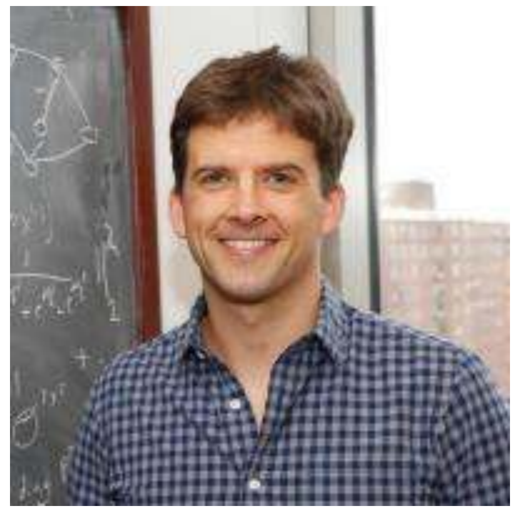
Reverse Diffusion Sequential Monte Carlo Samplers

Luhuan Wu. Flatiron Institute

Monte Carlo Online Seminar

Mar 10 2026

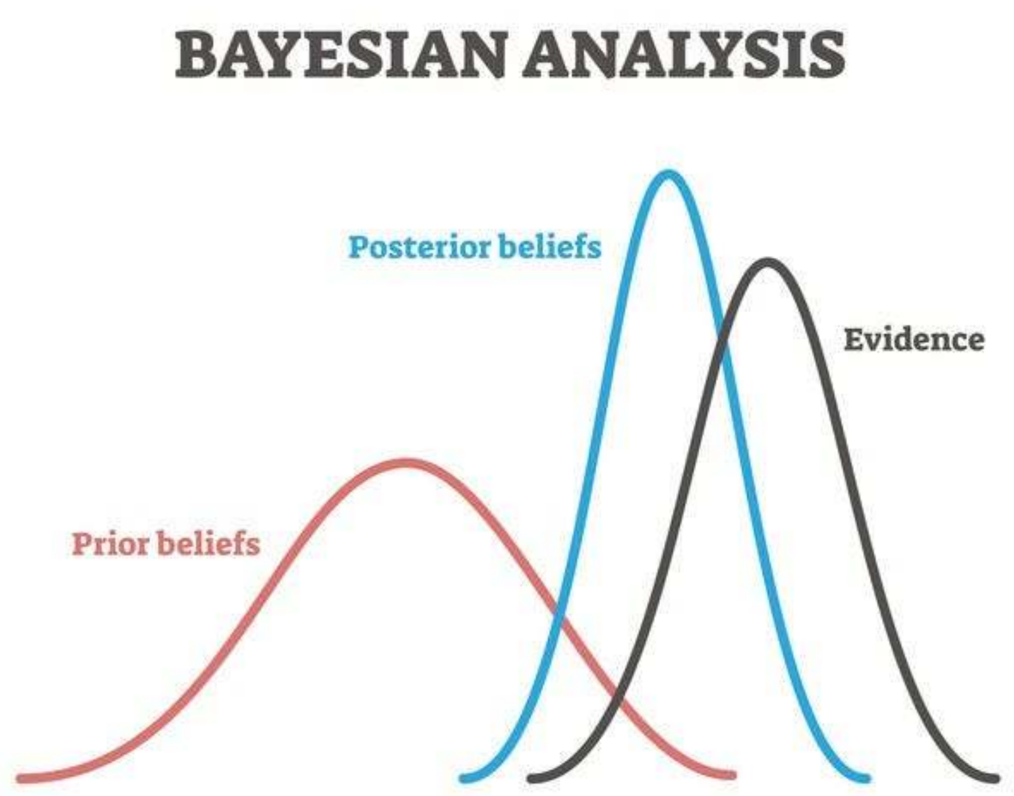


Yi Han Christian A. Naesseth John P. Cunningham

Sampling and Annealing

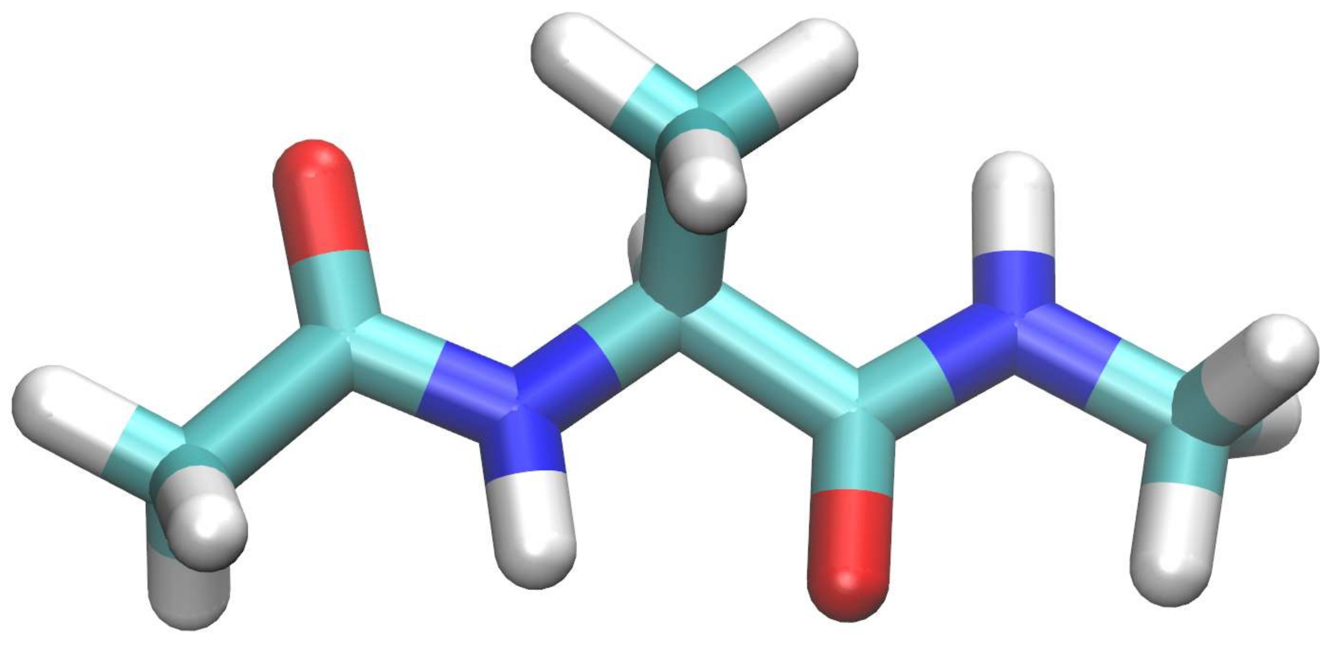
The sampling problem

Goal: sample from $\pi(x) = \dfrac{1}{Z}\tilde{\pi}(x)$, where the unnormalized $\tilde{\pi}(x)$ is accessible



