Diffusion-Inspired Enhanced Sampling

Wenlin Chen
University of Cambridge & Max Planck Institute for Intelligent Systems

https://wenlin-chen.github.io/
chen.wenlin@outlook.com

Sampling

Sampling from unnormalized distributions

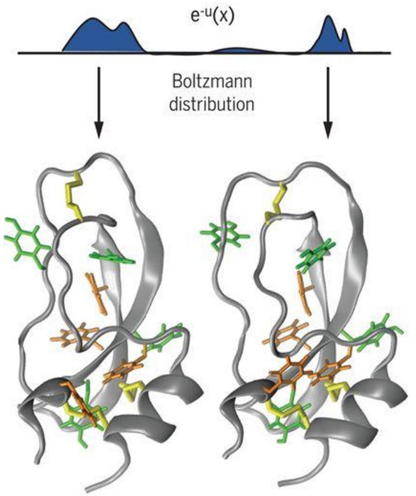
$$p(x) = \frac{\tilde{p}(x)}{Z}$$

Boltzmann distribution with unnormalized density

$$\tilde{p}(x) = \exp(-E(x)/kT)$$

- $\tilde{p}(x)$ easy to evaluate but hard to sample from

- Score function (force) can be evaluated: $\nabla_x \log p(x) = -\nabla_x E(x)$



Noé, Frank, et al. "Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning." Science 365.6457 (2019): eaaw1147.

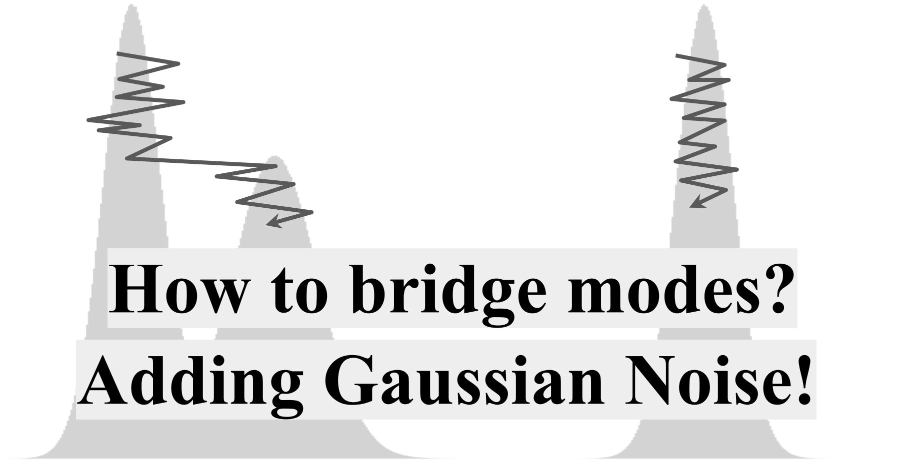
Markov Chain Monte Carlo

Sampling from unnormalized distributions

$$p(x) = \frac{\tilde{p}(x)}{Z}$$

“Standard” solution: Markov chain Monte Carlo (MCMC)

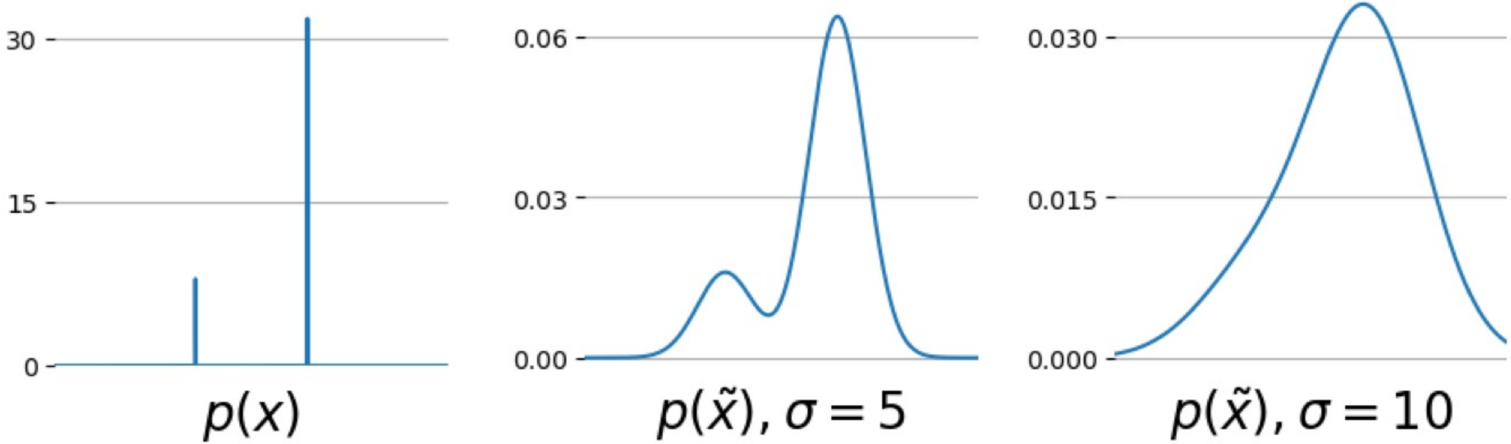
Challenges:



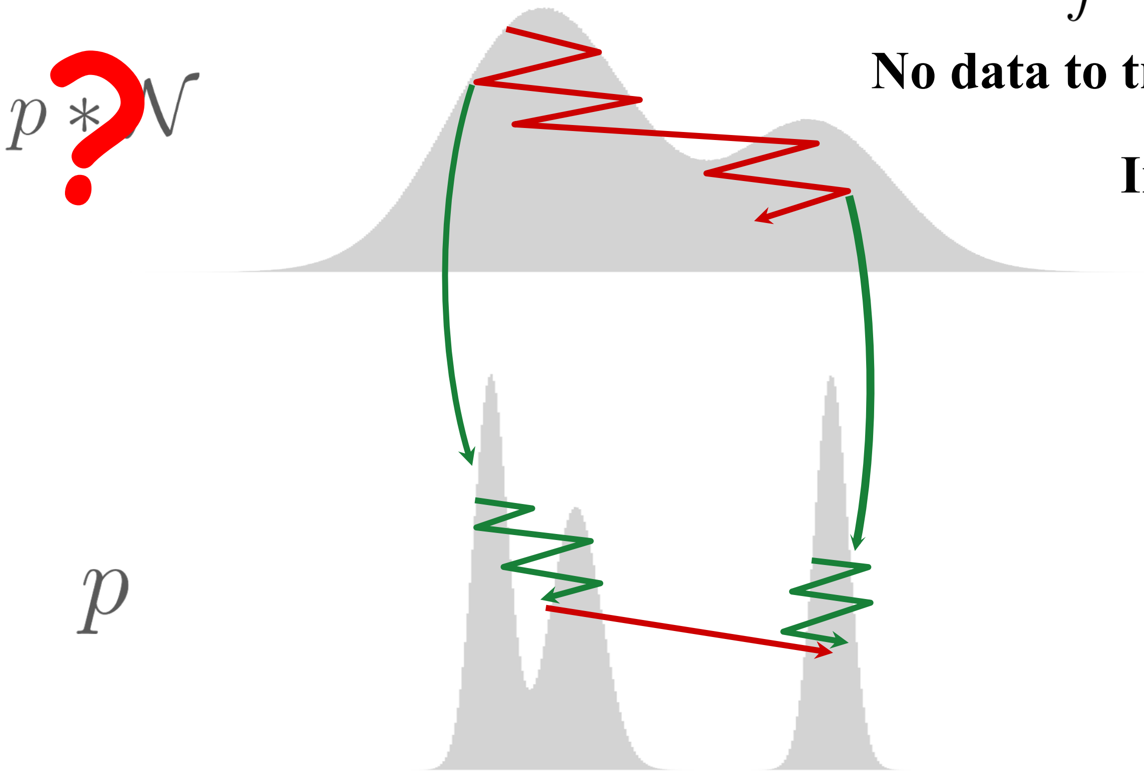
How to bridge modes?

Adding Gaussian Noise!

Diffusion Connects Modes



Sampling in the Noisy Space



$$p * \mathcal{N} = \int \mathcal{N}(x_\sigma | x, \sigma) p(x) dx$$

No data to train a score model!

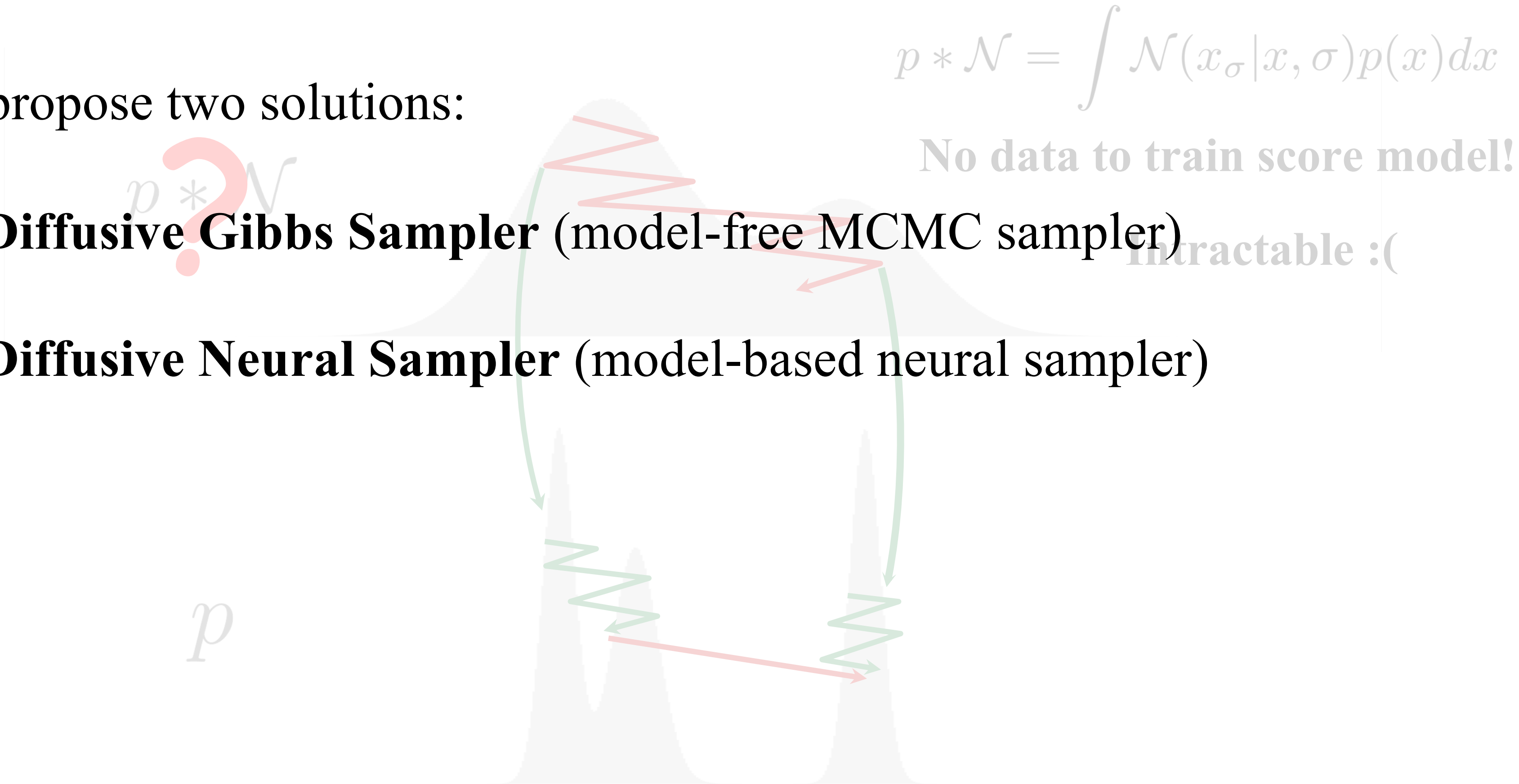
Intractable :(

Diffusion-Inspired Samplers

We propose two solutions:

- **Diffusive Gibbs Sampler** (model-free MCMC sampler)

- **Diffusive Neural Sampler** (model-based neural sampler)



Diffusive Gibbs Sampler (Naïve)

