulation for new samples

Generative models can easily address them!
But is it worth it?



Parallel Tempering/Replica Exchange

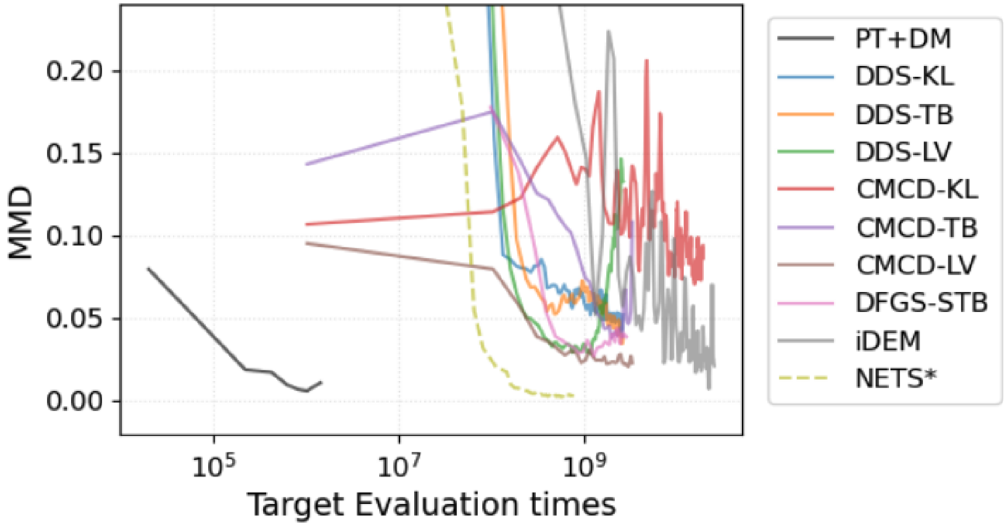
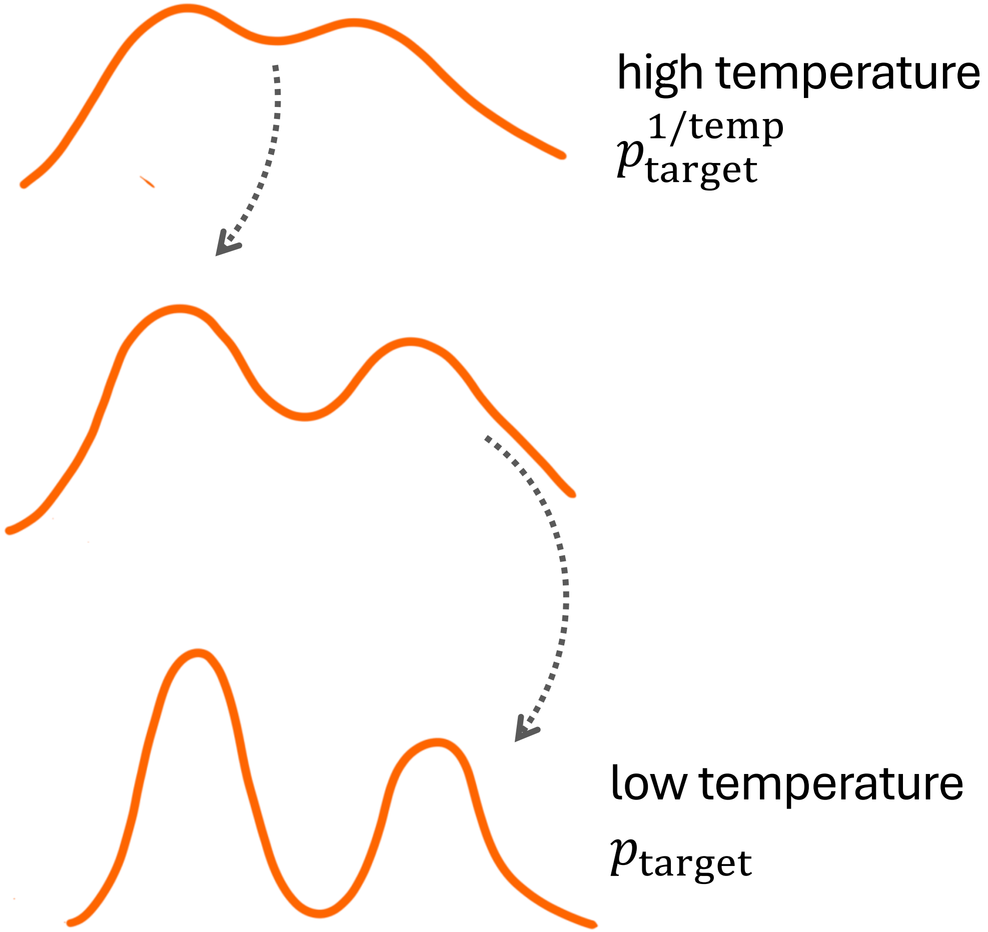


Figure 2: Sample quality vs target evaluation times for different approaches with different objectives on GMM-40 target. *NETS uses mode interpolation, which is distinct from that employed in others.



Discussion & Takeaway

1. **Langevin** term plays an importance role in neural samplers

Discussion & Takeaway

1. **Langevin** term plays an importance role in neural samplers

2. If we need Langevin gradient anyway, we need to **think more on the sample efficiency** (might need to be open to using data)

Discussion & Takeaway

1. **Langevin** term plays an importance role in neural samplers

2. If we need Langevin gradient anyway, we need to **think more on the sample efficiency** (might need to be open to using data)

3. Incorporating with / use network to improve **PT** might be a promising direction

Discussion & Takeaway

1. **Langevin** term plays an importance role in neural samplers

2. If we need Langevin gradient anyway, we need to **think more on the sample efficiency** (might need to be open to using data)

3. Incorporating with / use network to improve **PT** might be a promising direction

4. **Better prior, interpolant, explorative objectives** still needed

Thank you!

Jiajun He

https://jiajunhe98.github.io/

jh2383@cam.ac.uk