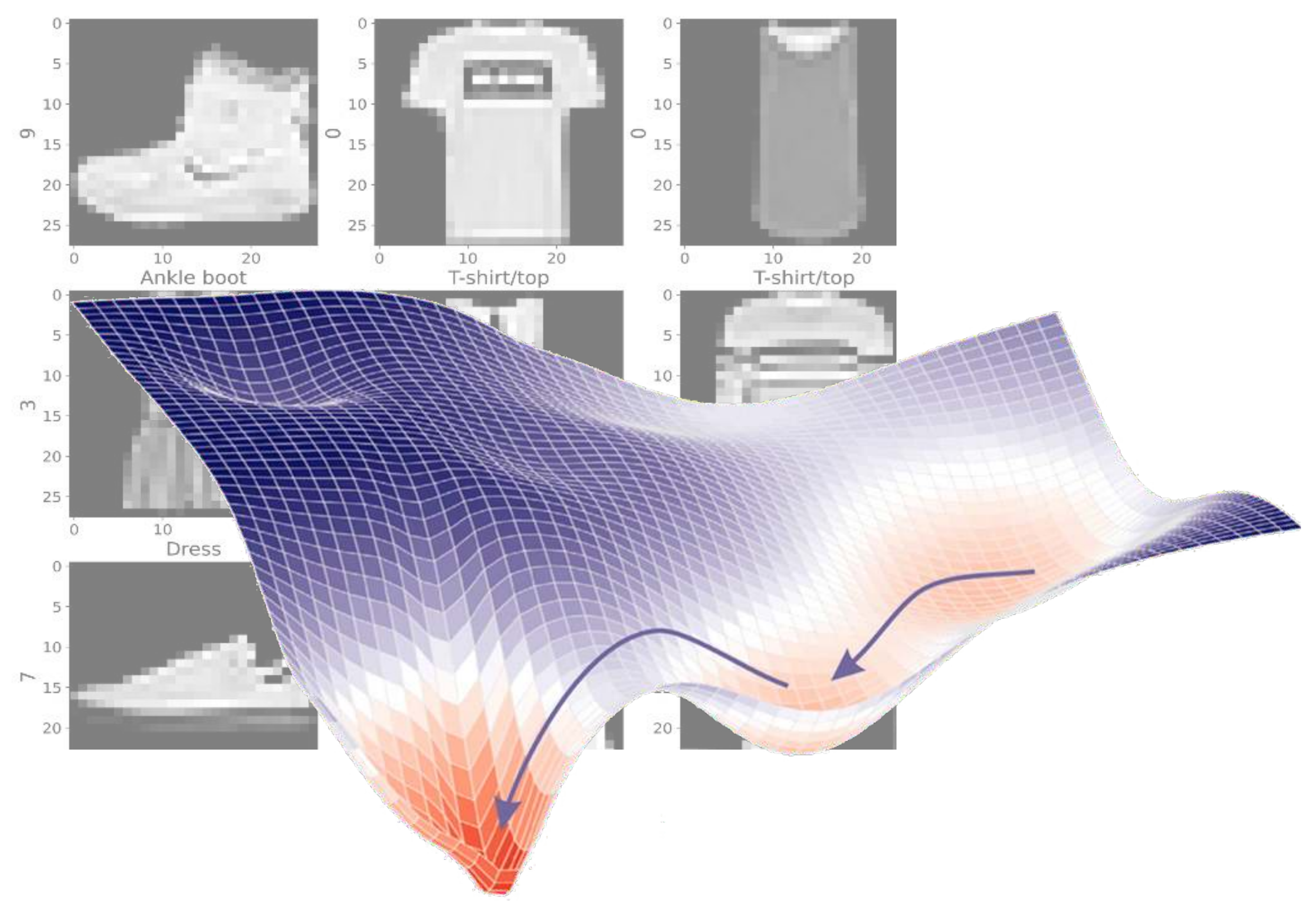
Bayesian inference

$\pi(x) \propto p(x)p(y\,|\,x)$



Computational chemistry

$\pi(x) \propto \exp\{-E(x)\}$



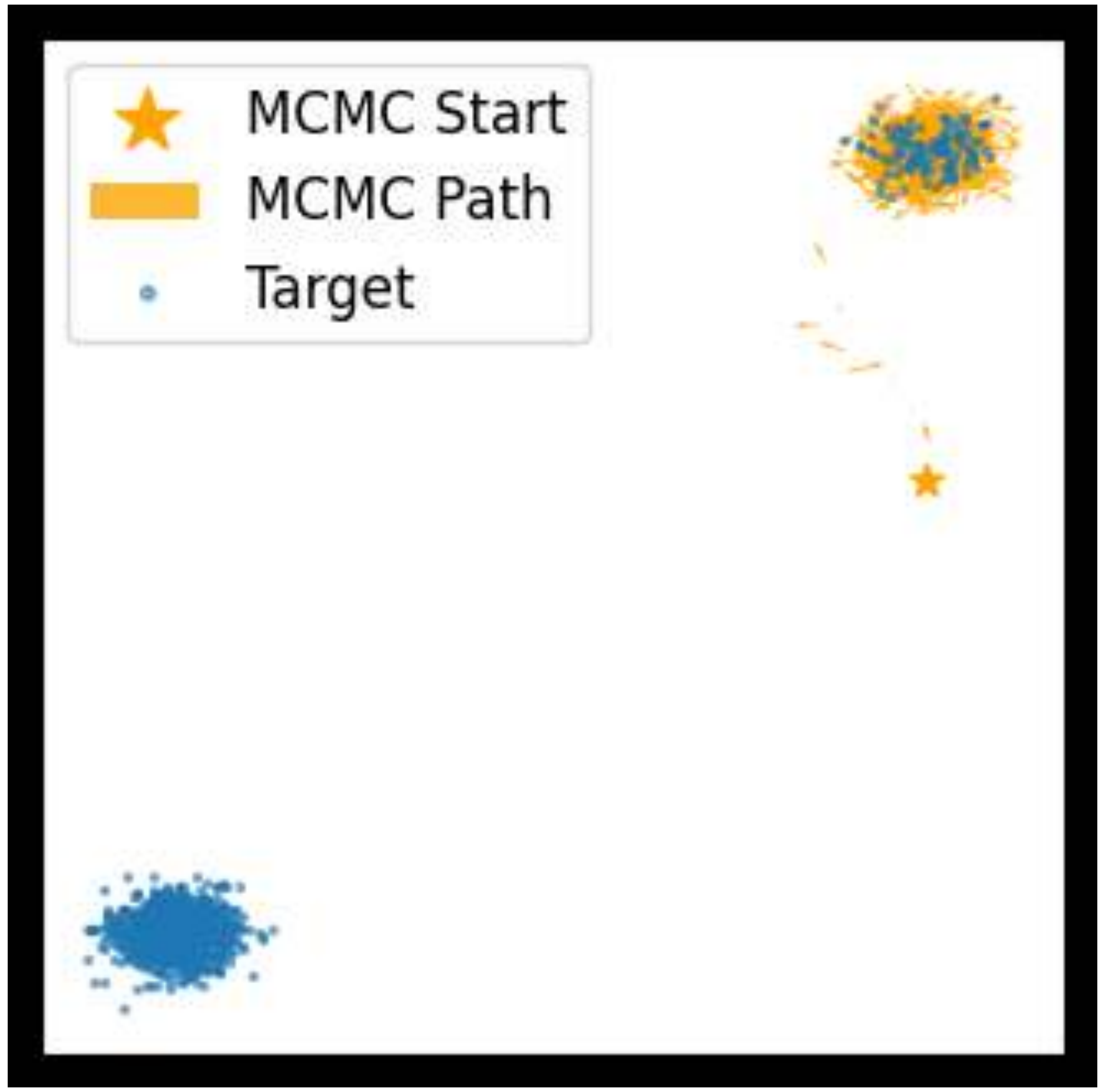
Energy based modeling

$\pi(x) \propto \exp\{-E_\theta(x)\}$

Sampling: challenges

- Multimodality
- Poor isoperimetric properties
- High dimensionality

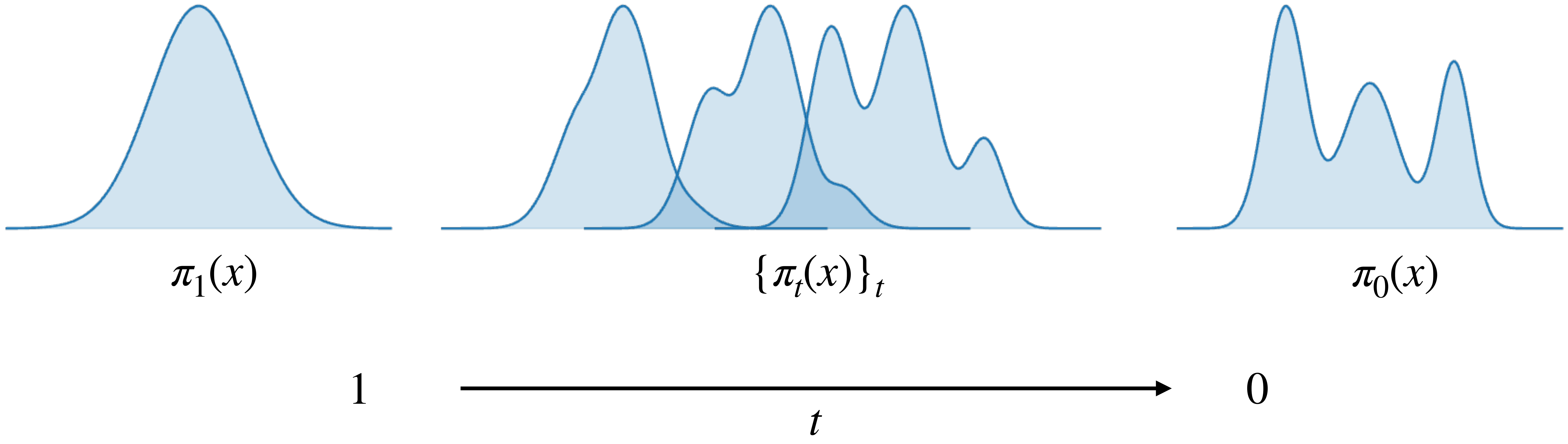
…



MCMC with local moves suffers from slow mixing and high energy barriers

Annealing

Introduce a sequence of "easier" intermediate distributions



✓ Often used in an AIS / SMC framework to mitigate biases

Geometric annealing

- Geometric interpolation between target $\pi_0(x) := \pi(x)$ and a base $\pi_1(x)$ e.g. $\mathcal{N}(0,1)$

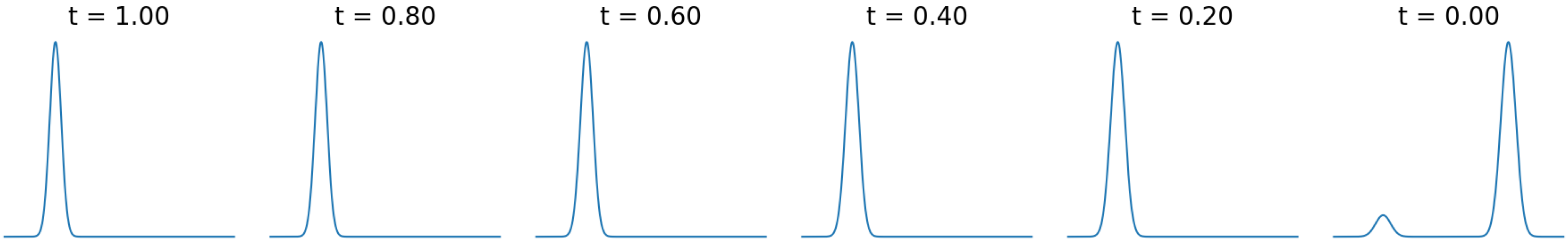
$$\pi_t(x) \propto \pi_0(x)^{1-t}\pi_1(x)^t$$

- Equivalently, linear interpolation in the log density space

$$\Leftrightarrow \log \pi_t(x) \asymp (1-t)\log \pi_0(x) + t \log \pi_1(x)$$

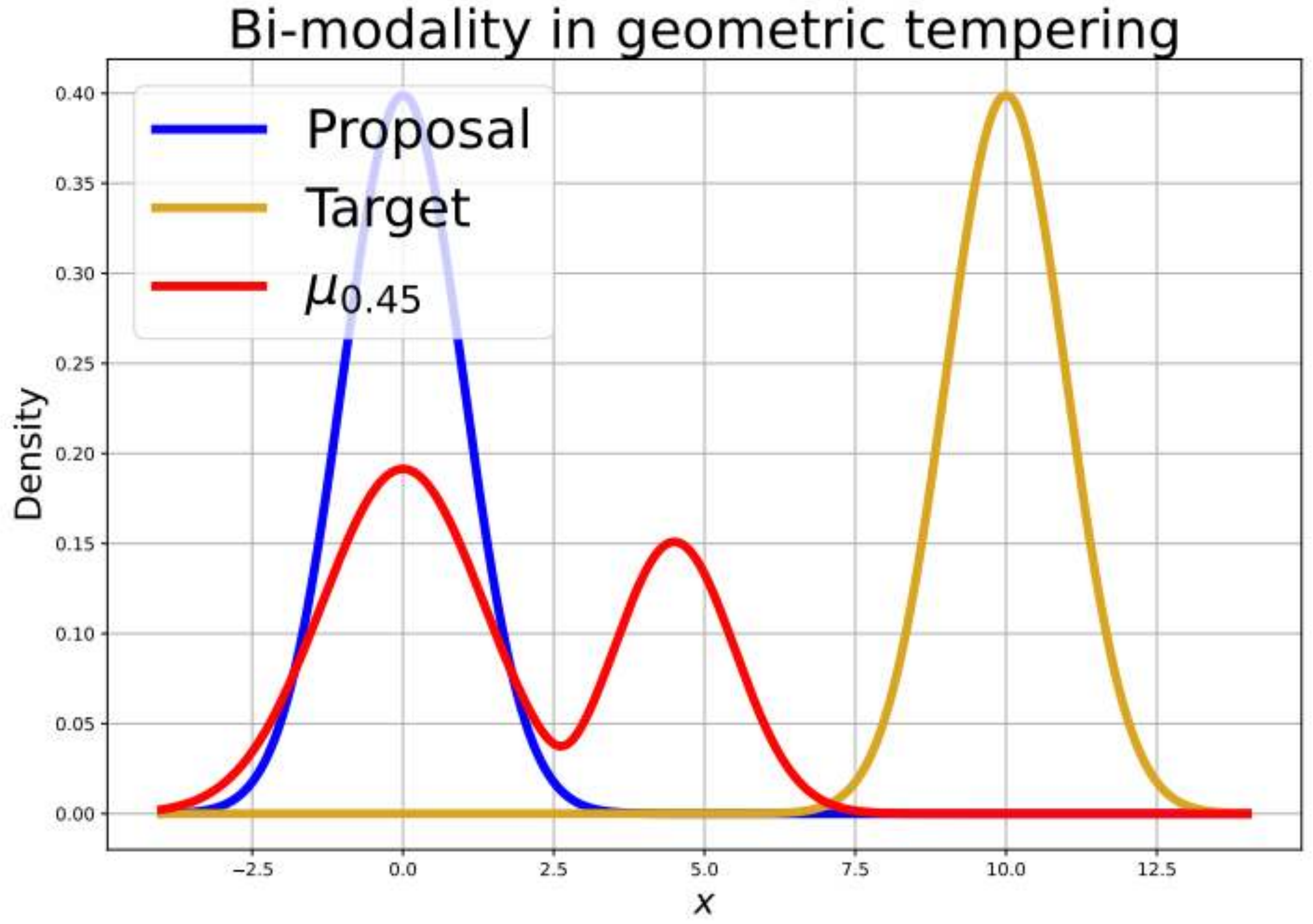
Geometric annealing: limitations

- Can be sensitive to initial base distribution and time discretization schemes
  - Example: phase transitions in the end



$\pi_1(x) = \mathcal{N}(0,1)$ $\pi_0(x)$

- Spurious modes



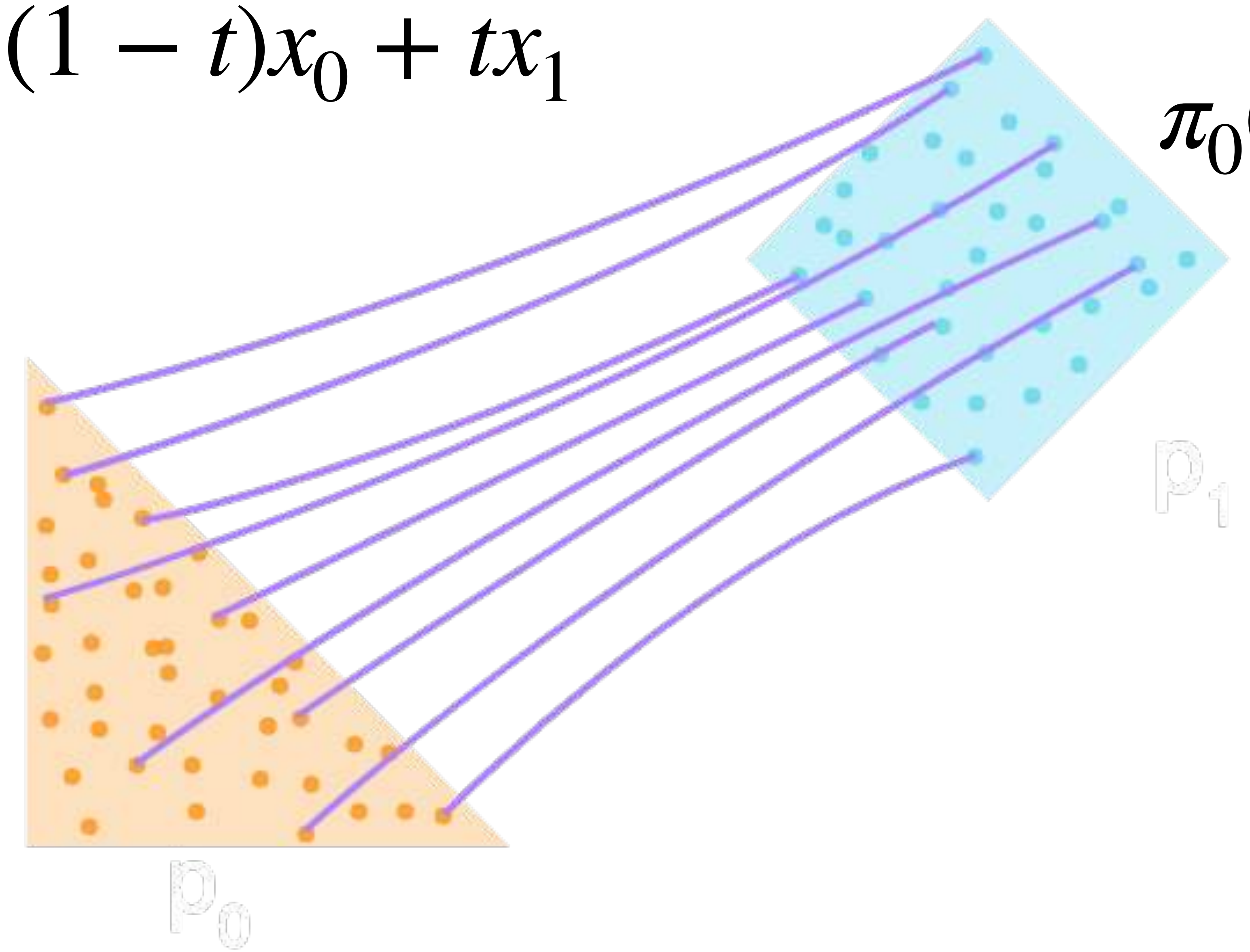
[Chehab et al 2024]