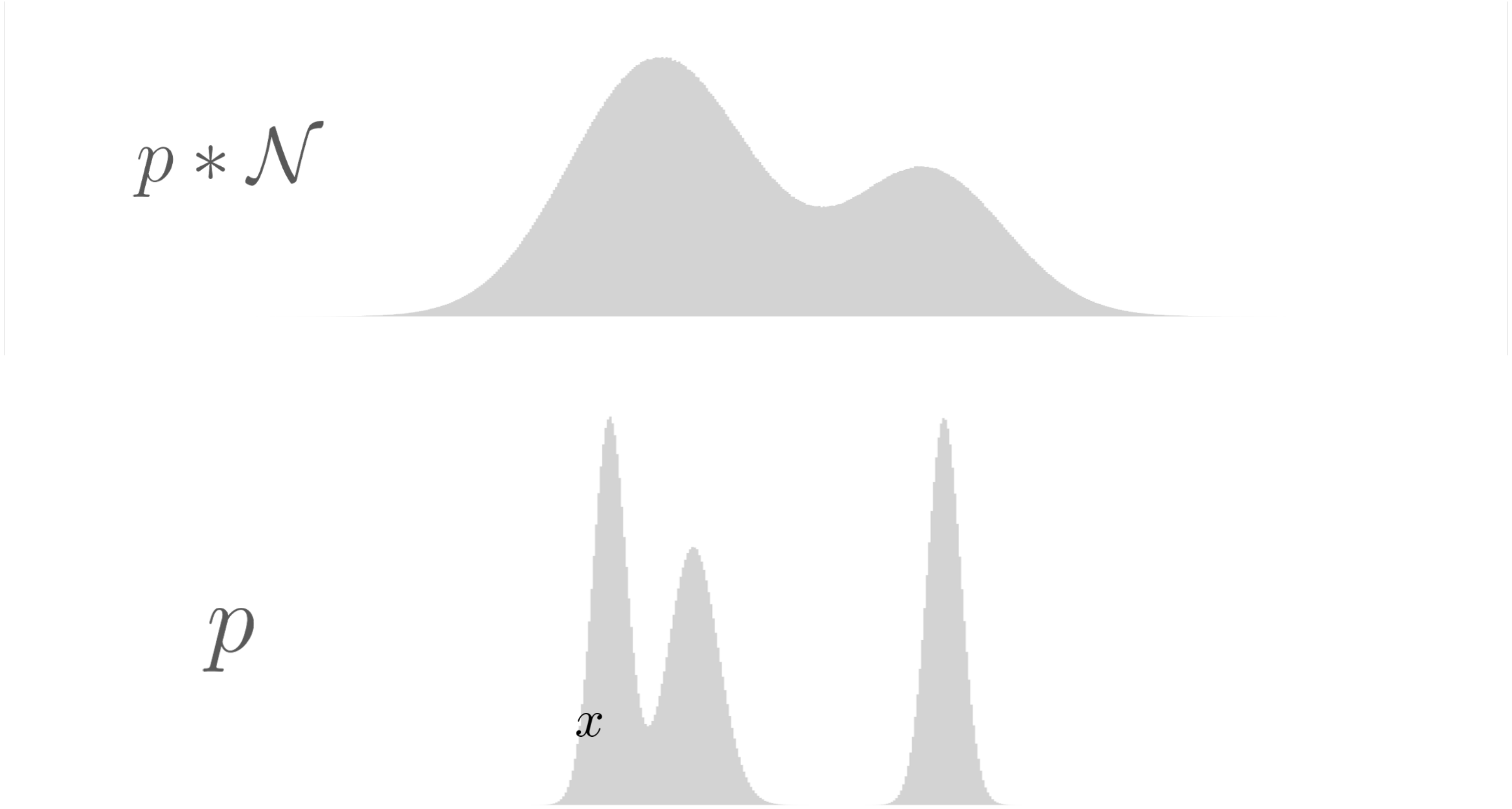
Joint space is more tractable :)

$$p(x, x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma | x, \sigma)$$

Diffusive Gibbs Sampler (Naïve)

Joint space is more tractable :)

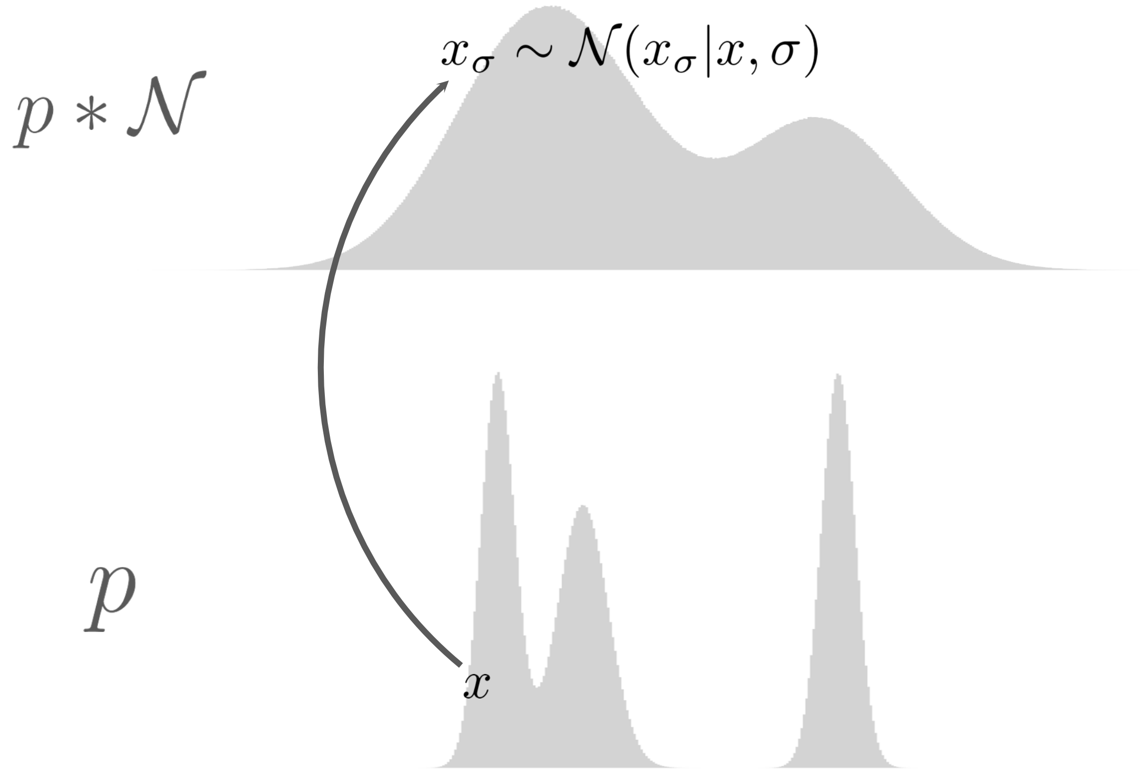
$$p(x, x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma | x, \sigma)$$



Diffusive Gibbs Sampler (Naïve)

Joint space is more tractable :)

$$p(x, x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma | x, \sigma)$$



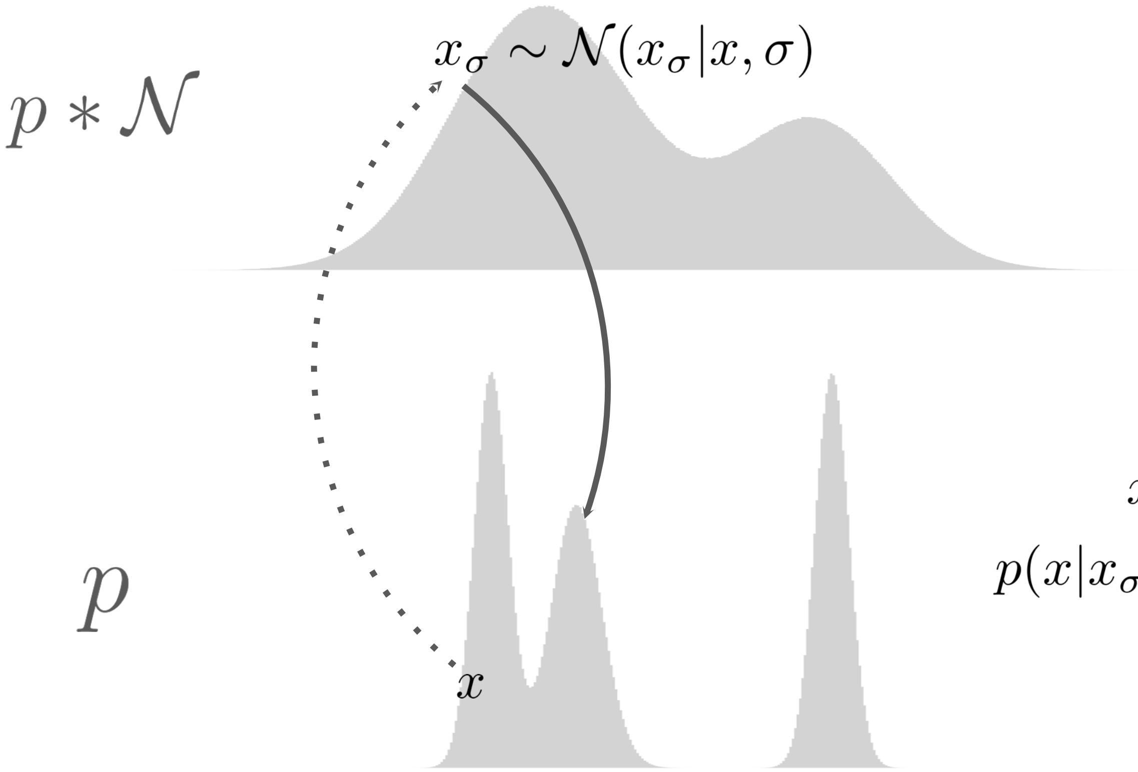
$$x_\sigma = x + \sigma\epsilon$$

$$\epsilon \sim \mathcal{N}(0, I)$$

Diffusive Gibbs Sampler (Naïve)

Joint space is more tractable :)

$$p(x, x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma | x, \sigma)$$



$$x_\sigma = x + \sigma\epsilon$$

$$\epsilon \sim \mathcal{N}(0, I)$$

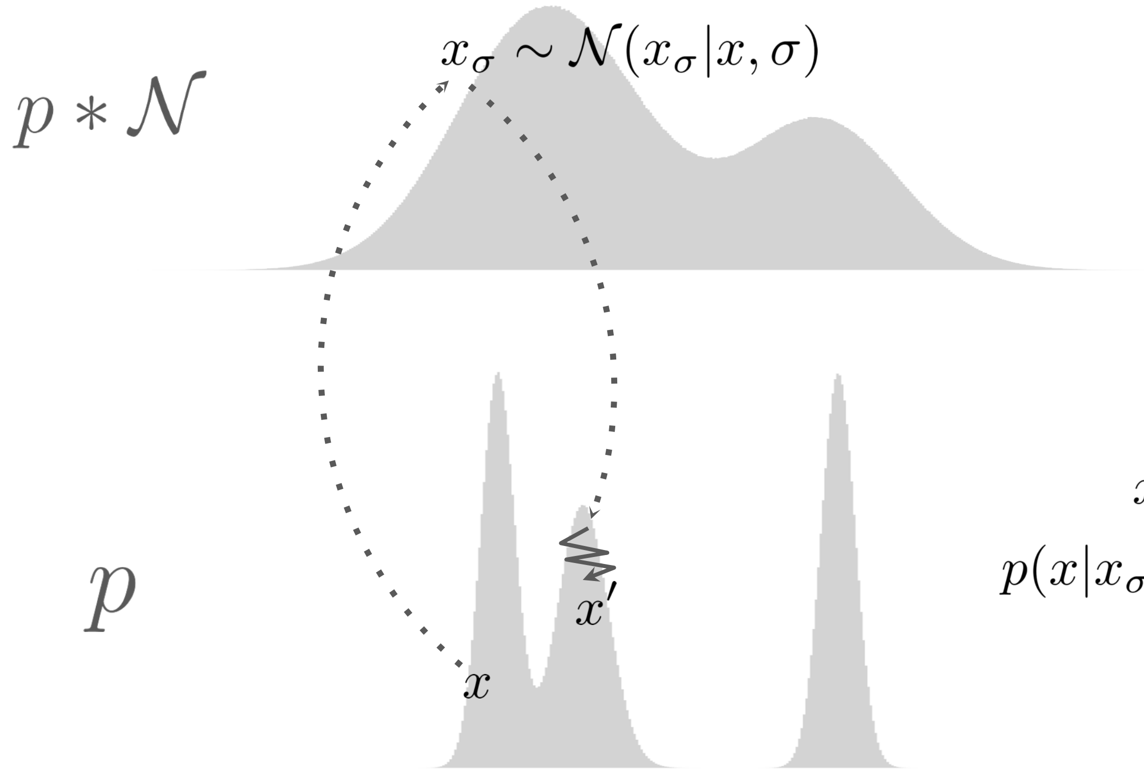
$$x' \sim p(x | x_\sigma)$$

$$p(x | x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma | x, \sigma)$$

Diffusive Gibbs Sampler (Naïve)

