Diffusion Neural Sampler 101

Jiajun He

Cambridge MLG reading group

26/02/2025

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:

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

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

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:

$$p_{\text{prior}} \xrightarrow[\text{NF, Diffusion, etc.}]{} p_{\text{target}}$$

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

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

$\boxed{\text{And } X_t \sim Y_{T-t},}$ “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 $X_T \sim p_{\text{target}}$.

If we have a "target" process

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

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

$\boxed{\text{And } X_t \sim Y_{T-t},}$ "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 $X_T \sim p_{\text{target}}$.

If we have a "target" process

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

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

$\boxed{\text{And } X_t \sim Y_{T-t},}$ "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 $X_T \sim p_{\mathrm{target}}$.

Want a sample process (prior to target),

If we have a “target” process

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

To be the **time-reversal,**

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

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

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

Want a sample process (prior to target),

If we have a “target” process

$dY_t = \boxed{g(Y_t, t)}dt + \sigma\sqrt{2}dW_t, Y_0 \sim p_{\text{target}}$

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 $X_T \sim Y_{T-T} = Y_0$

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

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

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

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

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

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

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

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

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

Diffusion Neural samplers - idea 1.1

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

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

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

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

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

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

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

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

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

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

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

Diffusion Neural samplers - idea 1.1

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

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

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

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

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

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

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

Diffusion Neural samplers - idea 1.1

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

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

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

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

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

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

$$p_{\mathrm{target}}(\underbrace{\cancel{Y_0}}_{X_{t_N}}) N\left(\underbrace{\cancel{Y_{t_1}}}_{X_{t_{N-1}}} | \underbrace{\cancel{Y_0}}_{\ldots}\right) N\left(Y_{t_2} | Y_{t_1}\right)_{\ldots} \ldots N\left(Y_{t_N} | Y_{t_{N-1}}\right)$$

... ...

Diffusion Neural samplers - idea 1.1

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

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

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

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

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

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

Diffusion Neural samplers - idea 1.1

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

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

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

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

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

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

Diffusion Neural samplers - idea 1.1

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

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

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

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

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

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

Diffusion Neural samplers - idea 1.1

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

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

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

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

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

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

Diffusion Neural samplers - idea 1.1

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

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

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

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

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

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

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

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

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 $\tilde{p}\left(X_{0:t_N}\right)$: $\quad\Rightarrow\quad$ $\overrightarrow{\mathbf{Q}}(X), \overleftarrow{\mathbf{P}}(X)$

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

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

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

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

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

Diffusion Neural samplers - idea 1.1

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

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

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

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

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

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

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

Diffusion Neural samplers - idea 1.1

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

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

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

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

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

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

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

Diffusion Neural samplers - idea 1.1

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

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

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

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

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

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

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

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

Want a sample process (prior to target),

To be the **time-reversal,**

of a simple target process (target to prior)

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

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

How to achieve this?

matching forward and backward processes

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

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

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

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

Diffusion Neural samplers - idea 1.2

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

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

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

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

Recall in diffusion models, we learn

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

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

Diffusion Neural samplers - idea 1.2

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

Recall in diffusion models, we learn

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

What is this term?

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

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

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

What is this term?

The "score" at $T - t$

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

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

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

What is this term?

The "score" at $T - t$

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

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

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

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

What is this term?

The "score" at $T - t$

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

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

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

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

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

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

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

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

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

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

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

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

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

Gradient of Conv = Conv of gradient $= (\nabla p_{\text{target}} * N(\cdot\, | 0, v_t I))(Y_t) / \, p_t(Y_t)$

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

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

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

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

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

Diffusion Neural samplers - idea 1.2

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

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

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

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

Diffusion Neural samplers - idea 1.2

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

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

Diffusion Neural samplers - idea 1.2

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

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

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

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

Diffusion Neural samplers - idea 1.2

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Target score identity (TSI)

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

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

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

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

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 $p(Y_0|Y_t)$

Any other choices to achieve this? YEEEES!

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

Importance Sampling using q

Diffusion Neural samplers - idea 1.3

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

Recall in diffusion models, we learn

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

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

Diffusion Neural samplers - idea 1.3

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

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

Diffusion Neural samplers - idea 1.3

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

Diffusion Neural samplers - idea 1.3

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

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

Diffusion Neural samplers - idea 1.3

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

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

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

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

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

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)

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

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

Diffusion Neural samplers - idea 1.3

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

Fokker-Planck equation (in log space)

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

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

Diffusion Neural samplers - idea 1.3

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

Fokker-Planck equation (in log space)

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

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

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

Diffusion Neural samplers - idea 1.3

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

Want a sample process (prior to target),

Fokker-Planck equation (in log space)

To be the **time-reversal,**

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

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

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

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

Diffusion Neural samplers - idea 2

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

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

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

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

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

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 2.1

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

Fokker-Planck equation (in log space)

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

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

Diffusion Neural samplers - idea 2.1

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

Fokker-Planck equation (in log space)

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

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

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

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)

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

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

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

Diffusion Neural samplers - idea 2.1

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

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

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

Fokker-Planck equation (in log space)

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

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

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

Diffusion Neural samplers - idea 2.1

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

Fokker-Planck equation (in log space)

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

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

Again, do not worry on this formula

Let's focus on the high-level idea

Diffusion Neural samplers - idea 2.1

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

Fokker-Planck equation (in log space)

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

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

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,

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

$\partial_t \log p_{\text{prior}}^{\beta_t} p_{\text{target}}^{1-\beta_t} - \partial_t Z_{\pi_t}$ 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 $Z_\theta(t), f(X,t)$ and learn it by $\min\left|\left|\text{LFS}\right|\right|^2$

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,

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

aligns with known interpolants between prior and target

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

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

Diffusion Neural samplers - idea 2.2

Want a sample process (prior to target),

whose marginal density at every time step,

aligns with known interpolants between prior and target

Any other ways? YES!

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

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

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

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

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

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

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

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

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

known term

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

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

$$\mathrm{d}X_t = \underbrace{f(X_t, t)\mathrm{dt}}_{\text{Time-dependent network}} + \sigma\sqrt{2}\mathrm{d}W_t, X_0 \sim p_{\text{prior}},$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

"Nelson's Condition" is an iff condition

Diffusion Neural samplers - idea 2.2

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

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

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

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

“Nelson’s Condition” is an iff condition

Diffusion Neural samplers - idea 2.2

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

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

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

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

"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 - summary

Overall framework:

Objectives:

Diffusion Neural samplers - summary

Overall framework:

👉 Make the sampling process as the reversal of a target process

👉 Make the marginal of the sampling agree with some target interpolants

Objectives:

Diffusion Neural samplers - summary

Overall framework:

👉Make the sampling process as the reversal of a target process

👉Make the marginal of the sampling agree with some target interpolants

Objectives:

👉Write down backward and forward, align them

👉Write down the marginal, align it with the sampling process

Diffusion Neural samplers - summary

Overall framework:

- 👉 Make the sampling process as the reversal of a target process
- 👉 Make the marginal of the sampling agree with some target interpolants

You can combine them freely!

Objectives:

- 👉 Write down backward and forward, align them
- 👉 Write down the marginal, align it with the sampling process

Thank you!

Jiajun He

https://jiajunhe98.github.io/

jh2383@cam.ac.uk