A New Annealing Strategy via Diffusion Paths

This work: diffusion-based annealing

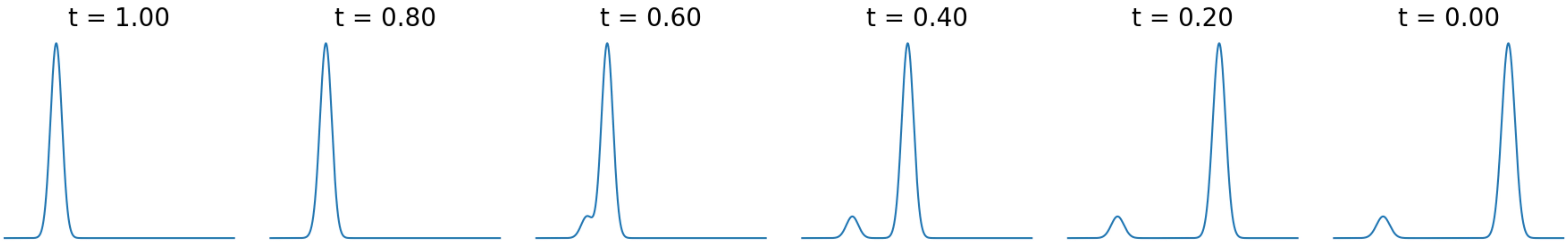
Sample space interpolation, e.g. $x_t = (1-t)x_0 + tx_1$

$\pi_0(x)$



$\pi_1(x) = \mathcal{N}(0,1)$

- Gradually transporting probability mass across modes



Outline

- **Diffusion model background**

  ➡ Defines a annealing path for sampling

- **Generic diffusion Monte Carlo samplers**

  ➡ Practical, but biased algorithms

- ***RDSMC*: reverse diffusion sequential Monte Carlo samplers**

  ➡ Asymptotic exact guarantees almost “for free”

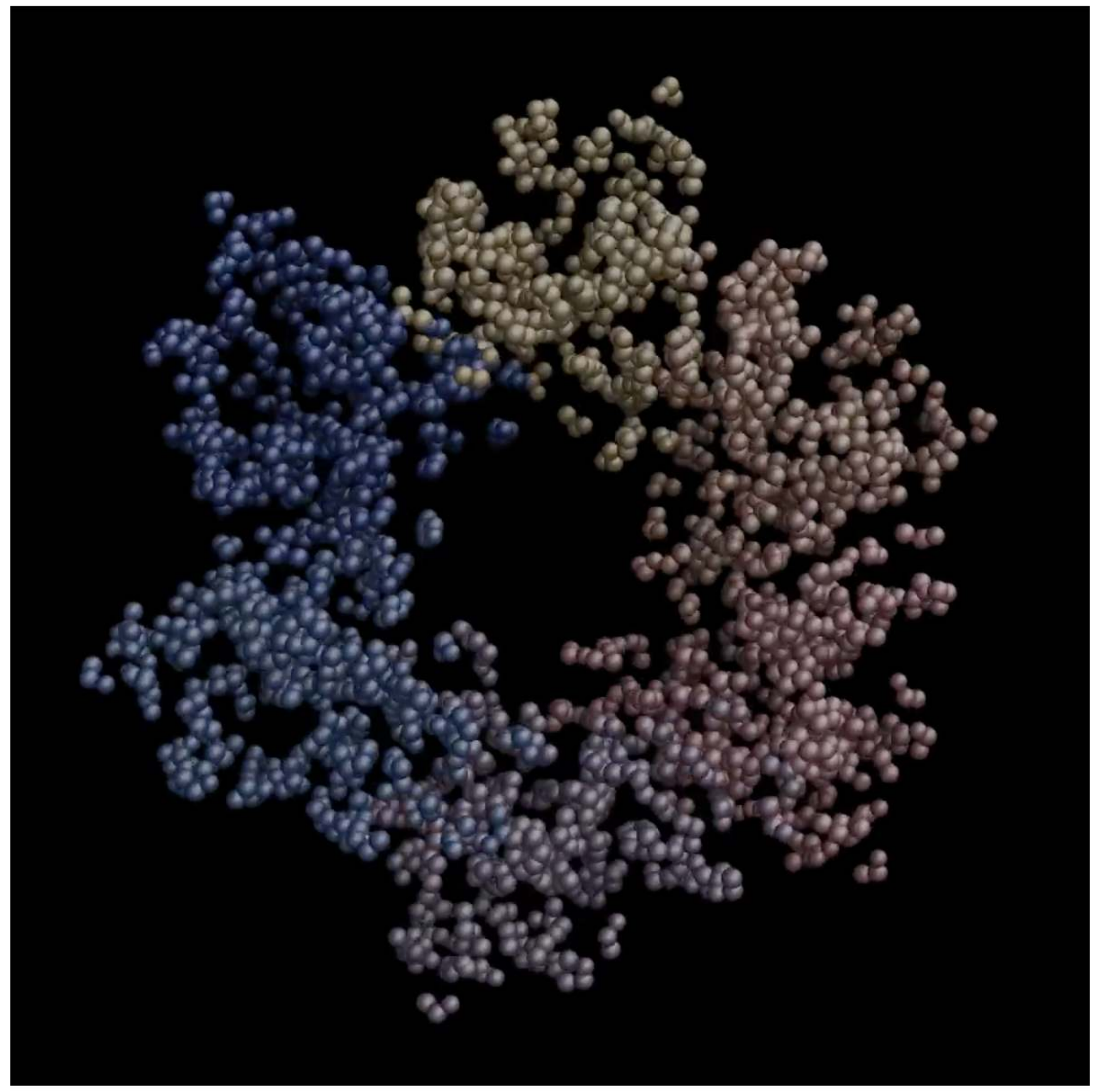
Diffusion model background

Diffusion models

A.I. Turns Its Artistry to Creating New Human Proteins

Inspired by digital art generators like DALL-E, biologists are building artificial intelligences that can fight cancer, flu and Covid.

Share full article



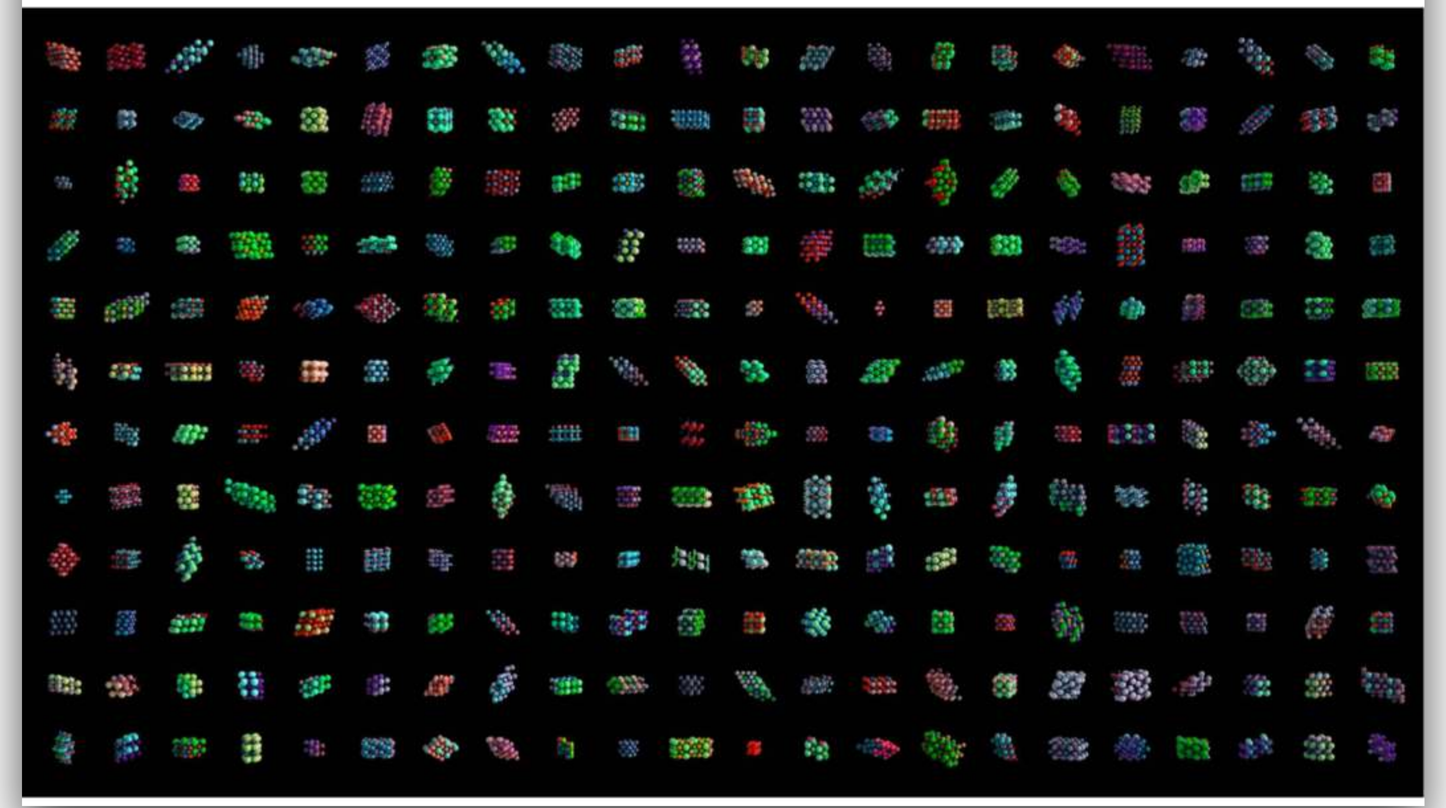
An example of an animated diffusion model of A.I.-generated proteins. Video by Ian C. Haydon/University of Washington Institute for Protein Design

MatterGen: A new paradigm of materials design with generative AI

Published January 16, 2025

By Claudio Zeni, Senior Researcher; Robert Pinsler, Senior Researcher; Daniel Zügner, Senior Researcher; Andrew Fowler, Senior Researcher; Matthew Horton, Senior Research SDE; Ryota Tomioka, Senior Principal Research Manager; Tian Xie, Principal Research Manager

Share this page



[e.g. Ho et al 2020, Song et al 2020]