Joint space is more tractable :)

$$p(x, x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma | x, \sigma)$$



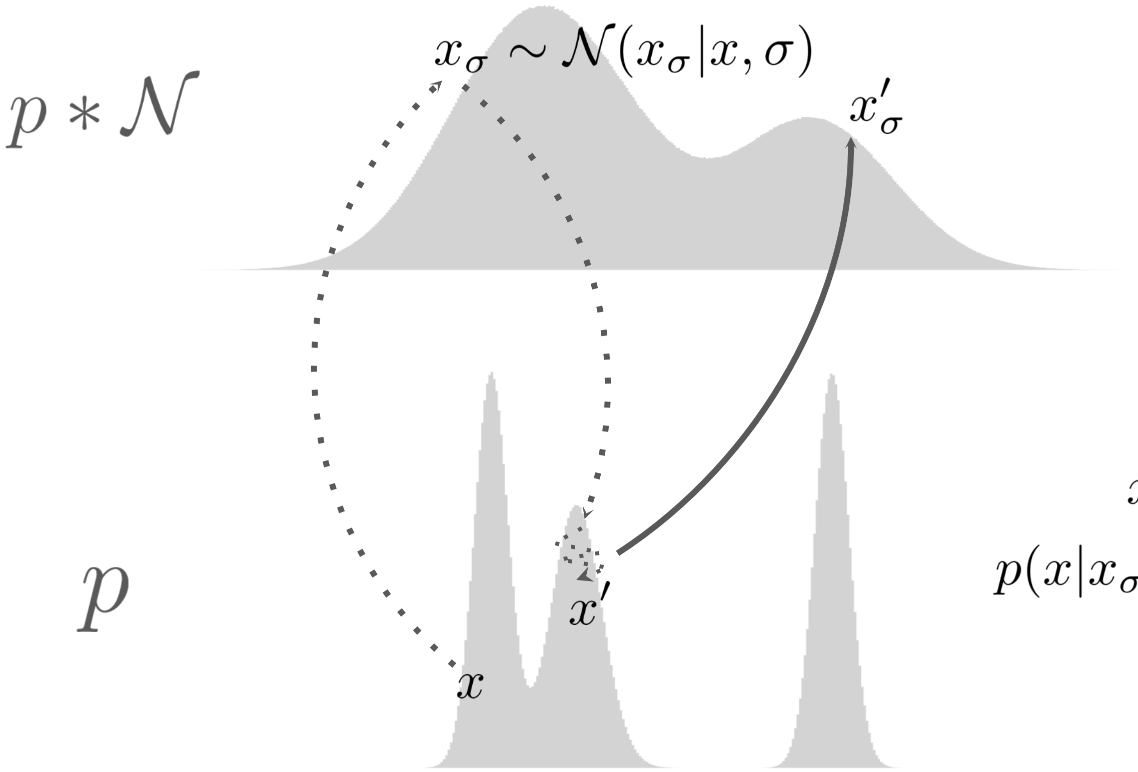
$$x_\sigma = x + \sigma\epsilon$$
$$\epsilon \sim \mathcal{N}(0, I)$$

$$x' \sim p(x | x_\sigma)$$
$$p(x | x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma | x, \sigma)$$

Diffusive Gibbs Sampler (Naïve)

Joint space is more tractable :)

$$p(x, x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma | x, \sigma)$$



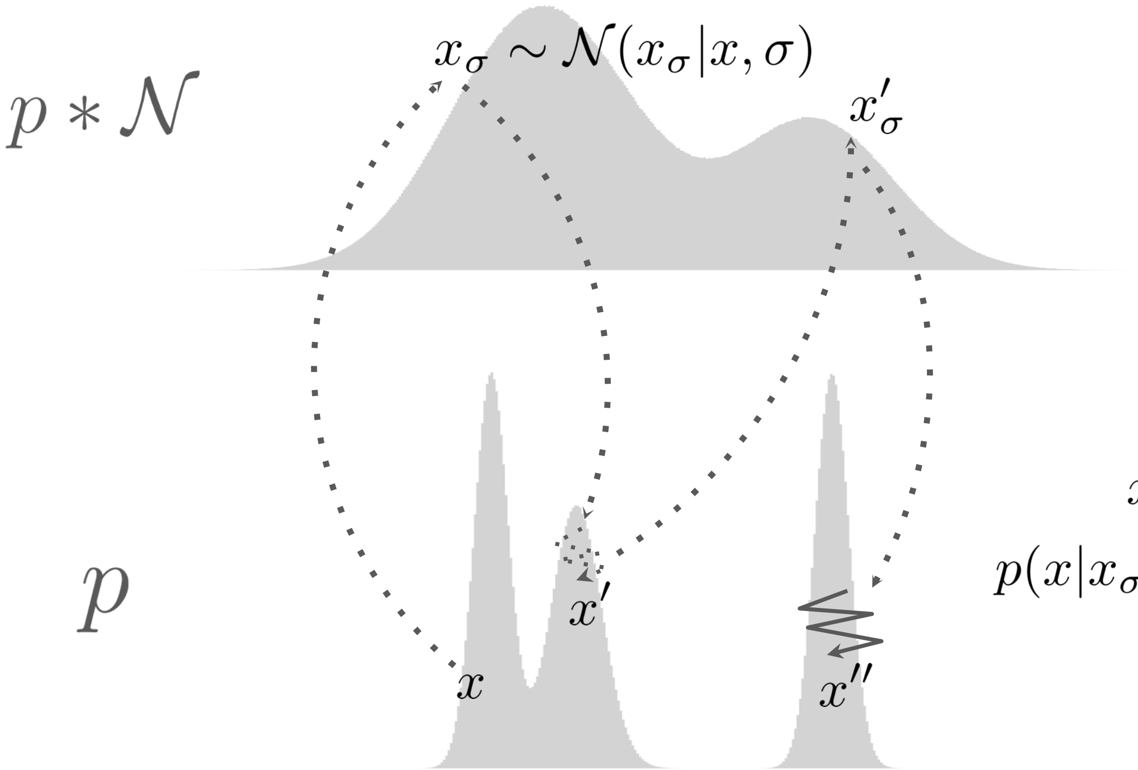
$$x_\sigma = x + \sigma\epsilon$$
$$\epsilon \sim \mathcal{N}(0, I)$$

$$x' \sim p(x | x_\sigma)$$
$$p(x | x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma | x, \sigma)$$

Diffusive Gibbs Sampler (Naïve)

Joint space is more tractable :)

$$p(x, x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma | x, \sigma)$$



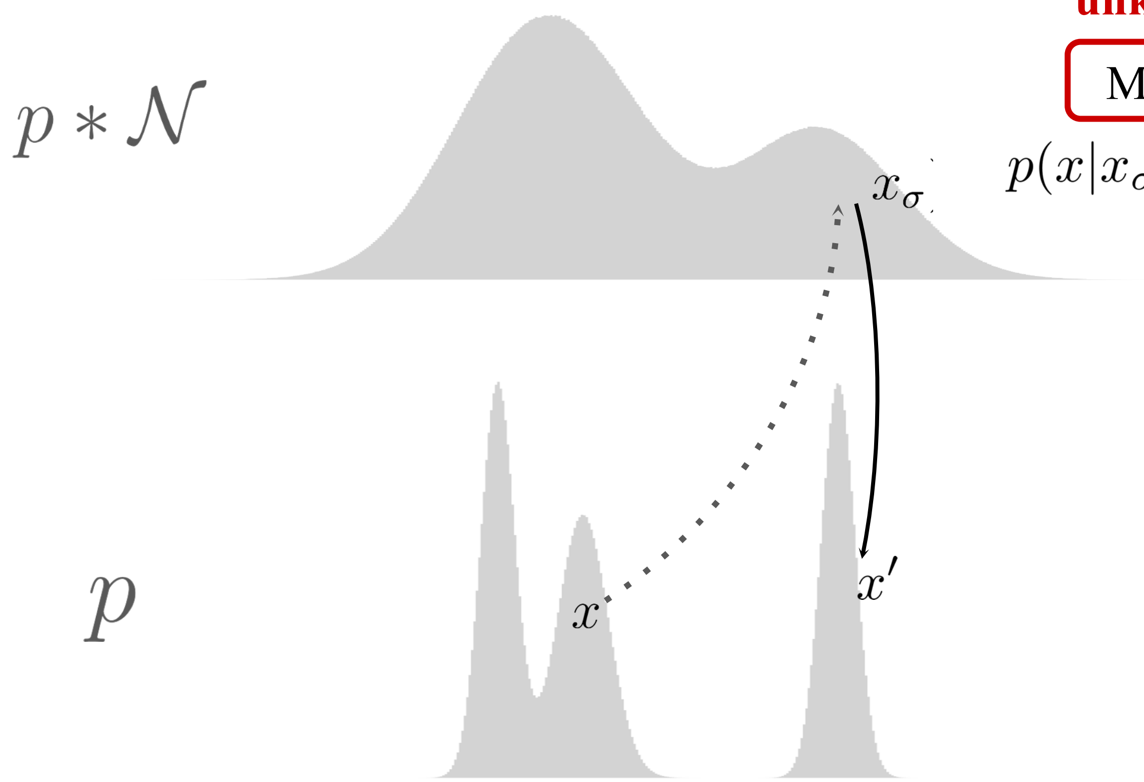
$$x_\sigma = x + \sigma\epsilon$$

$$\epsilon \sim \mathcal{N}(0, I)$$

$$x' \sim p(x | x_\sigma)$$

$$p(x | x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma | x, \sigma)$$

A Caveat in Denoising Posterior Sampling

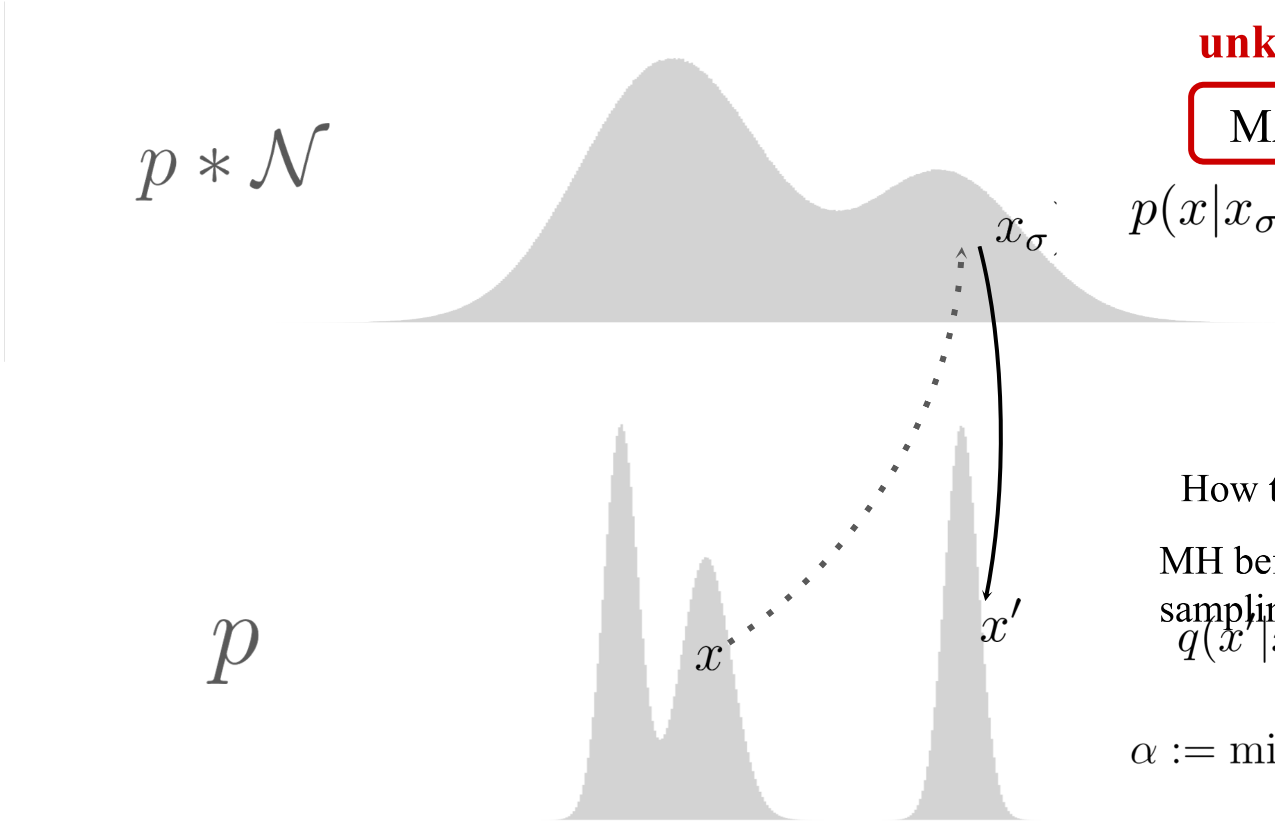


unknown mixing time!

MALA/HMC/AIS/…

$$p(x|x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma|x,\sigma)$$

Metropolis-within-Gibbs



unknown mixing time!

MALA/HMC/AIS/…

$$p(x|x_\sigma) \propto \tilde{p}(x)\mathcal{N}(x_\sigma|x,\sigma)$$

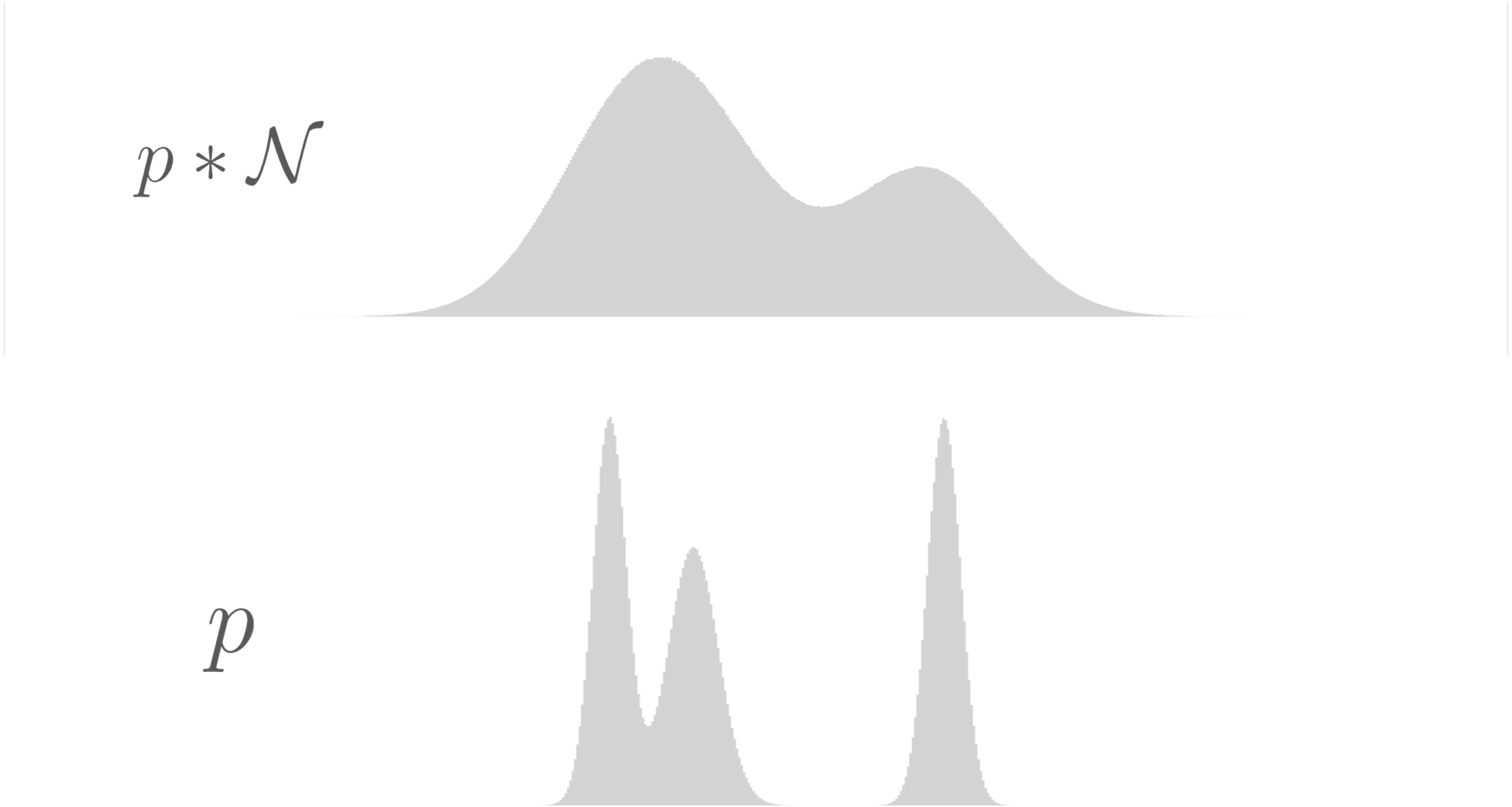
How to ensure $x' \sim p(x|x_\sigma)$?

MH before posterior sampling!

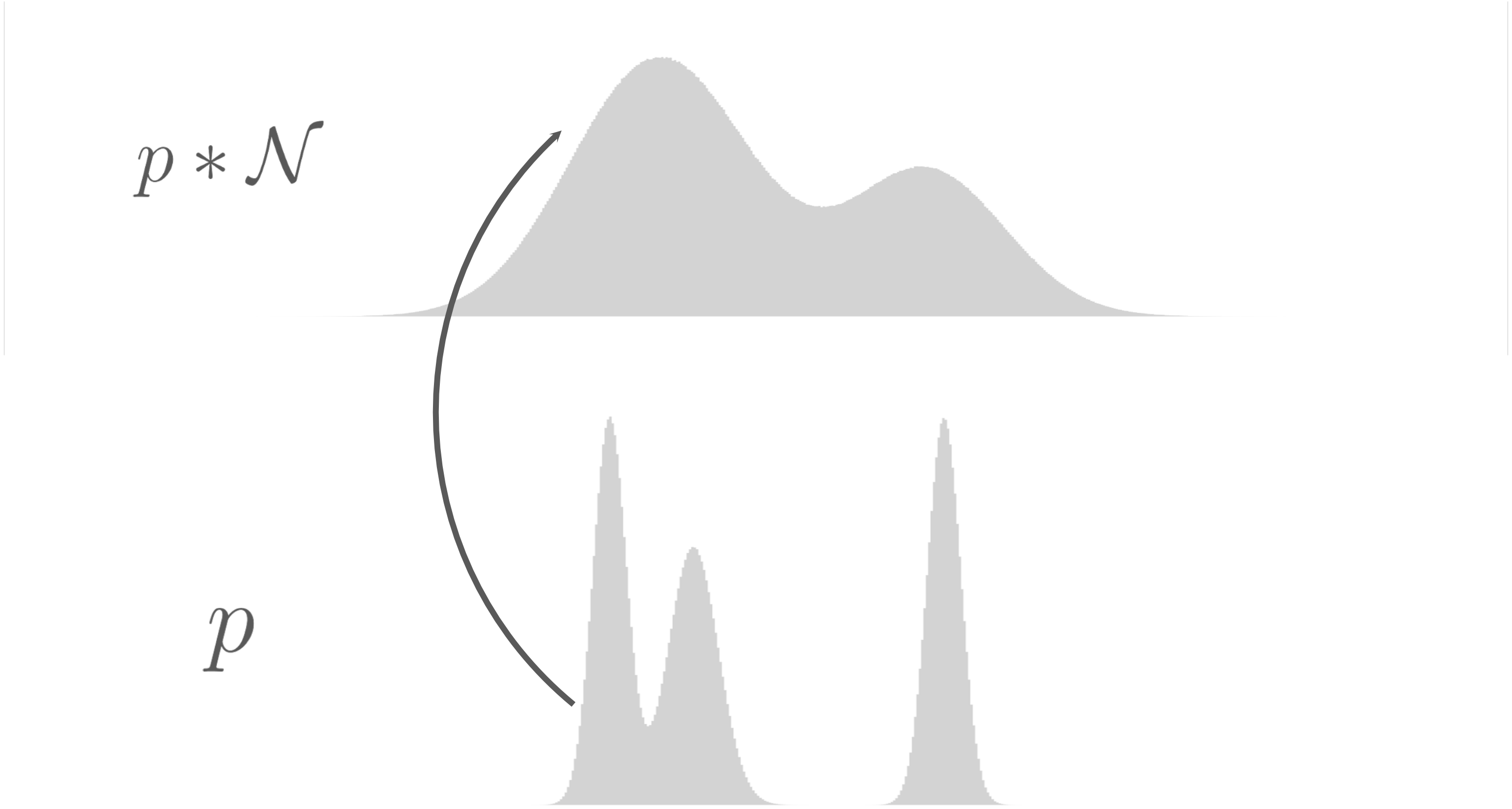
$$q(x'|x_\sigma) = \mathcal{N}(x'|x_\sigma,\sigma)$$

$$\alpha := \min\left(1, \frac{p(x'|x_\sigma)q(x|x_\sigma)}{p(x|x_\sigma)q(x'|x_\sigma)}\right)$$

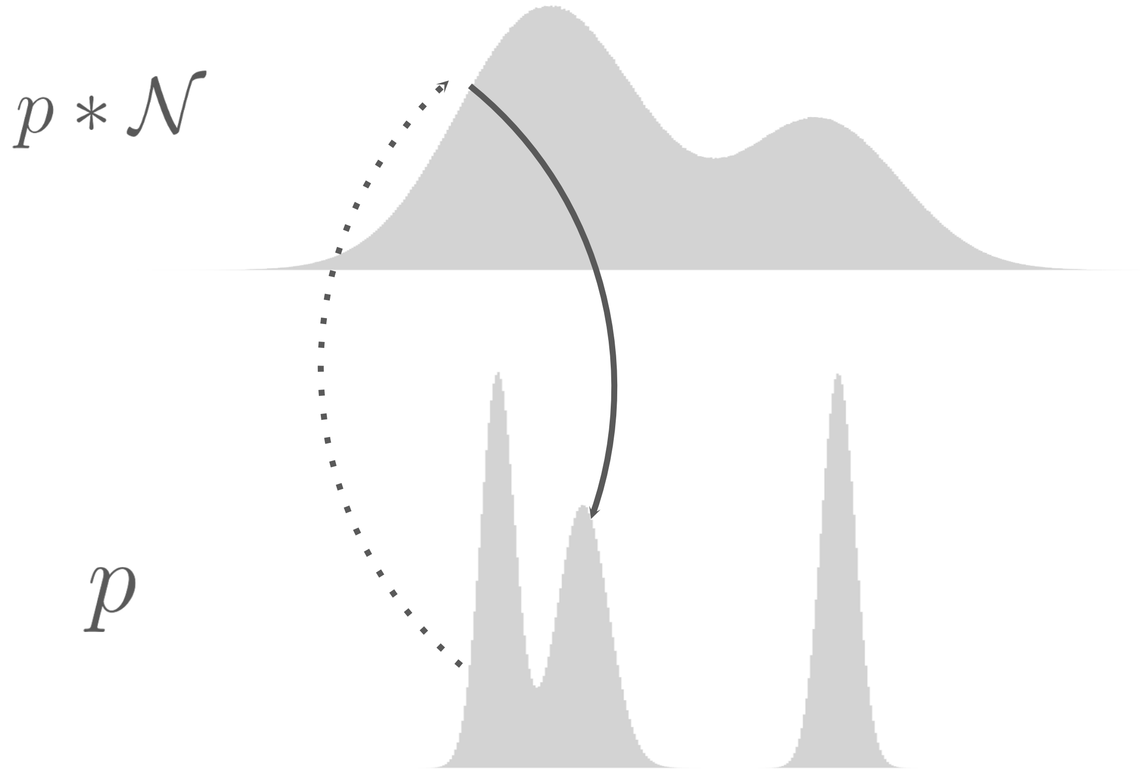
Diffusive Gibbs Sampler (with MH corrector)