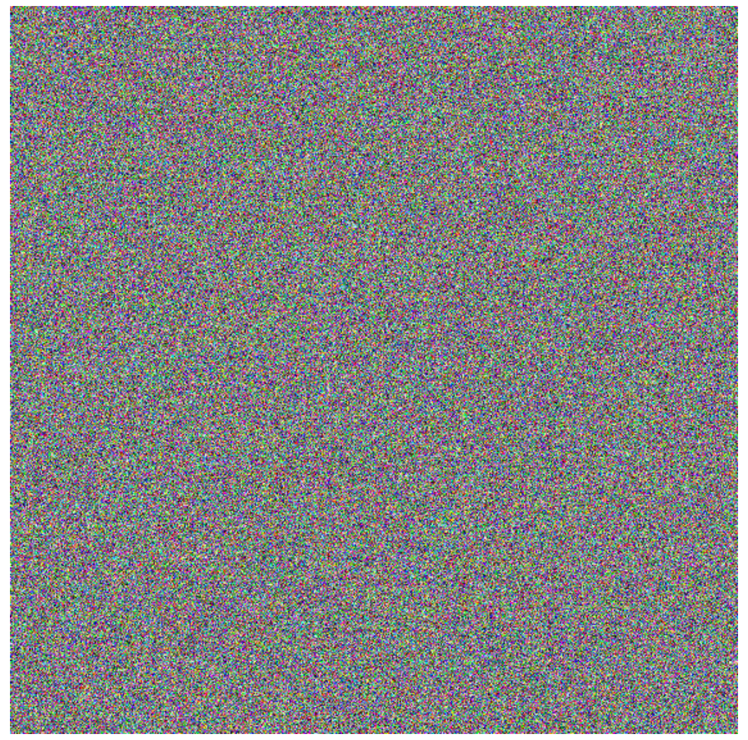
Forward process

$$dx_t = f_t x_t dt + g_t dB_t$$

$\pi_0(x)$ $\pi_t(x_t)$ $\pi_1(x)$

$\cdots$

$\cdots$

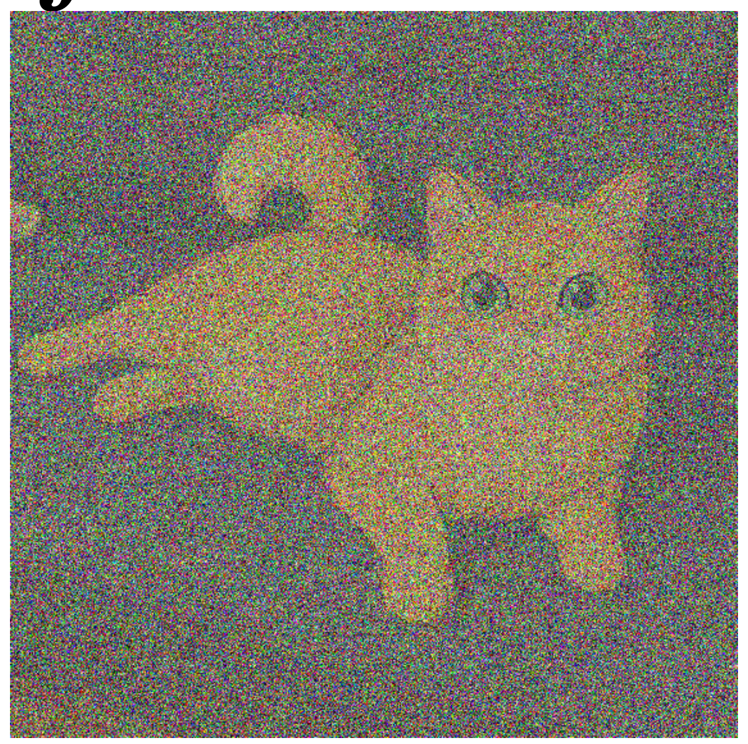


Forward process: defines an annealing path $\{\pi_t\}_t$

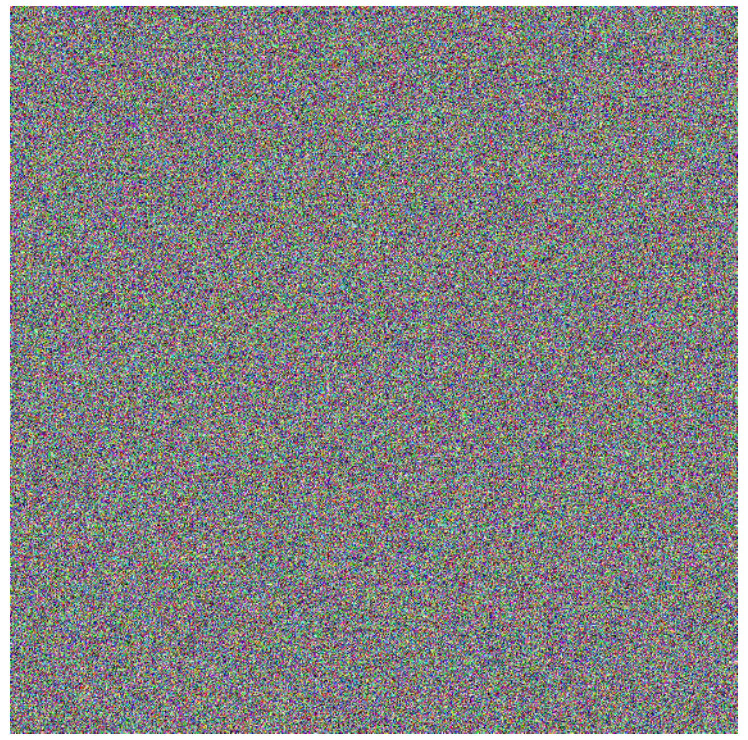
$$dx_t = f_t x_t dt + g_t dB_t$$

$\pi_0(x)$ $\qquad$ $\pi_t(x_t) = \int \pi_0(x_0)\mathcal{N}(x_t \mid \alpha_t x_0, \sigma_t^2)dx_0$ $\qquad$ $\pi_1(x)$

$\cdots$



$\cdots$



Sample-space view: $x_t = \alpha_t x_0 + \sigma_t x_1$

$\{\alpha_t, \sigma_t\}$:**signal and noise parameters determined by** $\{f_t, g_t\}$

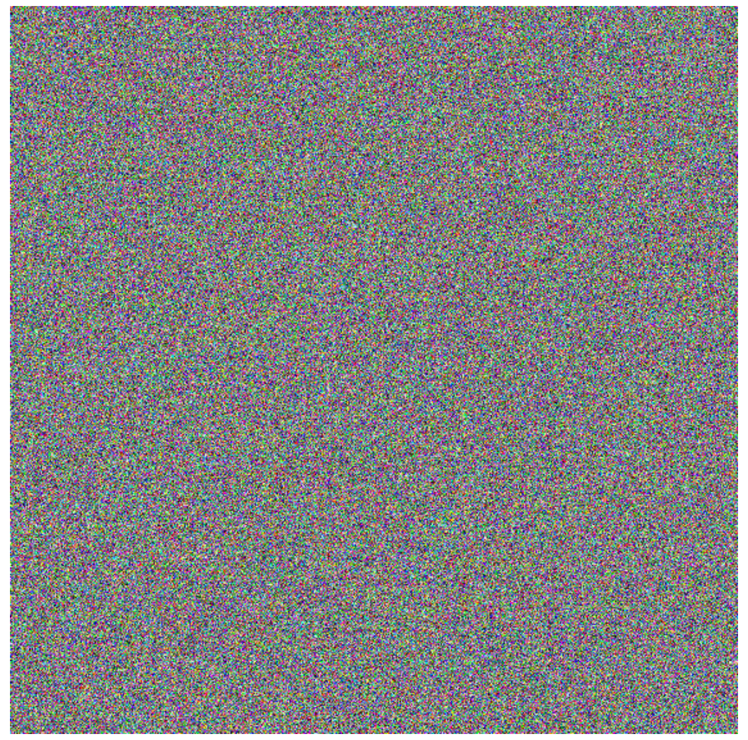
Reverse process: simulates the annealing path

$$dx_t = f_t x_t dt + g_t dB_t$$

$\pi_0(x)$ $\pi_t(x_t)$ $\pi_1(x)$

$\cdots$

$\cdots$



$$dx_t = [f_t x_t - g_t^2 \underbrace{\nabla_{x_t} \log \pi_t(x_t)}_{\text{the “score function”}}]dt + g_t dB_t$$

Diffusion annealed sampler (conceptually)

$$\pi_0(x) \longleftarrow \pi_1(x)$$

$$dx_t = [f_t x_t - g_t^2 \nabla_{x_t} \log \pi_t(x_t)]dt + g_t dB_t$$

Diffusion annealed sampler (conceptually)

$$\pi_0(x) \longleftarrow \pi_1(x)$$

$$dx_t = [f_t x_t - g_t^2 \nabla_{x_t} \log \pi_t(x_t)]dt + g_t dB_t$$

1. Sample from base $x_1 \sim \pi_1(x) = \mathcal{N}(0,1)$
2. Simulate reverse SDE initialized at x_1
3. Return x_0, and $x_0 \sim \pi_0(x_0)$

Diffusion annealed sampler: challenge

$$\pi_0(x) \longleftarrow \pi_1(x)$$

$$dx_t = [f_t x_t - g_t^2 \boxed{\nabla_{x_t} \log \pi_t(x_t)}]dt + g_t dB_t$$

Score is intractable. Recall $\pi_t(x_t) = \int \pi_0(x_0)\mathcal{N}(x_t \mid \alpha_t x_0, \sigma_t^2)dx_0$

with $\pi(x_0)$ known up to a constant

1. Sample from base $x_1 \sim \pi_1(x) = \mathcal{N}(0,1)$
2. Simulate reverse SDE initialized at x_1
3. Return x_0, and $x_0 \sim \pi_0(x_0)$

Diffusion Monte Carlo Sampler

Monte Carlo (MC) score estimation

Tweedie's formula

Given $x_0 \sim \pi_0(x_0), x_t \mid x_0 \sim \mathcal{N}(x_t \mid \alpha_t x_0, \sigma_t^2)$, we have

$$\nabla_{x_t} \log \pi_t(x_t) = \frac{\alpha_t \mathbb{E}[x_0 \mid x_t] - x_t}{\sigma_t^2}$$

where $\pi_t(x_t) = \int \pi_0(x_0) \mathcal{N}(x_t \mid \alpha_t x_0, \sigma_t^2) dx_0$ is the marginal,
and $\mathbb{E}[x_0 \mid x_t] = \int x_0 \pi(x_0 \mid x_t) dx_0$ is the posterior mean.

Score estimation $\longleftrightarrow$ **Posterior mean estimation**

Monte Carlo (MC) score estimation

Tweedie's formula $$\nabla_{x_t} \log \pi_t(x_t) = \frac{\alpha_t \mathbb{E}[x_0 \mid x_t] - x_t}{\sigma_t^2}$$

- Draw Monte Carlo (MC) samples from posterior $\pi(x_0 \mid x_t) \propto \tilde{\pi}(x_0)\mathcal{N}(x_t \mid \alpha_t x_0, \sigma_t^2)$

Example: Importance Sampling

1. Draw M importance samples $x_0^{(1:M)} \sim q(\cdot \mid x_t) := \mathcal{N}(\cdot \mid x_t/\alpha_t, \sigma_t^2/\alpha_t^2)$
2. Compute importance weights $w^{(m)} \leftarrow \dfrac{\tilde{\pi}(x_0^{(m)})\mathcal{N}(x_t \mid \alpha_t x_0^{(m)}, \sigma_t^2)}{q(x_0^{(m)} \mid x_t)}$

- Estimate posterior mean $\hat{x}_0 \approx \mathbb{E}_\pi[x_0 \mid x_t]$

$$\hat{x}_0 \leftarrow \sum_{m=1}^{M} w^{(m)} x_0^{(m)} / \sum_{m=1}^{M} w^{(m)}$$

- Estimate score $\hat{s}(x_t) := \dfrac{\alpha_t \hat{x}_0 - x_t}{\sigma_t^2}$ by Tweedie's

Monte Carlo (MC) score estimation

Tweedie's formula $\nabla_{x_t} \log \pi_t(x_t) = \dfrac{\alpha_t \mathbb{E}[x_0 \mid x_t] - x_t}{\sigma_t^2}$

↳ Draw Monte Carlo (MC) samples from posterior $\pi(x_0 \mid x_t) \propto \tilde{\pi}(x_0)\mathcal{N}(x_t \mid \alpha_t x_0, \sigma_t^2)$

Alternative MC methods to Importance Sampling:

AIS, SMC, rejection sampling, MCMC

[e.g. Huang et al., ICLR 2024; Grenioux et al., ICML 2024; He et al., NeurIPS 2024]

↳ Estimate posterior mean $\hat{x}_0 \approx \mathbb{E}_\pi[x_0 \mid x_t]$

↳ Estimate score $\hat{s}(x_t) := \dfrac{\alpha_t \hat{x}_0 - x_t}{\sigma_t^2}$ by Tweedie's

Generic recipe for diffusion MC sampler

$$\pi_0(x) \longleftarrow \pi_1(x)$$

$$dx_t = [f_t x_t - g_t^2 \boxed{\nabla_{x_t} \log \pi_t(x_t)}]dt + g_t dB_t$$

Score is intractable. ✅ Use MC estimate.

1. Sample from base $x_1 \sim \pi_1(x) = \mathcal{N}(0,1)$
2. For $n = N, N-1, \cdots, 1$ $(t_N = 1, t_0 = 0)$
   - Run MC targeting $\pi(x_0 \mid x_{t_n})$: estimate score $\hat{s}(x_{t_n})$
   - Sample $x_{t_{n-1}} \mid x_{t_n} \sim \mathcal{N}(x_{t_{n-1}} \mid [x_{t_n} - (f_{t_n} x_{t_n} - g_{t_n}^2 \hat{s}(x_{t_n})](t_n - t_{n-1}), g_{t_n}^2 (t_n - t_{n-1}))$
3. Return x_0, and $x_0 \sim \pi_0(x_0)$ approximately

Generic recipe for diffusion MC sampler

$$\pi_0(x) \longleftarrow \pi_1(x)$$

$$dx_t = [f_t x_t - g_t^2 \boxed{\nabla_{x_t} \log \pi_t(x_t)}]dt + g_t dB_t$$

Score is intractable. ✅ Use MC estimate.