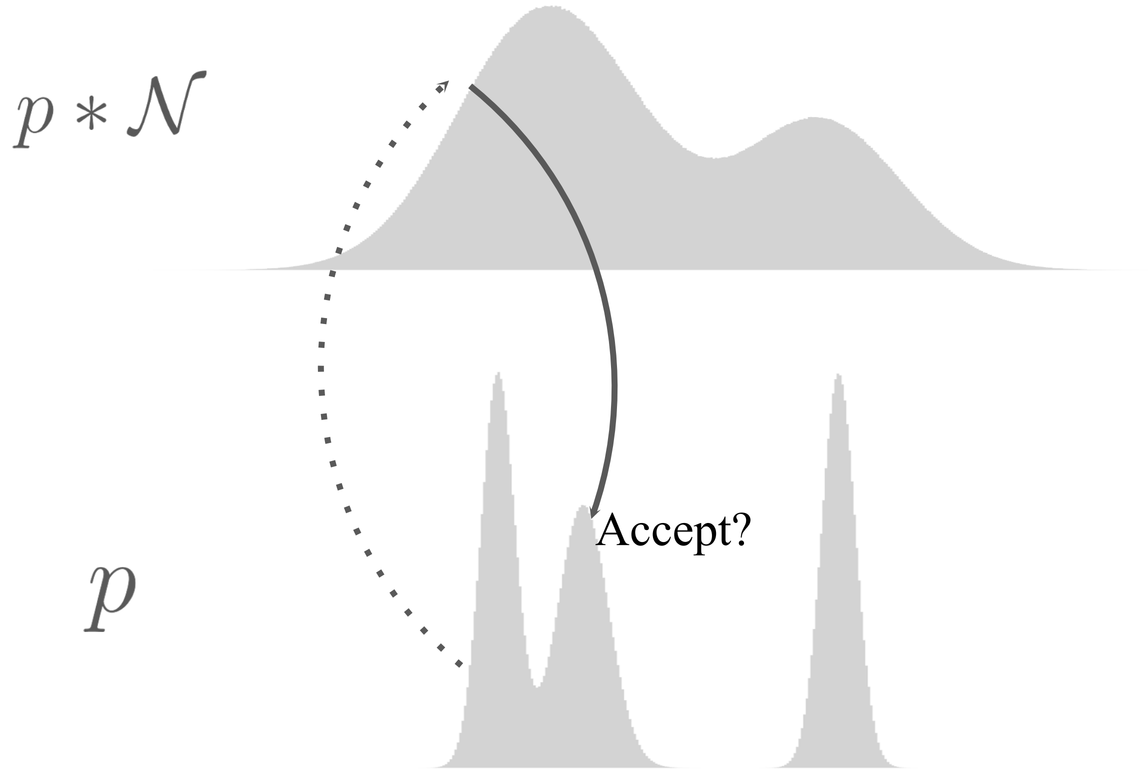
Diffusive Gibbs Sampler (with MH corrector)



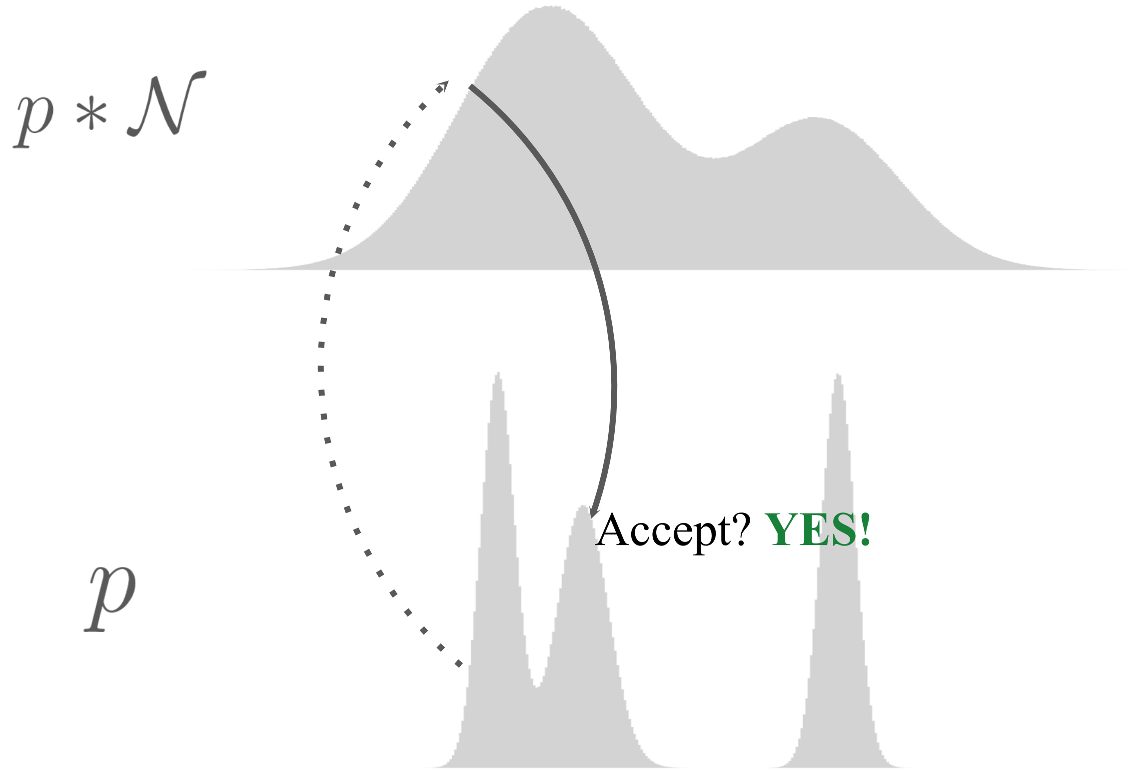
Diffusive Gibbs Sampler (with MH corrector)



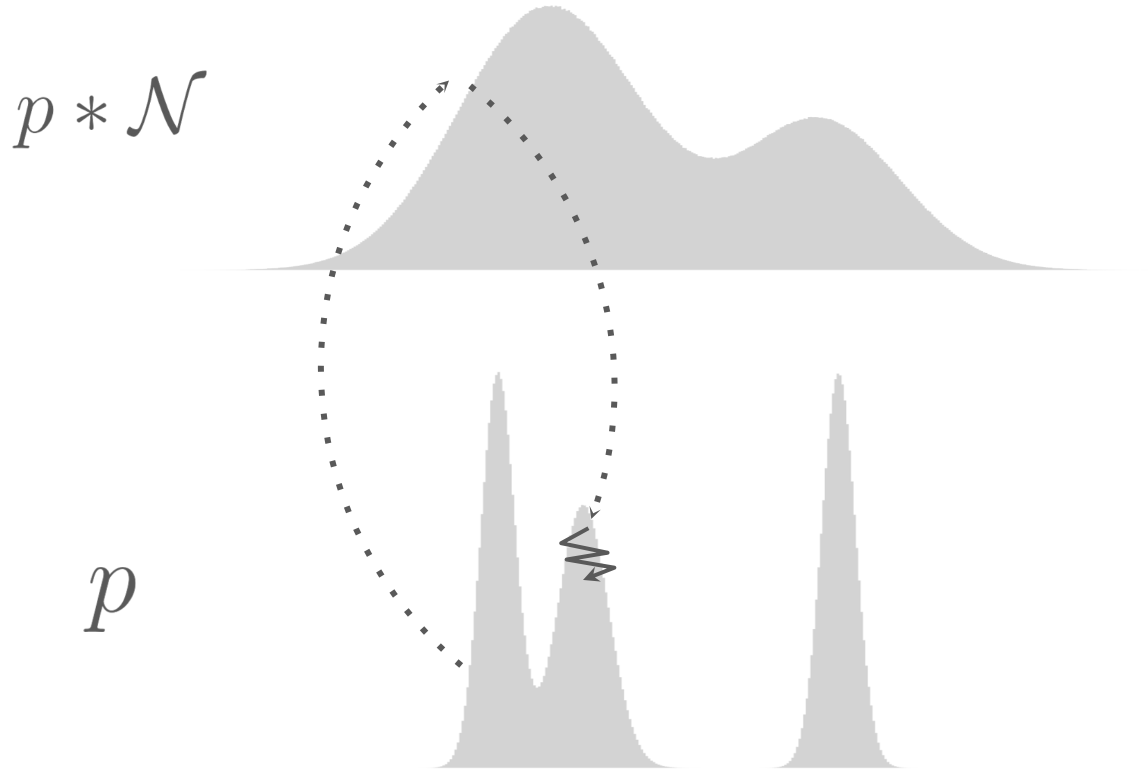
Diffusive Gibbs Sampler (with MH corrector)



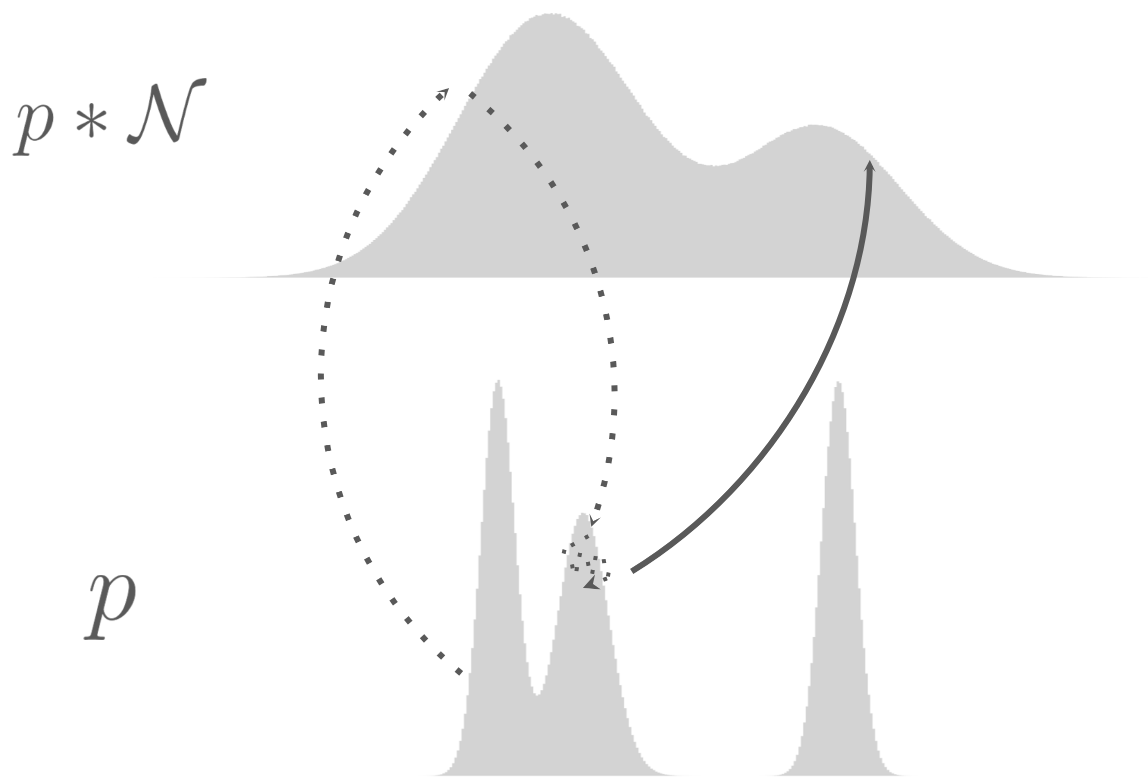
Diffusive Gibbs Sampler (with MH corrector)



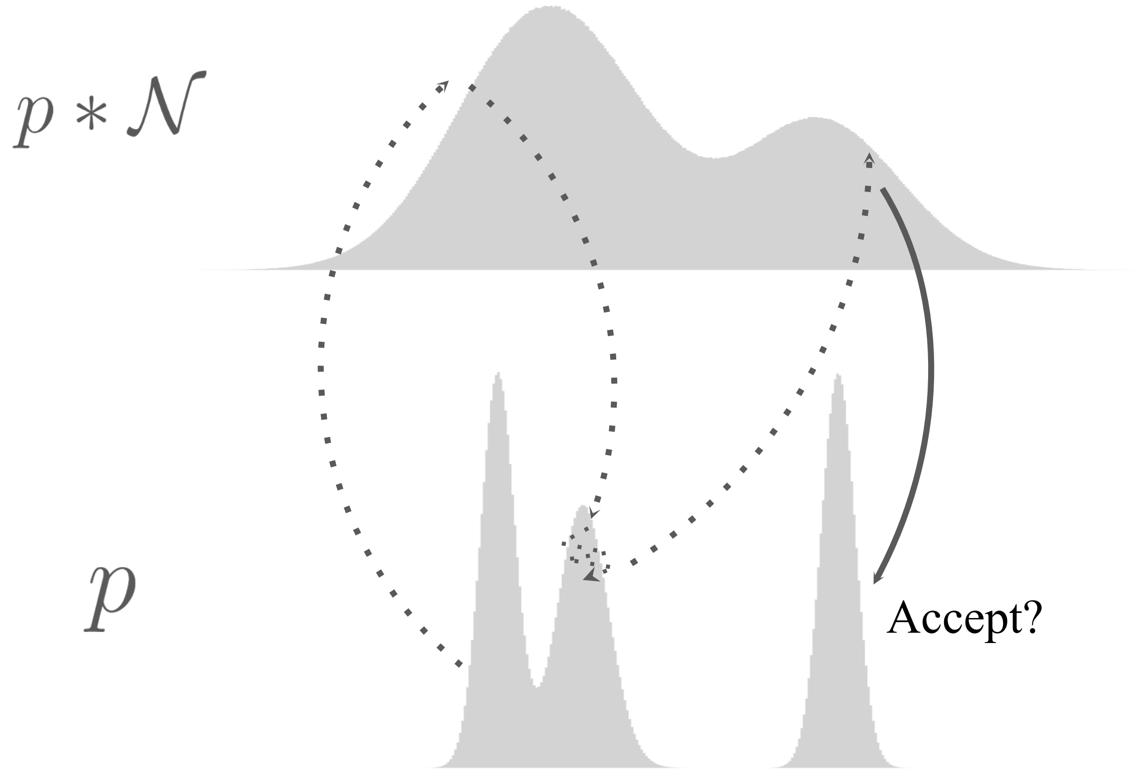
Diffusive Gibbs Sampler (with MH corrector)