1. Sample from base $x_1 \sim \pi_1(x) = \mathcal{N}(0,1)$
2. For $n = N, N-1, \cdots, 1$ $(t_N = 1, t_0 = 0)$

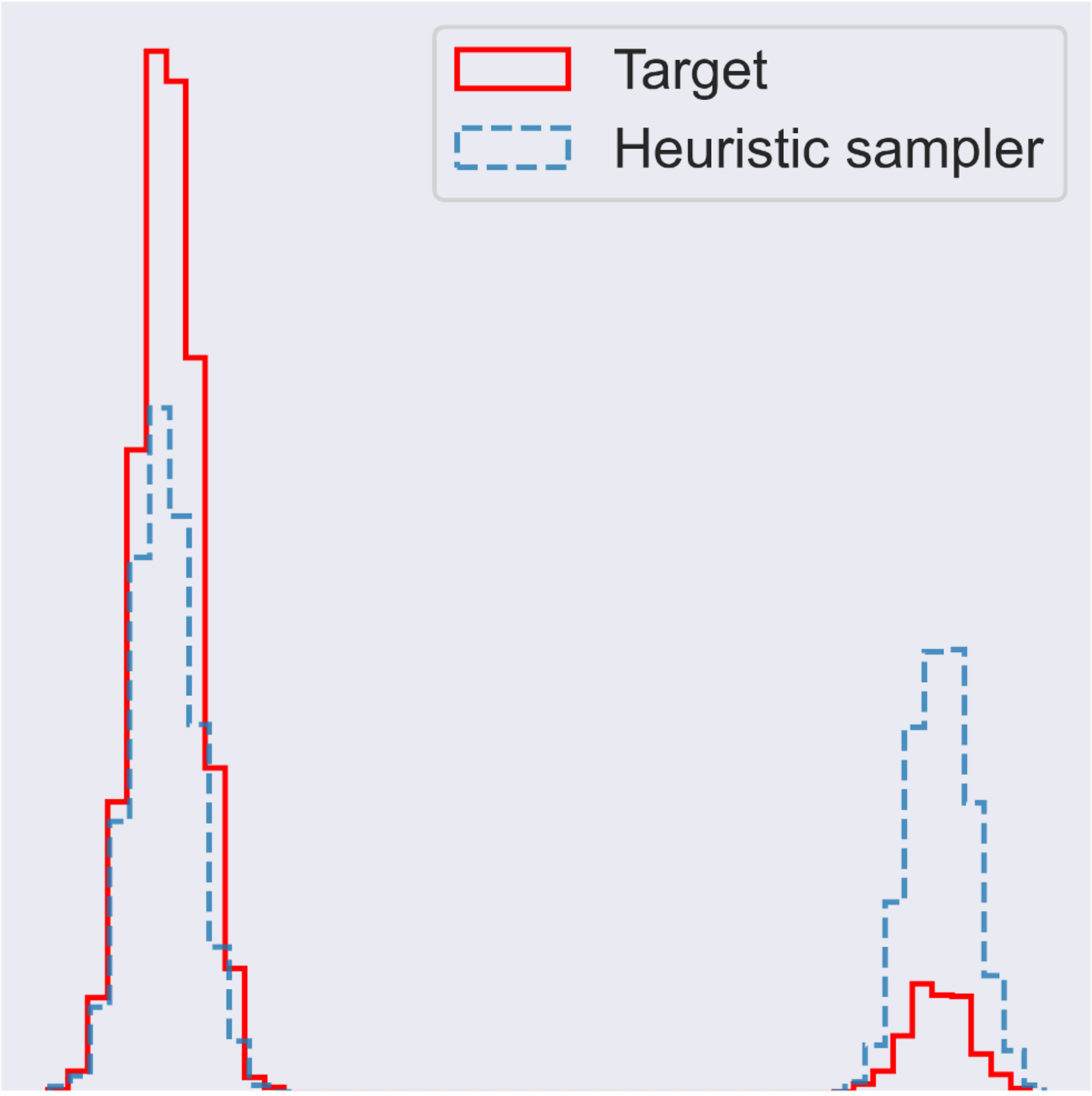
   score estimation bias

   - Run MC targeting $\pi(x_0 \mid x_{t_n})$: $\boxed{\text{estimate score } \hat{s}(x_{t_n})}$

   SDE discretization bias

   - Sample $\boxed{x_{t_{n-1}} \mid x_{t_n} \sim \mathcal{N}(x_{t_{n-1}} \mid [x_{t_n} - (f_{t_n} x_{t_n} - g_{t_n}^2 \hat{s}(x_{t_n})](t_n - t_{n-1}), g_{t_n}^2 (t_n - t_{n-1}))}$
3. Return x_0, and $x_0 \sim \pi_0(x_0)$ approximately

Sampling bias illustration



✅ Coverage of both modes

✖ Mode weights are mis-calibrated

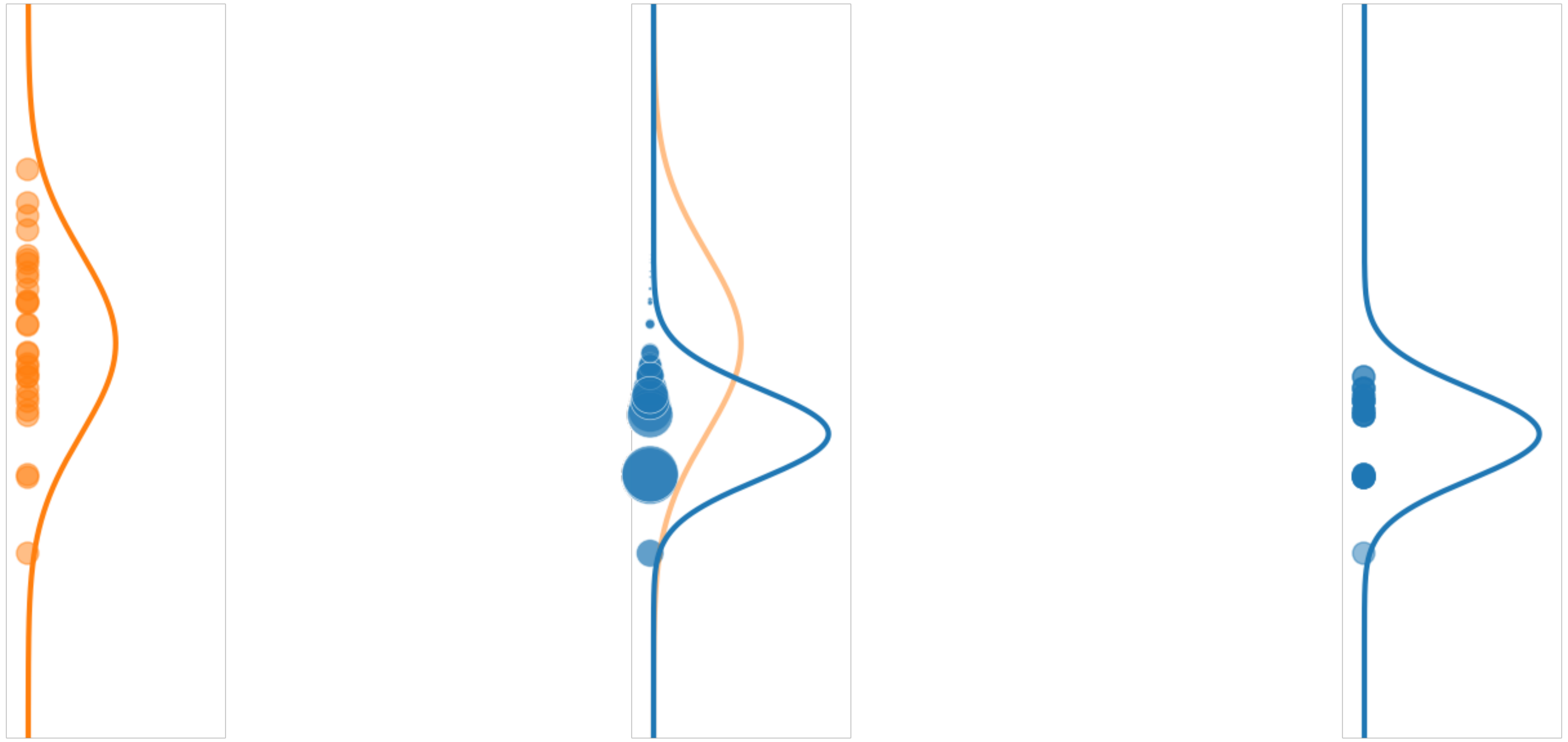
⚠️ Sampling bias

- score estimation
- SDE discretization

💡 *Next, use sequential Monte Carlo to debias.*

RDSMC: Reverse Diffusion Sequential Monte Carlo

Sequential Monte Carlo (SMC)



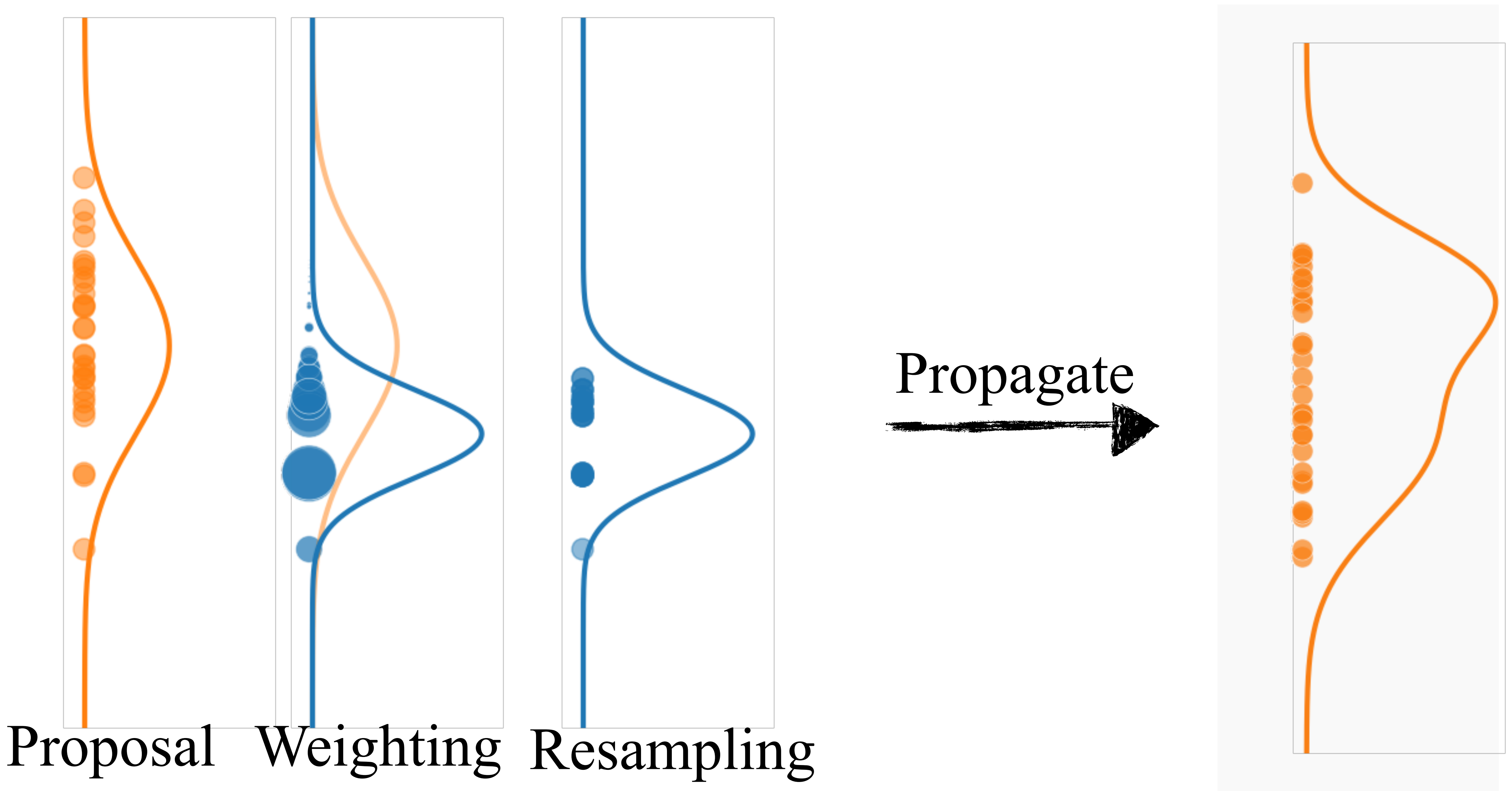
Proposal

Weighting

Resampling

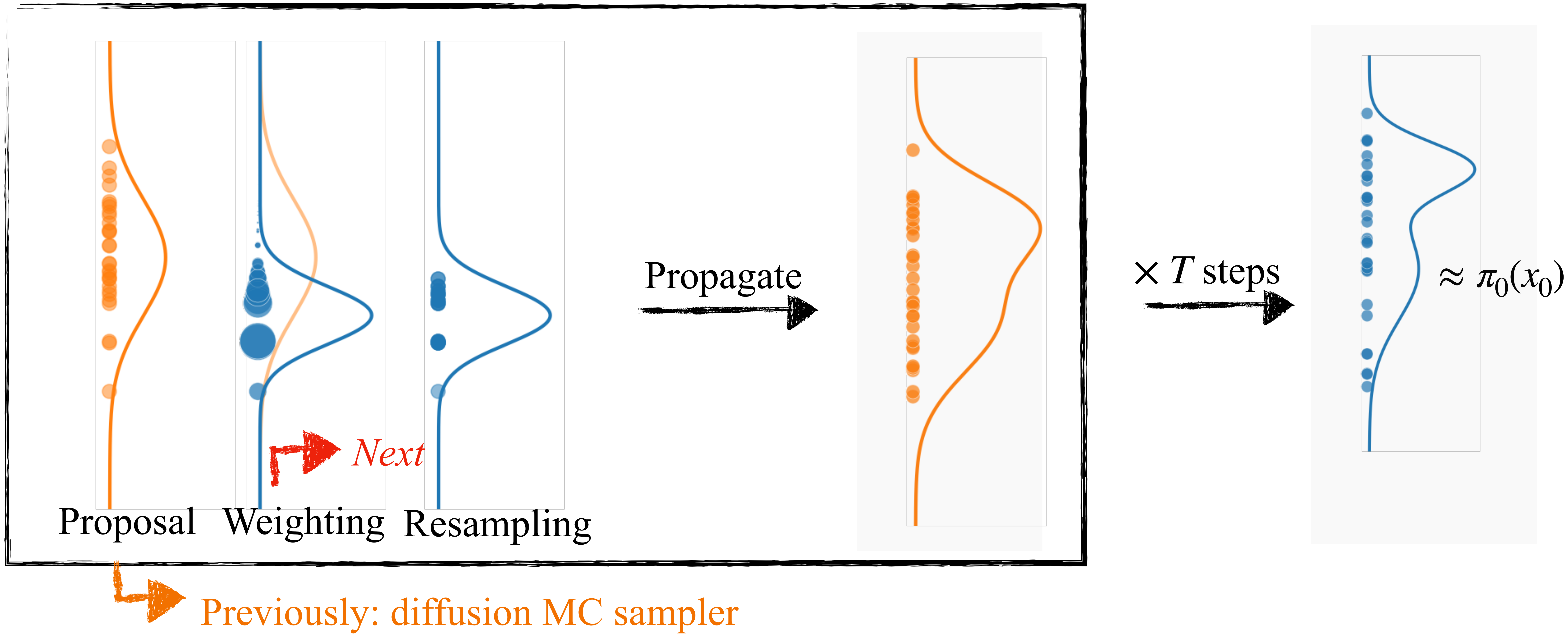
[e.g. Del Moral et al., JRSSB, 2006]