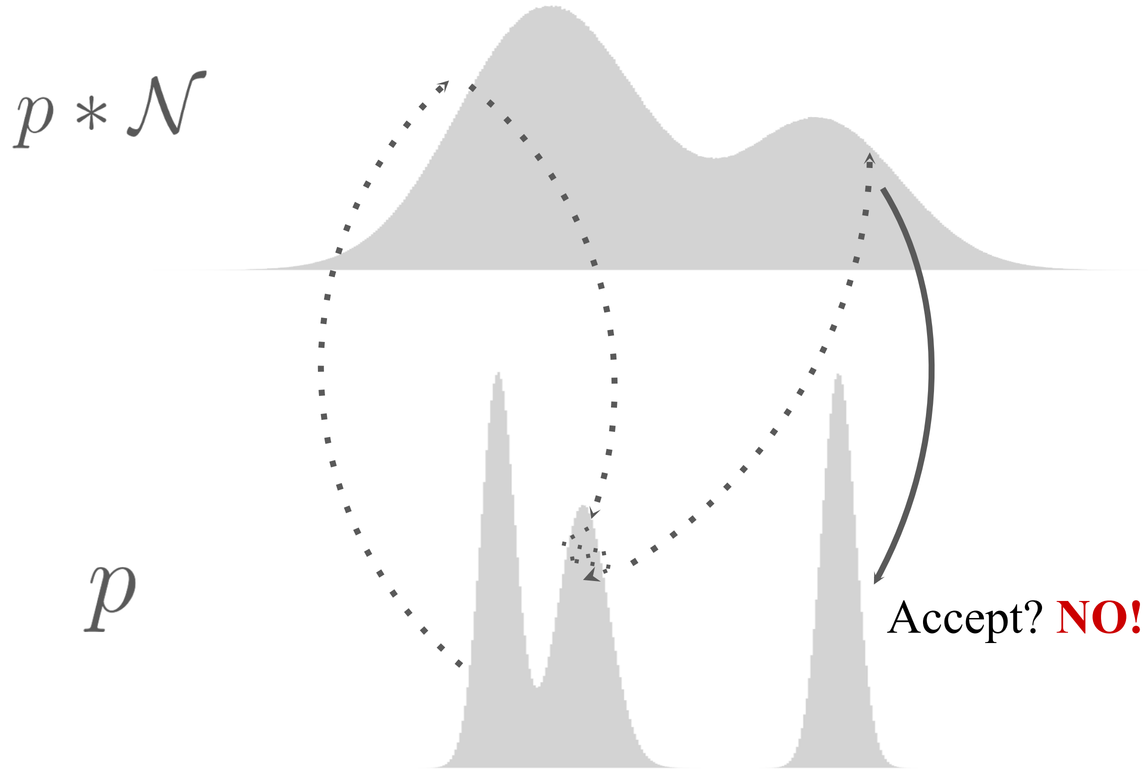
Diffusive Gibbs Sampler (with MH corrector)



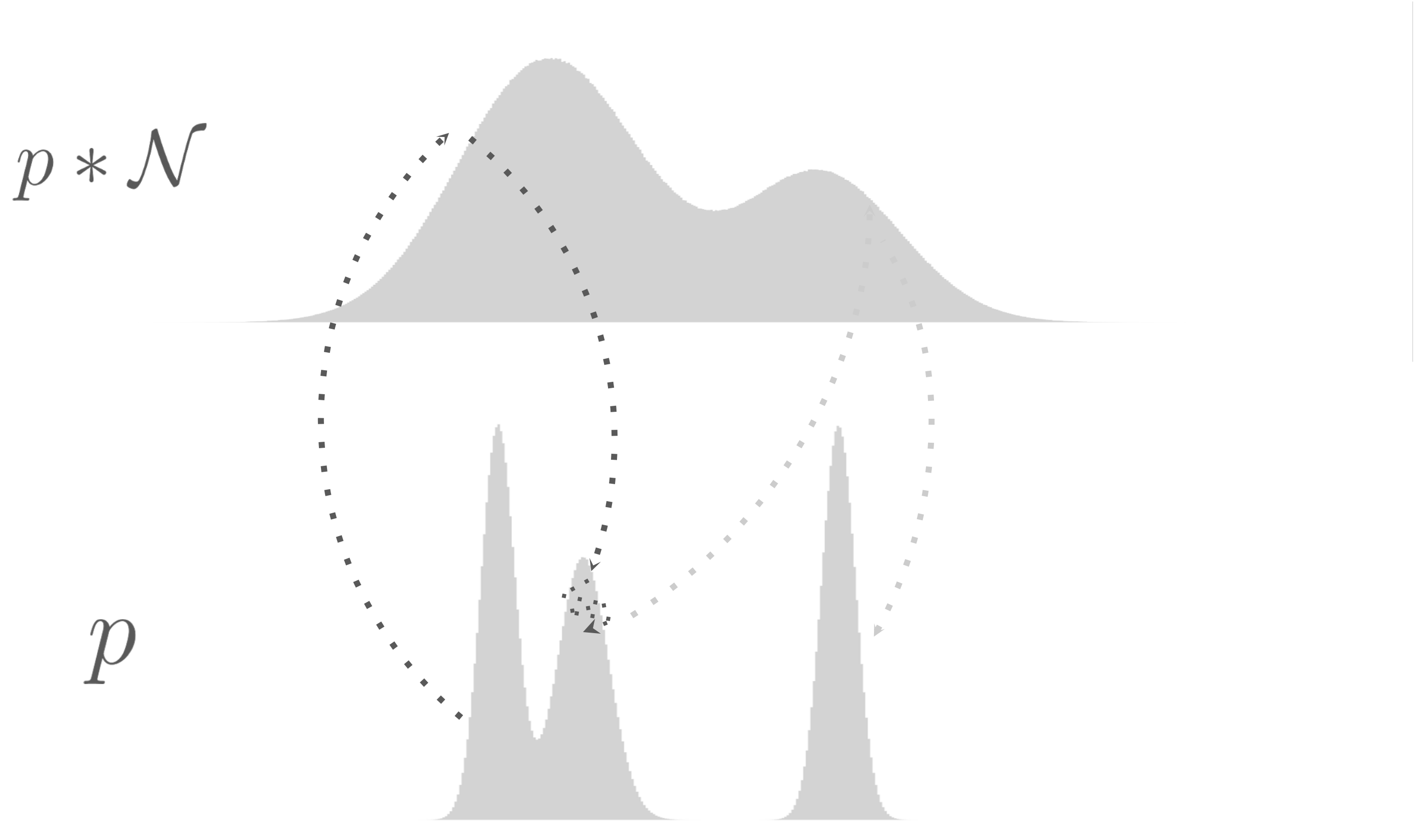
Diffusive Gibbs Sampler (with MH corrector)



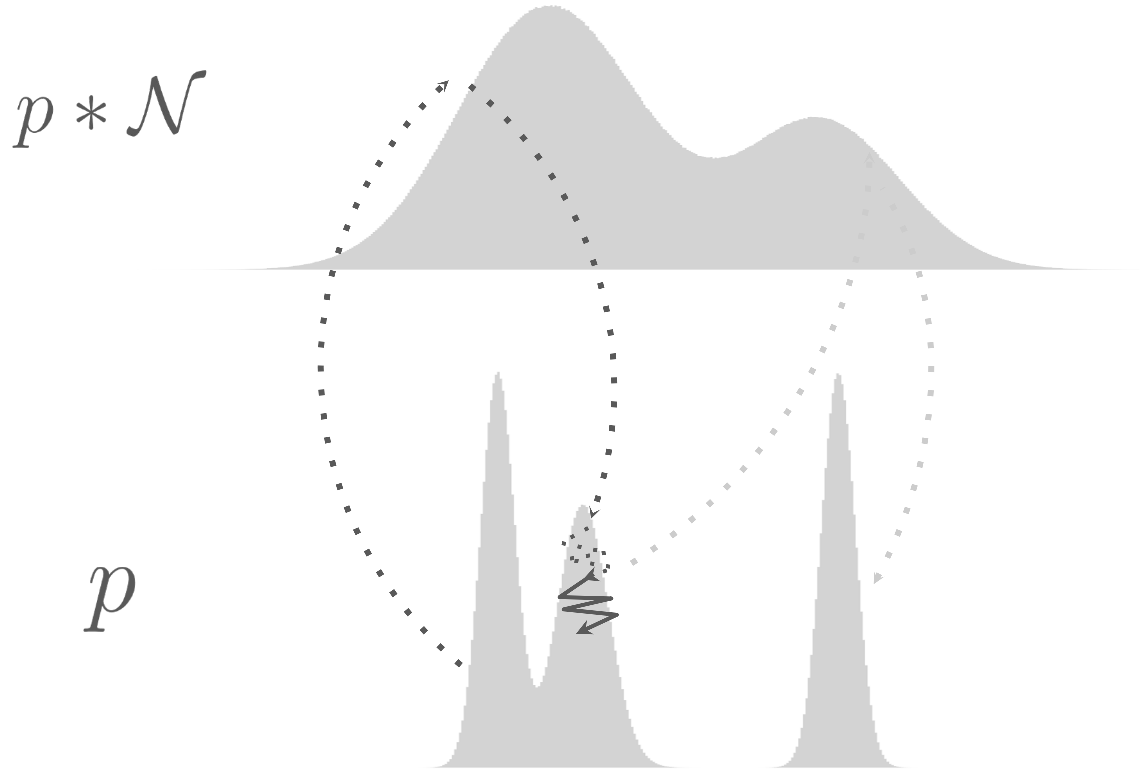
Diffusive Gibbs Sampler (with MH corrector)



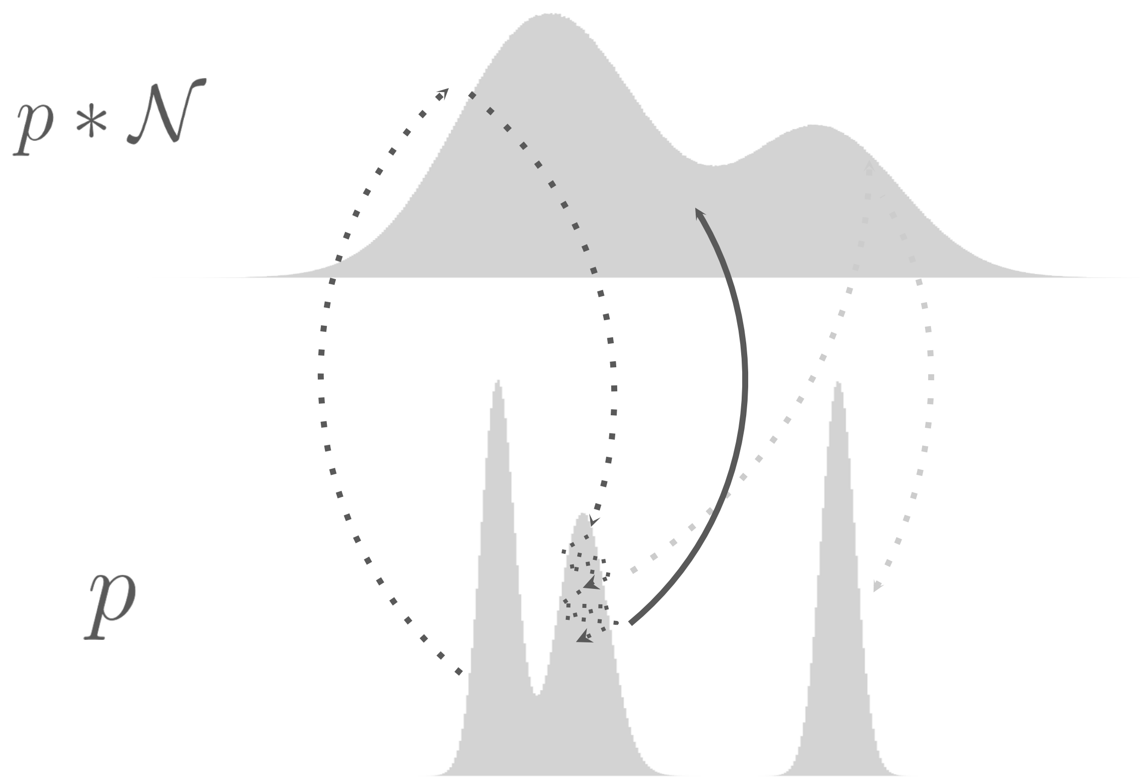
Diffusive Gibbs Sampler (with MH corrector)



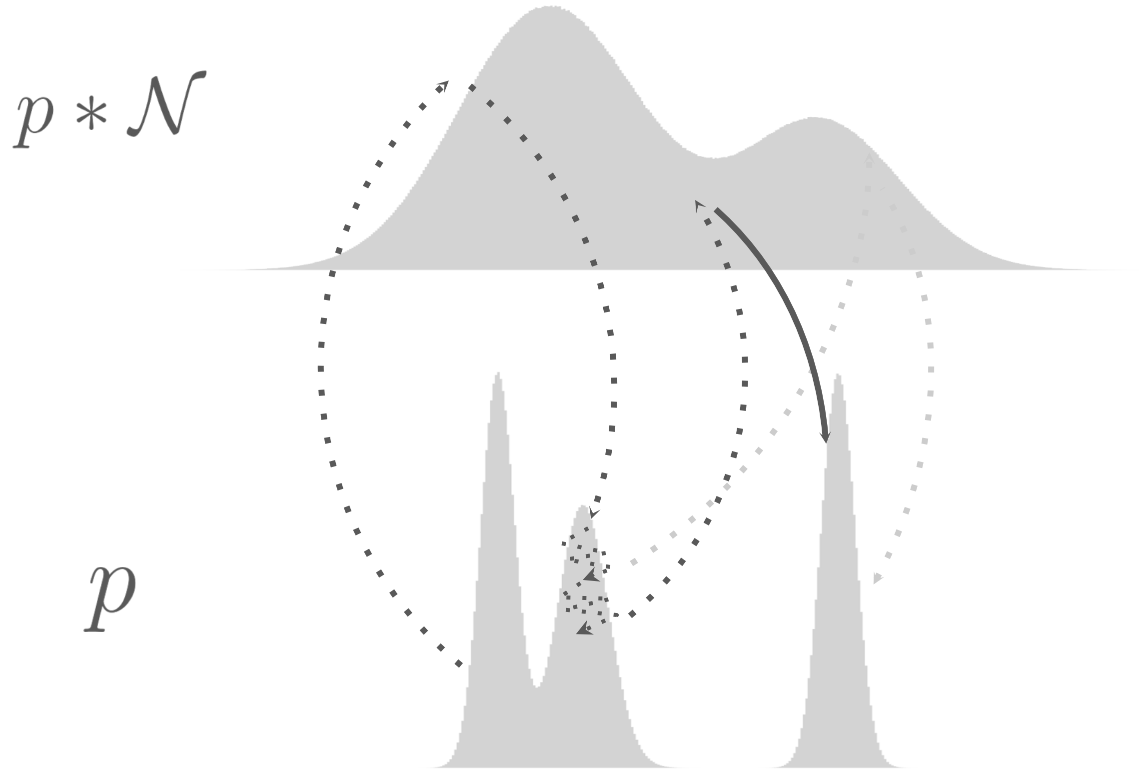
Diffusive Gibbs Sampler (with MH corrector)



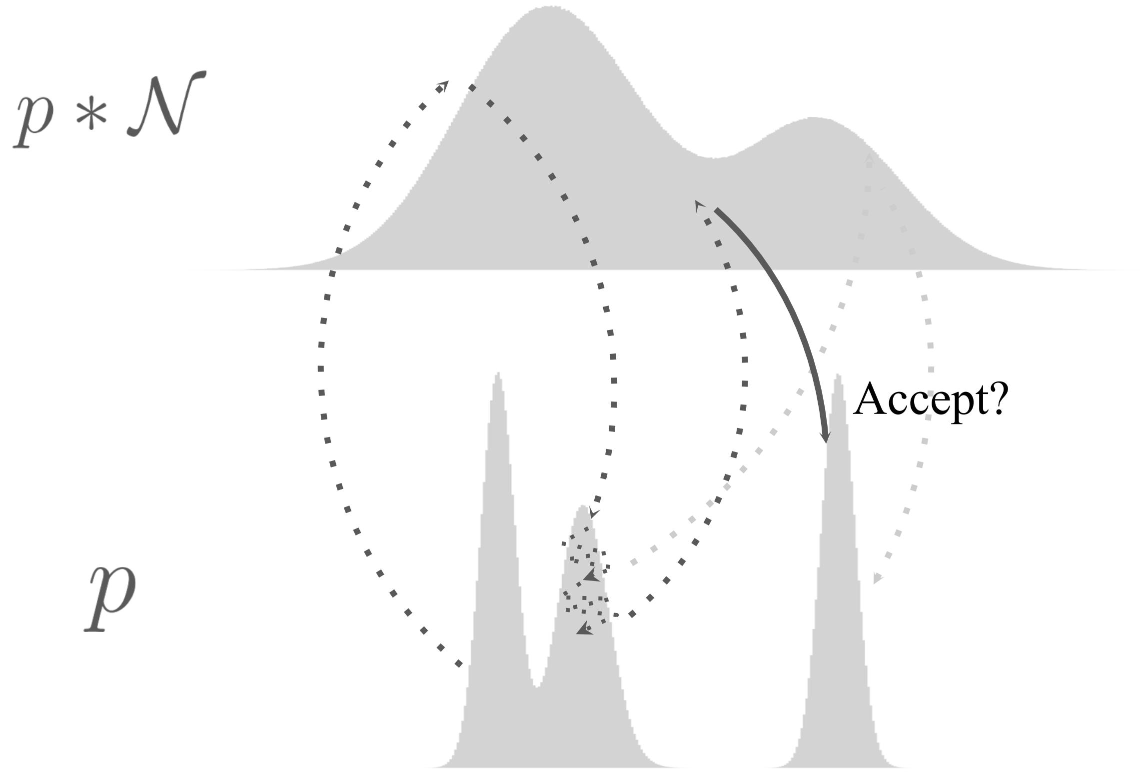
Diffusive Gibbs Sampler (with MH corrector)



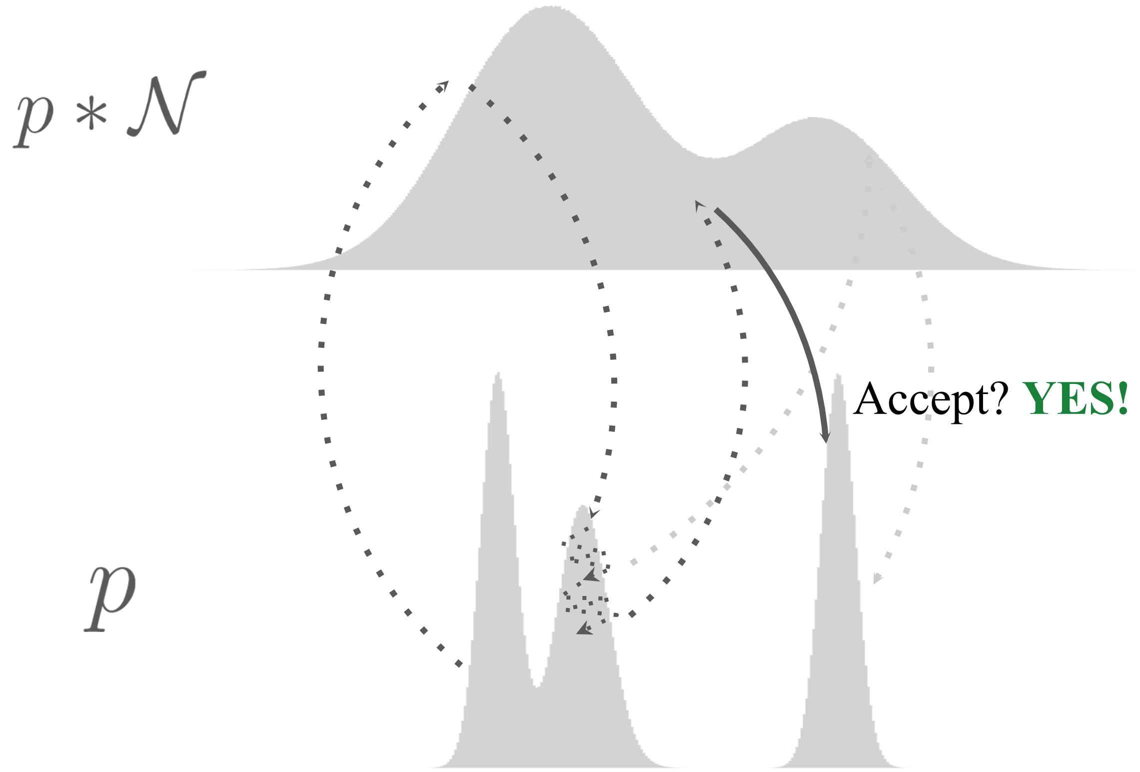
Diffusive Gibbs Sampler (with MH corrector)



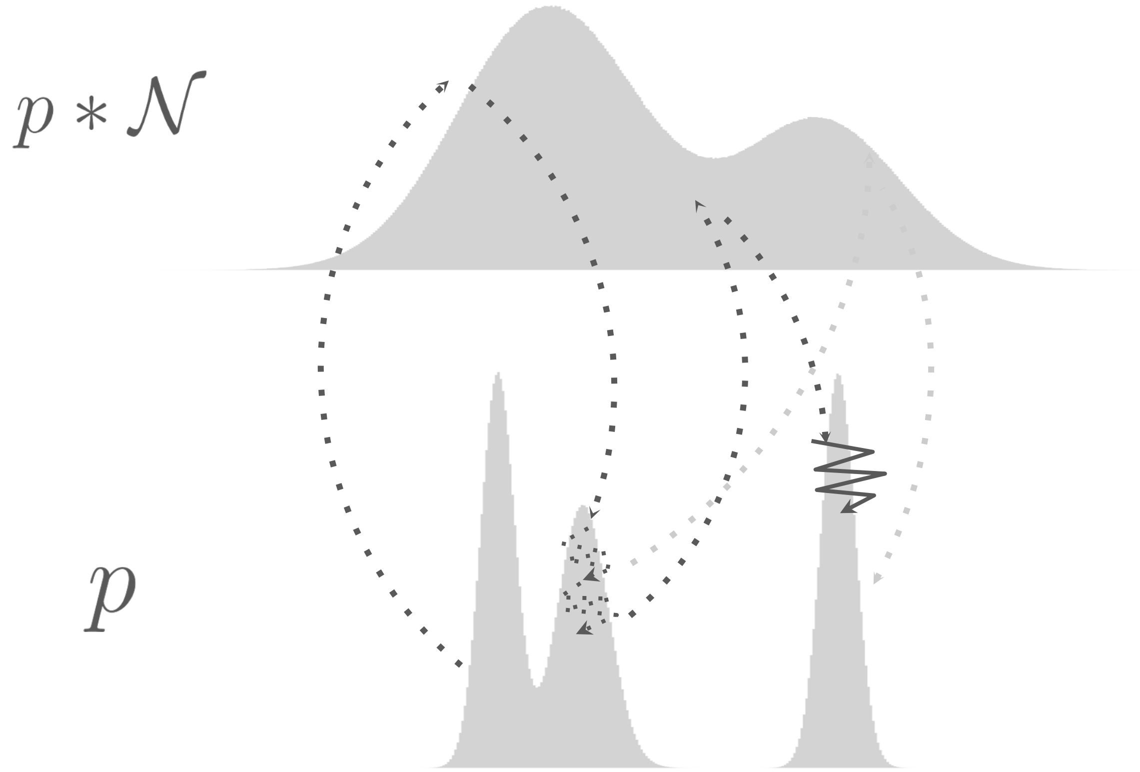
Diffusive Gibbs Sampler (with MH corrector)