Sequential Monte Carlo (SMC)



[e.g. Del Moral et al., JRSSB, 2006]

Sequential Monte Carlo (SMC)



Key property of SMC:

The particle approximation is asymptotically exact as # particles $\to \infty$

[e.g. Del Moral et al., JRSSB, 2006]

Designing weighting functions

💡 Weighting: calibrate the discrepancy between *proposal* and *exact reverse transition*

(transition in diffusion MC sampler)

$$w_{t_n} = \frac{\pi(x_{t_n} \mid x_{t_{n+1}})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

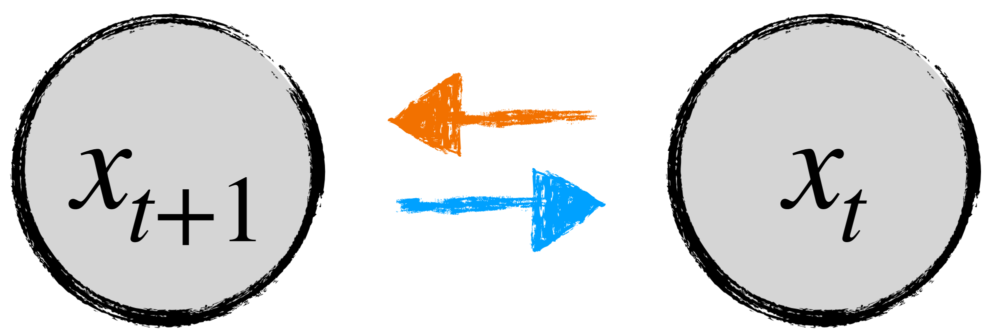
[WHNC, NeurIPS, 2025]

Designing weighting functions

💡 Weighting: calibrate the discrepancy between proposal and exact reverse transition

$$w_{t_n} = \frac{\pi(x_{t_n} \mid x_{t_{n+1}})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

$$\pi_{t_{n+1}}(x_{t_{n+1}})\pi(x_{t_n} \mid x_{t_{n+1}}) = \pi_{t_n}(x_{t_n})\pi(x_{t_{n+1}} \mid x_{t_n})$$



[WHNC, NeurIPS, 2025]

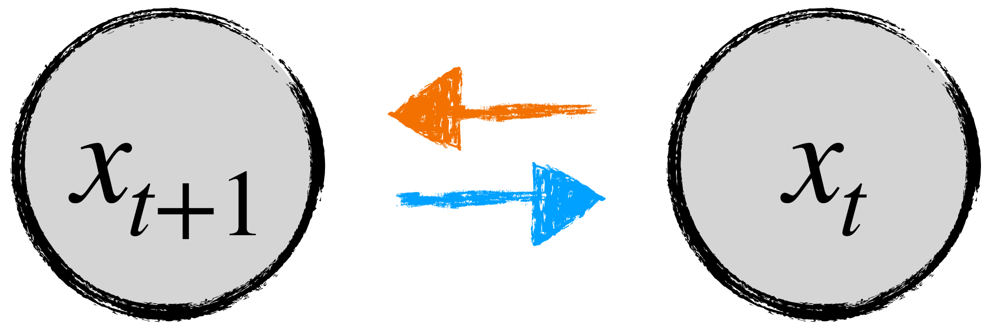
Designing weighting functions

💡 Weighting: calibrate the discrepancy between proposal and exact reverse transition

$$w_{t_n} = \frac{\pi(x_{t_n} \mid x_{t_{n+1}})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

$$= \frac{\pi_{t_n}(x_{t_n})}{\pi_{t_{n+1}}(x_{t_{n+1}})} \cdot \frac{\pi(x_{t_{n+1}} \mid x_{t_n})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

$$\pi_{t_{n+1}}(x_{t_{n+1}})\pi(x_{t_n} \mid x_{t_{n+1}}) = \pi_{t_n}(x_{t_n})\pi(x_{t_{n+1}} \mid x_{t_n})$$



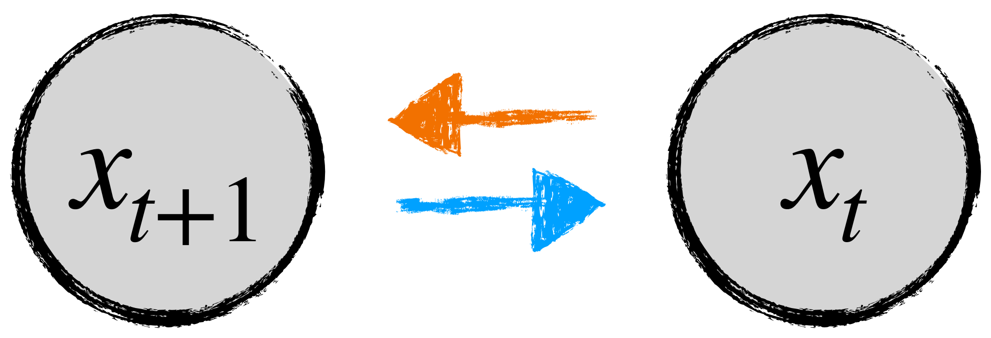
replaced with tractable forward transition

[WHNC, NeurIPS, 2025]

Designing weighting functions

💡 Weighting: calibrate the discrepancy between proposal and exact reverse transition

$$\pi_{t_{n+1}}(x_{t_{n+1}})\pi(x_{t_n} \mid x_{t_{n+1}}) = \pi_{t_n}(x_{t_n})\pi(x_{t_{n+1}} \mid x_{t_n})$$



$$w_{t_n} = \frac{\pi(x_{t_n} \mid x_{t_{n+1}})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

$$= \boxed{\frac{\pi_{t_n}(x_{t_n})}{\pi_{t_{n+1}}(x_{t_{n+1}})}} \cdot \frac{\pi(x_{t_{n+1}} \mid x_{t_n})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

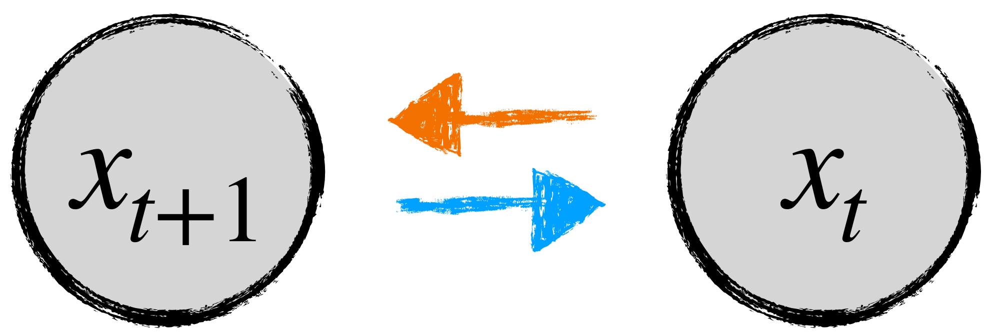
replaced with tractable forward transition

[WHNC, NeurIPS, 2025]

Designing weighting functions

💡 Weighting: calibrate the discrepancy between proposal and exact reverse transition

$$\pi_{t_{n+1}}(x_{t_{n+1}})\pi(x_{t_n} \mid x_{t_{n+1}}) = \pi_{t_n}(x_{t_n})\pi(x_{t_{n+1}} \mid x_{t_n})$$



$$w_{t_n} = \frac{\pi(x_{t_n} \mid x_{t_{n+1}})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

$$= \frac{\pi_{t_n}(x_{t_n})}{\pi_{t_{n+1}}(x_{t_{n+1}})} \cdot \frac{\pi(x_{t_{n+1}} \mid x_{t_n})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

replaced with tractable forward transition

$$\approx \boxed{\frac{\hat{\pi}_{t_n}(x_{t_n})}{\hat{\pi}_{t_{n+1}}(x_{t_{n+1}})}} \cdot \frac{\pi(x_{t_{n+1}} \mid x_{t_n})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

marginals estimated again by MC (next slides)

[WHNC, NeurIPS, 2025]

Recall Monte Carlo score estimation

- Draw Monte Carlo samples from posterior $\pi(x_0 \mid x_t) \propto \tilde{\pi}(x_0)\mathcal{N}(x_t \mid \alpha_t x_0, \sigma_t^2)$

Example: Importance Sampling

1. Draw M importance samples $x_0^{(1:M)} \sim q(\,\cdot \mid x_t) := \mathcal{N}(\,\cdot \mid x_t/\alpha_t, \sigma_t^2/\alpha_t^2)$
2. Compute importance weights $w^{(m)} \leftarrow \dfrac{\tilde{\pi}(x_0^{(m)})\mathcal{N}(x_t \mid \alpha_t x_0^{(m)}, \sigma_t^2)}{q(x_0^{(m)} \mid x_t)}$

- Estimate posterior mean $\hat{x}_0 \approx \mathbb{E}_\pi[x_0 \mid x_t]$

$$\hat{x}_0 \leftarrow \sum_{m=1}^{M} w^{(m)} x_0^{(m)} / \sum_{m=1}^{M} w^{(m)}$$

- Estimate score $\hat{s}(x_t) := \dfrac{\alpha_t \hat{x}_0 - x_t}{\sigma_t^2}$ by Tweedie's

Recall Monte Carlo score estimation

- Draw Monte Carlo samples from posterior $\pi(x_0 \mid x_t) \propto \tilde{\pi}(x_0)\mathcal{N}(x_t \mid \alpha_t x_0, \sigma_t^2)$

Example: Importance Sampling

1. Draw M importance samples $x_0^{(1:M)} \sim q(\cdot \mid x_t) := \mathcal{N}(\cdot \mid x_t/\alpha_t, \sigma_t^2/\alpha_t^2)$
2. Compute importance weights $w^{(m)} \leftarrow \dfrac{\tilde{\pi}(x_0^{(m)})\mathcal{N}(x_t \mid \alpha_t x_0^{(m)}, \sigma_t^2)}{q(x_0^{(m)} \mid x_t)}$

- Estimate posterior mean $\hat{x}_0 \approx \mathbb{E}_\pi[x_0 \mid x_t]$

$$\hat{x}_0 \leftarrow \sum_{m=1}^{M} w^{(m)} x_0^{(m)} / \sum_{m=1}^{M} w^{(m)}$$

- Estimate score $\hat{s}(x_t) := \dfrac{\alpha_t \hat{x}_0 - x_t}{\sigma_t^2}$ by Tweedie's

💡**Note $Z\pi_t(x_t) = \int \tilde{\pi}_0(x_0)\mathcal{N}(x_t \mid \alpha_t x_0, \sigma_t^2)dx_0$ is the normalization constant of posterior**

Recall Monte Carlo score estimation

- Draw Monte Carlo samples from posterior $\pi(x_0 \mid x_t) \propto \tilde{\pi}(x_0)\mathcal{N}(x_t \mid \alpha_t x_0, \sigma_t^2)$

Example: Importance Sampling

1. Draw M importance samples $x_0^{(1:M)} \sim q(\cdot \mid x_t) := \mathcal{N}(\cdot \mid x_t/\alpha_t, \sigma_t^2/\alpha_t^2)$
2. Compute importance weights $w^{(m)} \leftarrow \dfrac{\tilde{\pi}(x_0^{(m)})\mathcal{N}(x_t \mid \alpha_t x_0^{(m)}, \sigma_t^2)}{q(x_0^{(m)} \mid x_t)}$

- Estimate posterior mean $\hat{x}_0 \approx \mathbb{E}_\pi[x_0 \mid x_t]$

$$\hat{x}_0 \leftarrow \sum_{m=1}^{M} w^{(m)} x_0^{(m)} / \sum_{m=1}^{M} w^{(m)}$$

- Estimate score $\hat{s}(x_t) := \dfrac{\alpha_t \hat{x}_0 - x_t}{\sigma_t^2}$ by Tweedie's

- **Estimate marginal $\hat{\pi}_t(x_t) \approx Z\pi_t(x_t)$**

$$\hat{\pi}_t(x_t) \leftarrow \sum_{m=1}^{M} w^{(m)} \quad \text{(unbiased)}$$

Recall Monte Carlo score estimation

- Draw Monte Carlo samples from posterior $\pi(x_0 \mid x_t) \propto \tilde{\pi}(x_0)\mathcal{N}(x_t \mid \alpha_t x_0, \sigma_t^2)$

MC methods that emit both posterior samples and marginal estimates

IS, AIS, SMC, rejection sampling, ~~MCMC~~

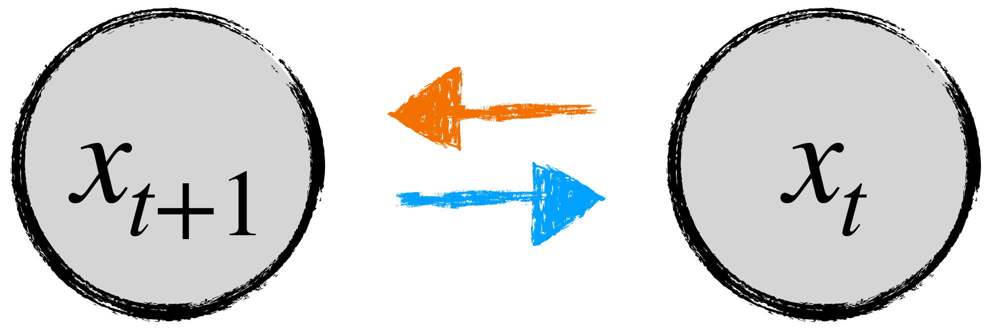
- Estimate posterior mean $\hat{x}_0 \approx \mathbb{E}_\pi[x_0 \mid x_t]$
- Estimate score $\hat{s}(x_t) := \dfrac{\alpha_t \hat{x}_0 - x_t}{\sigma_t^2}$ by Tweedie's
- **Estimate marginal $\hat{\pi}_t(x_t) \approx Z\pi_t(x_t)$**

Designing weighting functions

💡 Weighting: calibrate the discrepancy between proposal and exact reverse transition

$$w_{t_n} = \frac{\pi(x_{t_n} \mid x_{t_{n+1}})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

$$\pi_{t_{n+1}}(x_{t_{n+1}})\pi(x_{t_n} \mid x_{t_{n+1}}) = \pi_{t_n}(x_{t_n})\pi(x_{t_{n+1}} \mid x_{t_n})$$



$$= \frac{\pi_{t_n}(x_{t_n})}{\pi_{t_{n+1}}(x_{t_{n+1}})} \cdot \frac{\pi(x_{t_{n+1}} \mid x_{t_n})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

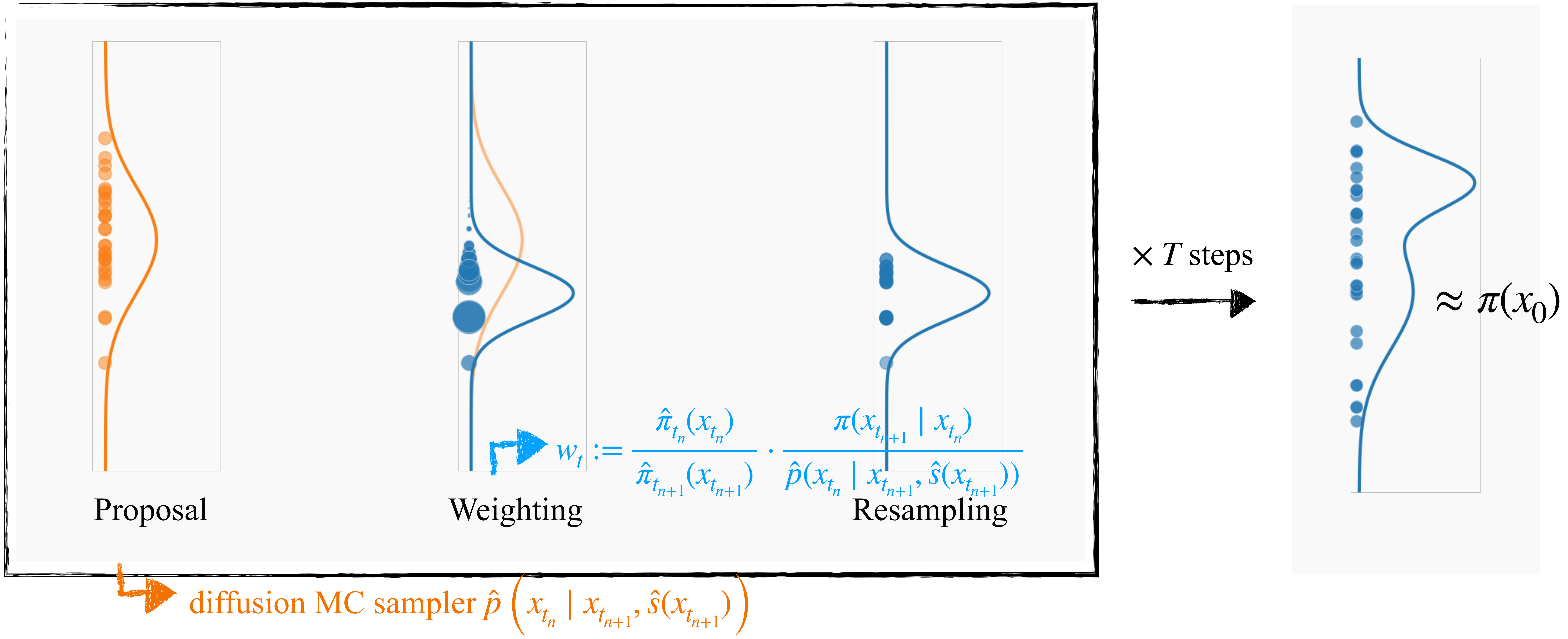
replaced with tractable forward transition

$$\approx \boxed{\frac{\hat{\pi}_{t_n}(x_{t_n})}{\hat{\pi}_{t_{n+1}}(x_{t_{n+1}})}} \cdot \frac{\pi(x_{t_{n+1}} \mid x_{t_n})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

marginals estimated again by MC

[WHNC, NeurIPS, 2025]

RDSMC: Reverse Diffusion Sequential Monte Carlo



[WHNC, NeurIPS, 2025]

RDSMC: Reverse Diffusion Sequential Monte Carlo

1. Sample K *particles* from base $x_1^{1:K} \sim \pi_1(x) = \mathcal{N}(0,1)$

   (i) Compute SMC weights $w_1^{1:K}$

2. For $n = N, N-1, \cdots, 1$ $(t_N = 1, t_0 = 0)$

   (i) ***Resample*** particles $x_{t_n}^{1:K}$ according to weights $w_{t_n}^{1:K}$

   (ii) ***Propose*** next particles $x_{t_{n-1}}^{1:K}$

$$x_{t_{n-1}} \mid x_{t_n} \sim \mathcal{N}(x_{t_{n-1}} \mid [x_{t_n} - (f_{t_n} x_{t_n} - g_{t_n}^2 \hat{s}(x_{t_n})](t_n - t_{n-1}), g_{t_n}^2 (t_n - t_{n-1}))$$

   (ii) Compute SMC ***weights*** $w_{t_{n-1}}^{1:K}$

3. Return $\{x_0^{1:K}, w_0^{1:K}\}$, and $\sum_k w_0^k \delta_{x_0^k} \to \pi_0$ as $K \to \infty$

[**W**HNC, NeurIPS, 2025]

Theory

Theorem (Informal)

RDSMC is asymptotically exact as the number of particles $K \to \infty$, and provides unbiased estimate of normalization constant for any fixed K

[WHNC, NeurIPS, 2025]

Theory

Theorem (Informal)

RDSMC is asymptotically exact as the number of particles $K \to \infty$, and provides unbiased estimate of normalization constant for any fixed K

- The final target of RDSMC recovers the desired $\pi(x_0)$

$$\gamma_0(x_{t_0}, \cdots, x_{t_N}) \propto \mathbb{E}\left[\text{proposal}(x_{t_0}, \cdots, x_{t_N}) \prod_{n=0}^{N} w_{t_n}\right]$$

[WHNC, NeurIPS, 2025]

Theory

Theorem (Informal)

RDSMC is asymptotically exact as the number of particles $K \to \infty$, and provides unbiased estimate of normalization constant for any fixed K

- The final target of RDSMC recovers the desired $\pi(x_0)$

Marginal estimates cancelled out in cross products

$$\gamma_0(x_{t_0}, \cdots, x_{t_N}) \propto \mathbb{E}\left[\text{proposal}(x_{t_0}, \cdots, x_{t_N}) \prod_{n=0}^{N} w_{t_n}\right] \xrightarrow{\text{Plug in}} w_{t_n} = \boxed{\frac{\hat{\pi}_{t_n}(x_{t_n})}{\hat{\pi}_{t_{n+1}}(x_{t_{n+1}})}} \cdot \frac{\pi(x_{t_{n+1}} \mid x_{t_n})}{\hat{p}(x_{t_n} \mid x_{t_{n+1}}, \hat{s}(x_{t_{n+1}}))}$$

$$= \tilde{\pi}_0(x_0) \prod_{n=0}^{N-1} \pi(x_{t_{n+1}} \mid x_{t_n})$$ The final marginal $\tilde{\pi}(x_0)$ is exact

$$\propto \pi(x_{t_0}, \cdots, x_{t_N})$$ Recovers the desired distribution (in joint space)

[WHNC, NeurIPS, 2025]

Theory

Theorem (Informal)

RDSMC is asymptotically exact as the number of particles $K \to \infty$, and provides unbiased estimate of normalization constant for any fixed K

- The intermediate target of RDSMC is *an exact approximation*

$$\gamma_n(x_{t_n}, \cdots, x_{t_N}) \propto \mathbb{E}\left[\text{proposal}(x_{t_n}, \cdots, x_{t_N}) \prod_{n'=n}^{N} w_{t'_n}\right]$$

$$= \mathbb{E}\left[\tilde{\pi}_n(x_n) \prod_{n'=n}^{N-1} \pi(x_{t_{n'+1}} \mid x_{t'_n})\right]$$

$$\propto \pi(x_{t_n}, \cdots, x_{t_N})$$

[WHNC, NeurIPS, 2025]

Theory

Theorem (Informal)

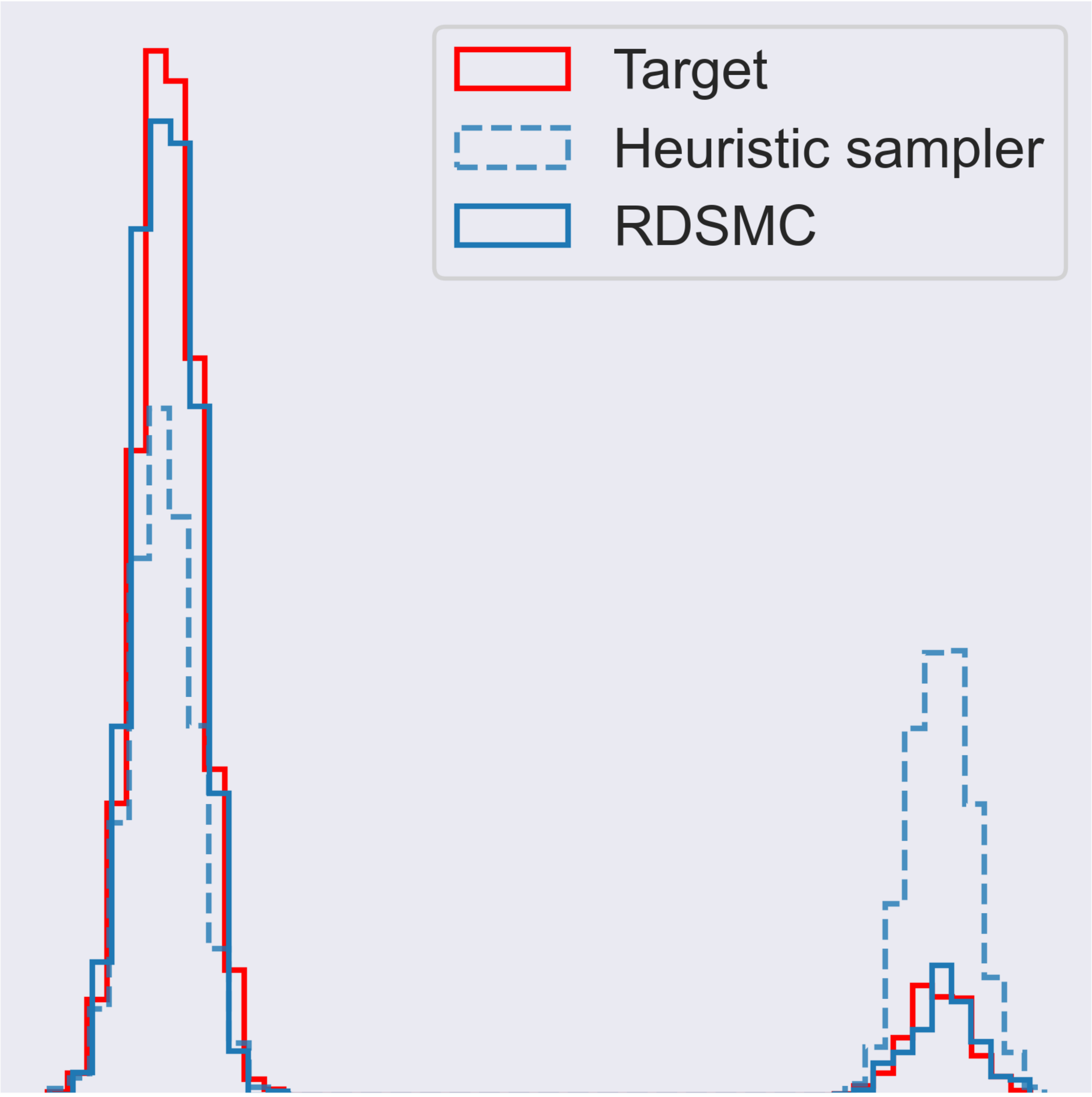
RDSMC is asymptotically exact as the number of particles $K \to \infty$, and provides unbiased estimate of normalization constant for any fixed K

- Practical implication: debias diffusion MC samplers at no additional cost

💡 Both *score* $\hat{s}(x_t)$ and *marginal* $\hat{\pi}(x_t)$ estimates rely on the same MC procedure targeting $\pi(x_0 \mid x_t)$

[WHNC, NeurIPS, 2025]

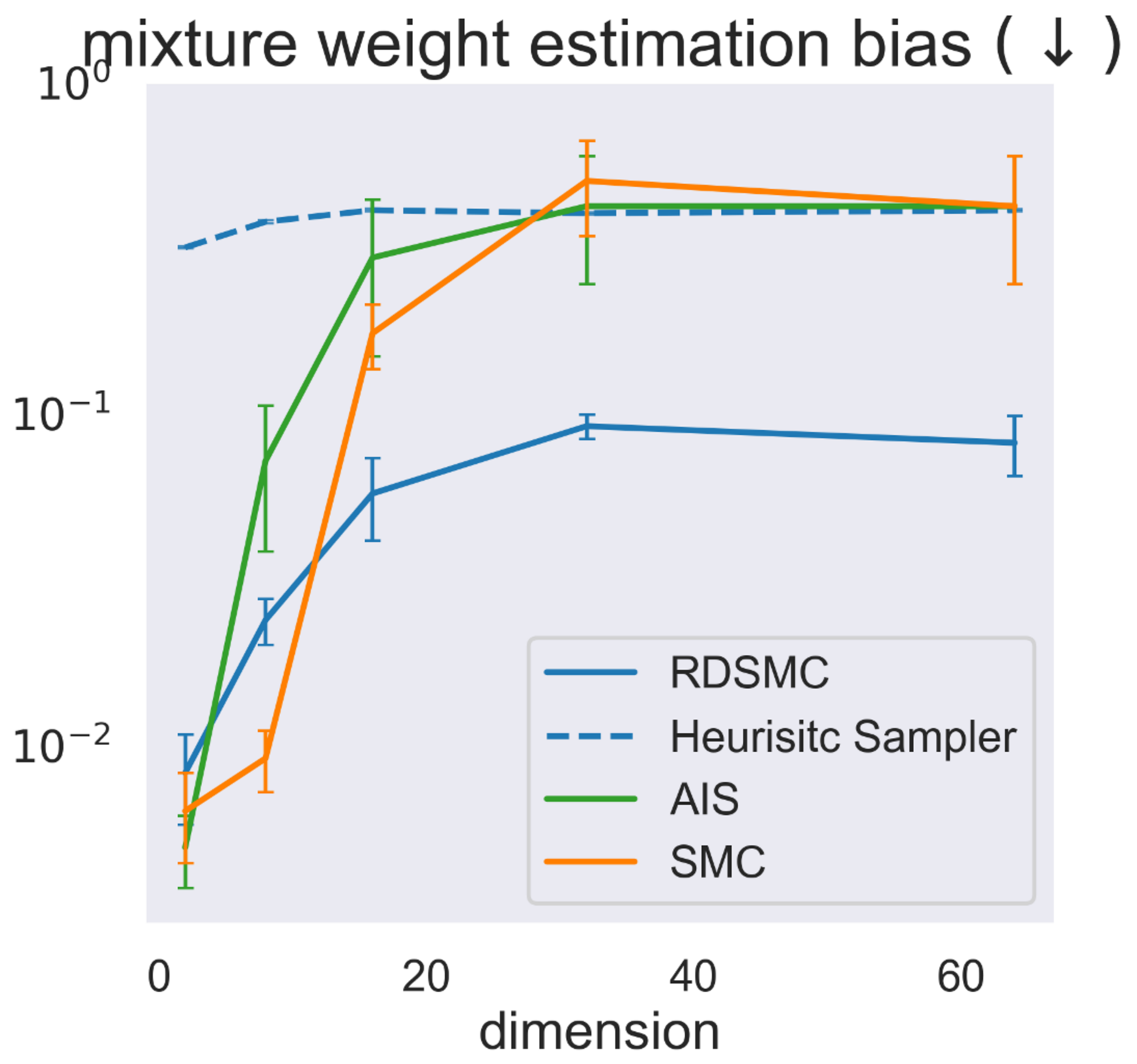
Revisiting bi-modal Gaussian mixtures

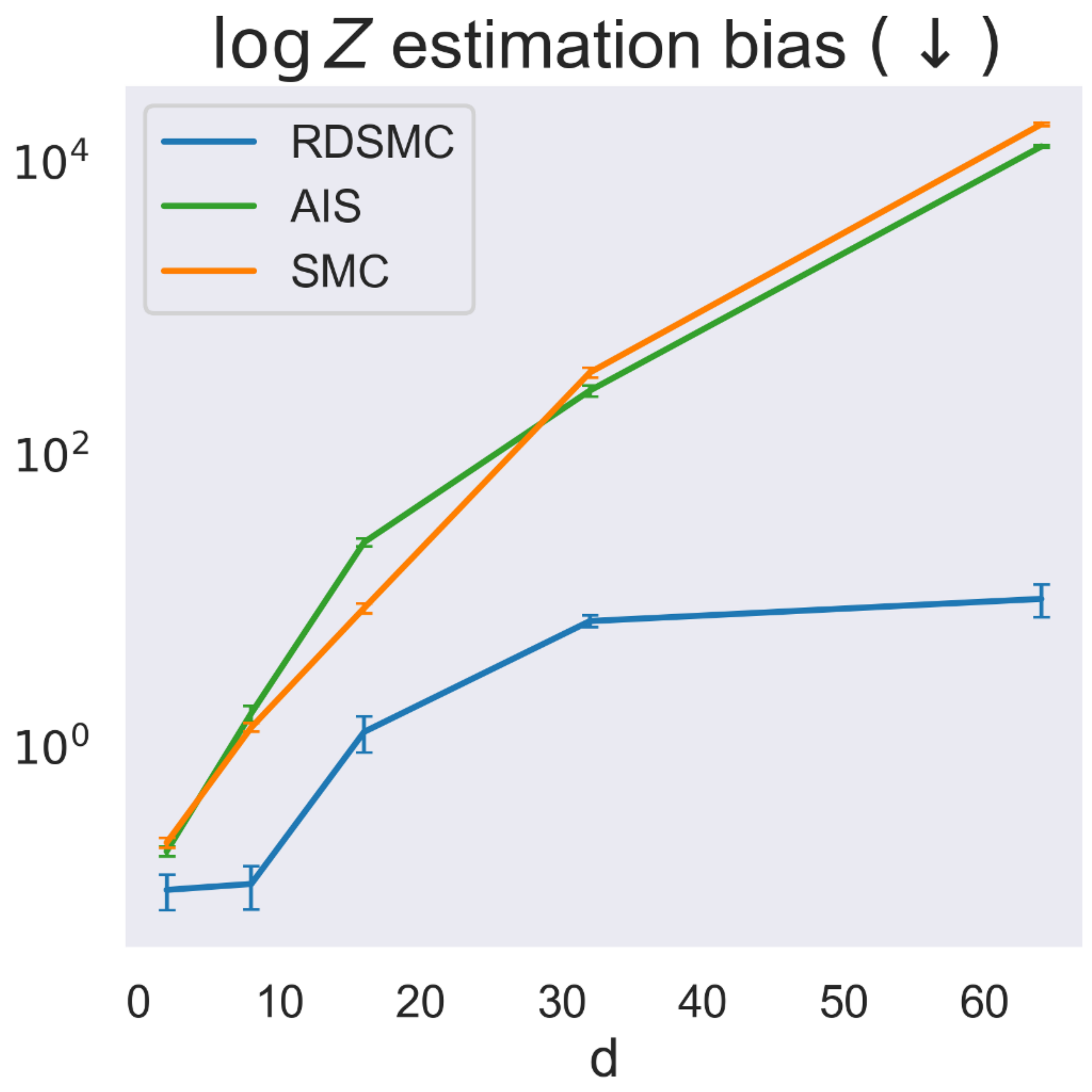


RDSMC generate calibrated samples, effectively debiasing heuristic approach

[WHNC, NeurIPS, 2025]

More on bimodal Gaussian mixtures





RDSMC produces lower estimation bias of mixture weight and normalization constant across dimensions.

Geometric-annealing-based AIS and SMC tend to have mode collapse, leading to larger estimation bias.

[WHNC, NeurIPS, 2025]

Bayesian logistic regression task

TEST LPPD $\uparrow$	Credit ($d = 25$)	Cancer ($d = 31$)	Ionosphere ($d = 35$)	Sonar ($d = 61$)
RDSMC	**-94.54 $\pm$ 1.34**	**-10.59 $\pm$ 0.45**	**-25.97 $\pm$ 0.73**	**-18.54 $\pm$ 0.28**
RDSMC-P	-94.62 $\pm$ 0.06	-50.24 $\pm$ 0.16	**-24.87 $\pm$ 0.02***	-18.91 $\pm$ 0.01
RDMC	-138.22 $\pm$ 0.35	-78.03 $\pm$ 0.12	-44.16 $\pm$ 0.07	-28.64 $\pm$ 0.03
SLIPS	**-92.42 $\pm$ 0.02***	-10.41 $\pm$ 0.01	-25.22 $\pm$ 0.01	**-18.40 $\pm$ 0.01***
AIS	**-92.91 $\pm$ 0.44**	**-10.13 $\pm$ 0.07**	-25.09 $\pm$ 0.02	**-18.41 $\pm$ 0.01**
SMC	**-92.55 $\pm$ 0.05**	**-10.13 $\pm$ 0.01***	-25.09 $\pm$ 0.02	**-18.41 $\pm$ 0.01**
SMS	-98.00 $\pm$ 0.27	-20.47 $\pm$ 0.16	-26.17 $\pm$ 0.10	-23.29 $\pm$ 0.08

Ours with and without bias correction

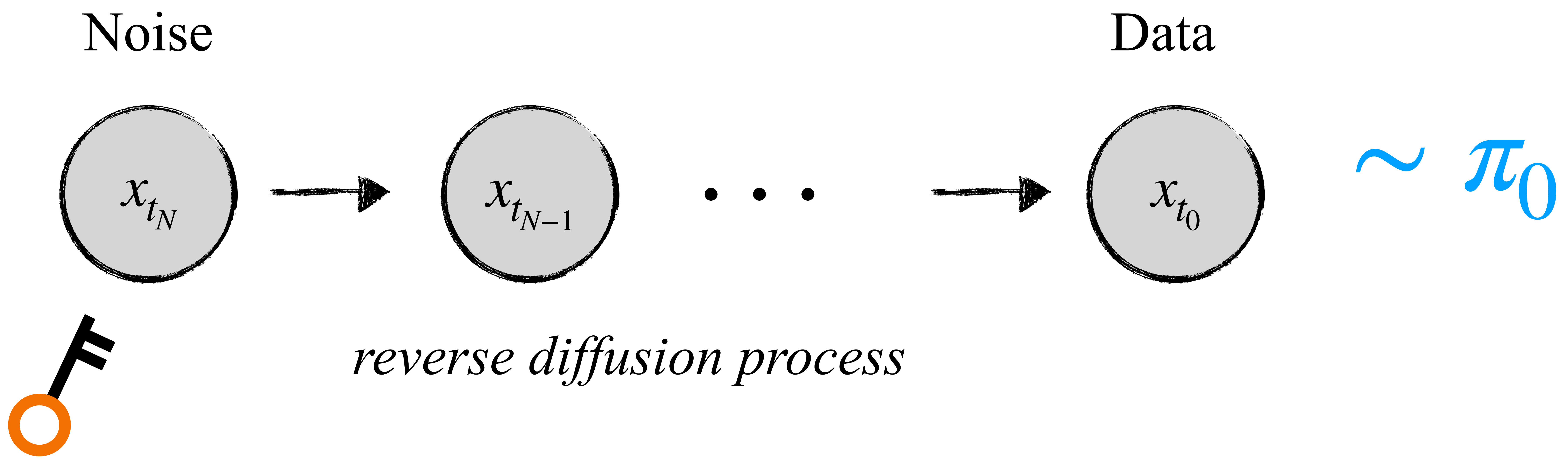
Diffusion-based annealing, without bias correction

Geometric annealing with bias correction

RDSMC outperforms or matches heuristic diffusion samplers, and geoemtric-annealing AIS/SMC samplers.

[WHNC, NeurIPS, 2025]

Summary



Reverse Diffusion Sequential Monte Carlo (RDSMC)

- Use Monte Carlo estimates to approximately simulate diffusion annealing paths
- Bias correction mechanism via SMC

[WHNC, NeurIPS, 2025]

Discussion

- SMC performance depends on the proposal quality
- Alternatively to SMC, other methods explore better initialization of diffusion paths

  [e.g. Huang et al., ICLR 2024; Grenioux et al., ICML 2024; He et al., NeurIPS 2024]

Future directions

- SMC variance reduction by increasing correlations
- Debiasing learning-based diffusion samplers
- Extending to discrete distributions