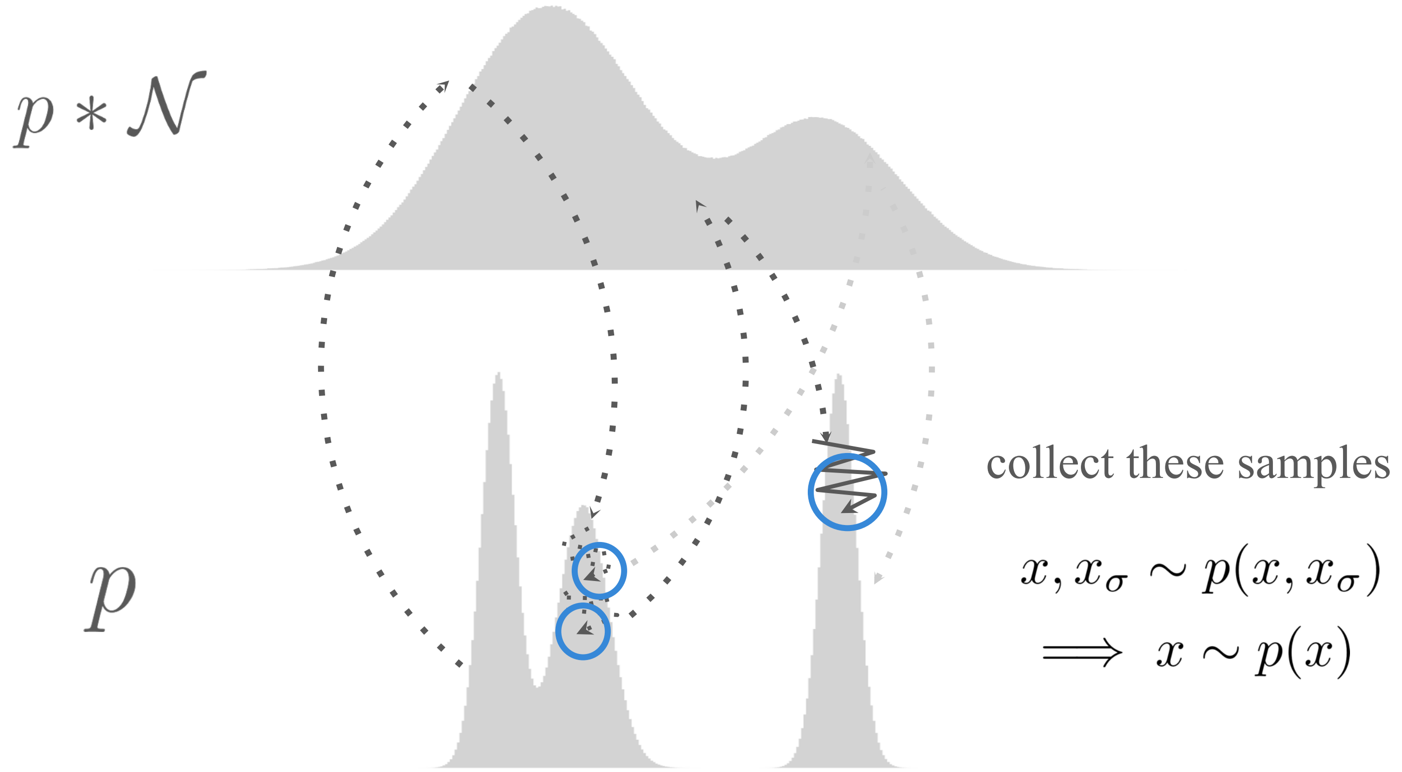
Diffusive Gibbs Sampler (with MH corrector)



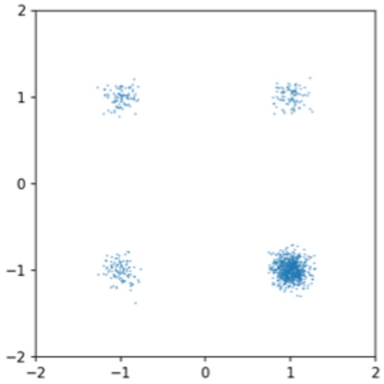
Diffusive Gibbs Sampler (with MH corrector)



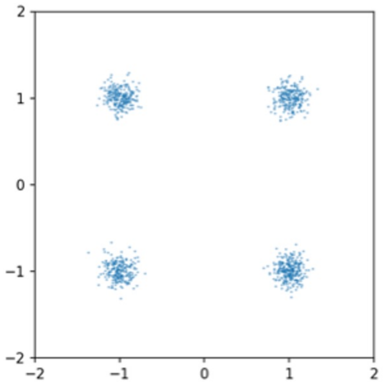
Diffusive Gibbs Sampler (with MH corrector)



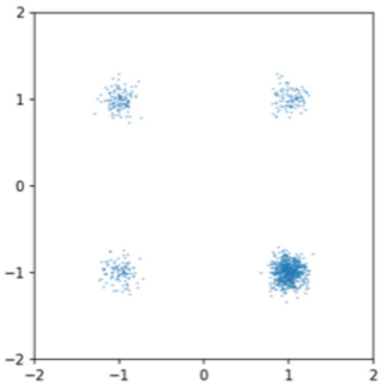
Unbalanced Modes



Ground Truth



no MH corrector



MH corrector

Choosing the Right Noise Level is Important

- Noise should not be too small

$$x \approx x_\sigma \sim \mathcal{N}(x_\sigma | x, \sigma)$$

- Noise should not be too large

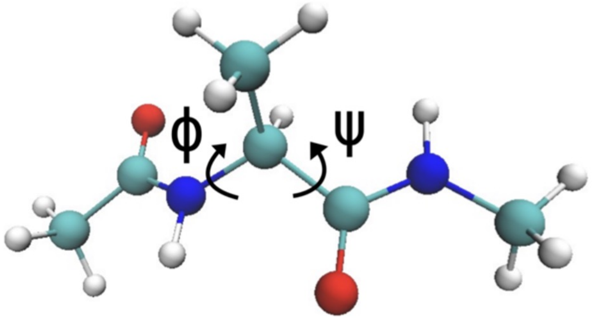
$$\boxed{p(x|x_\sigma)} \propto p(x) \boxed{\mathcal{N}(x_\sigma | x, \sigma)} \approx p(x)$$

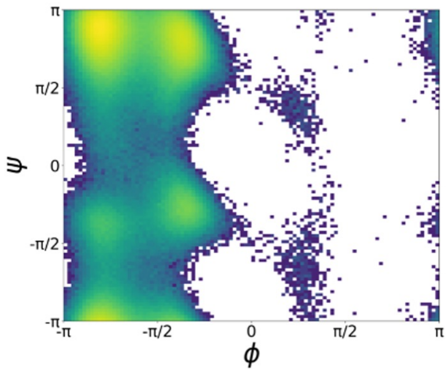
more Gaussian-like **“regularizer”**

- Use a noise schedule $\sigma_1 < \sigma_2 < \cdots < \sigma_T$

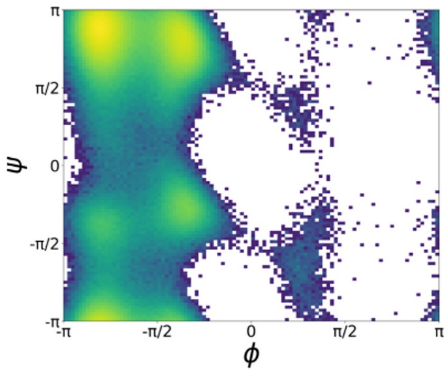
$$x \to x_{\sigma_T} \to x \to x_{\sigma_{T-1}} \to \cdots \to x \to x_{\sigma_1} \to x \to x_{\sigma_T} \to \cdots$$

Results: Alanine Dipeptide

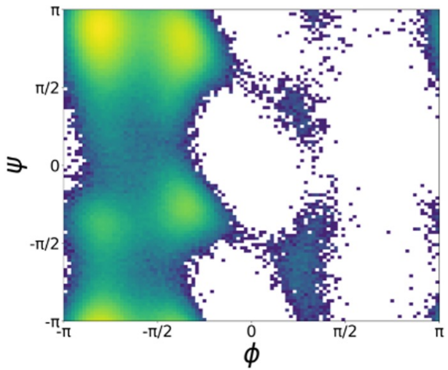




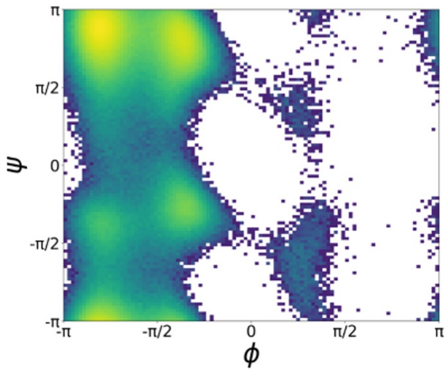
(c) MALA (10^6 samples, 1.0×10^9 energy evaluations)



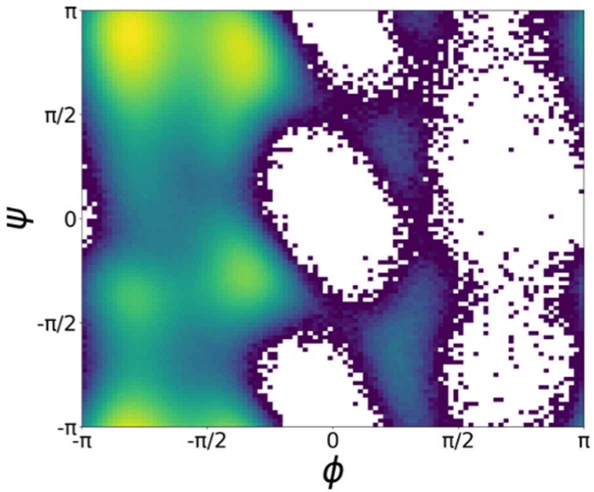
(d) HMC (10^6 samples, 1.0×10^9 energy evaluations)



(e) PT (10^6 samples, 2.3×10^{10} energy evaluations)



(f) *DiGS* (10^6 samples, 1.0×10^9 energy evaluations)



(b) MD (10^7 samples, 2.3×10^{11} energy evaluations)

Diffusive Gibbs Sampler

Limitation: samples are dependent

Can we train a neural network to generate independent samples?

Neural Sampler

Can we train a neural network to generate independent samples?

$$f_\theta : \mathcal{Z} \to \mathcal{X}$$

$$z \sim p(z), x = f_\theta(z)$$

$$p_\theta(x) = \int \delta(x - f_\theta(z)) p(z) dz$$

Without training data!

Reverse KL Divergence

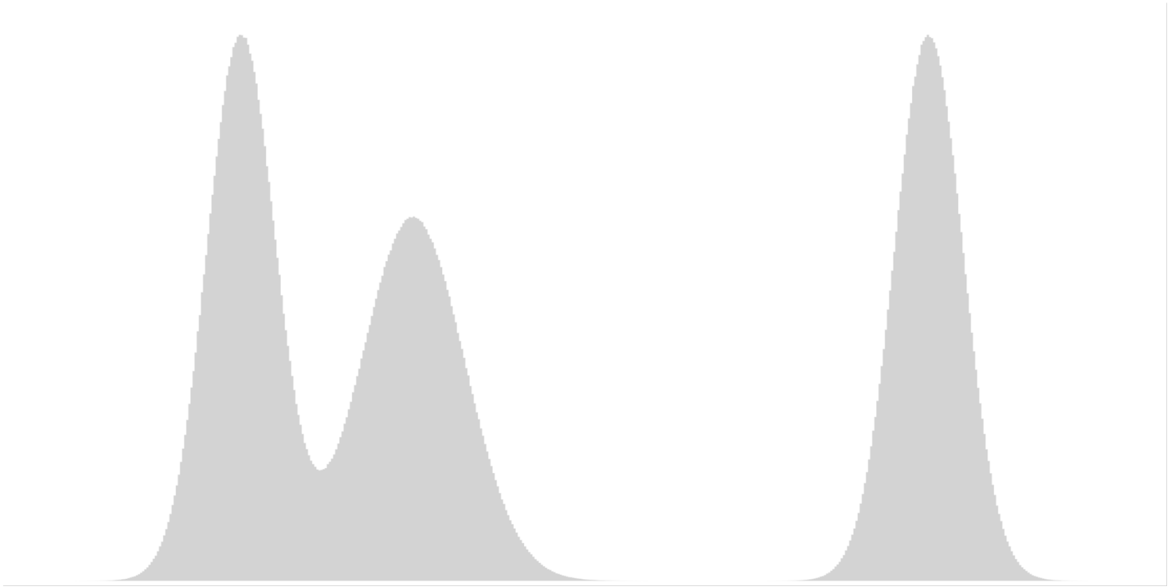
Can we train a neural network to generate independent samples?

…without training data

$$D_{\mathrm{KL}}[p_\theta || p] = \int p_\theta(x) \log \frac{p_\theta(x)}{p(x)} dx$$
$$= \int p_\theta(x) \log \frac{p_\theta(x)}{\tilde{p}(x)} dx + c.$$

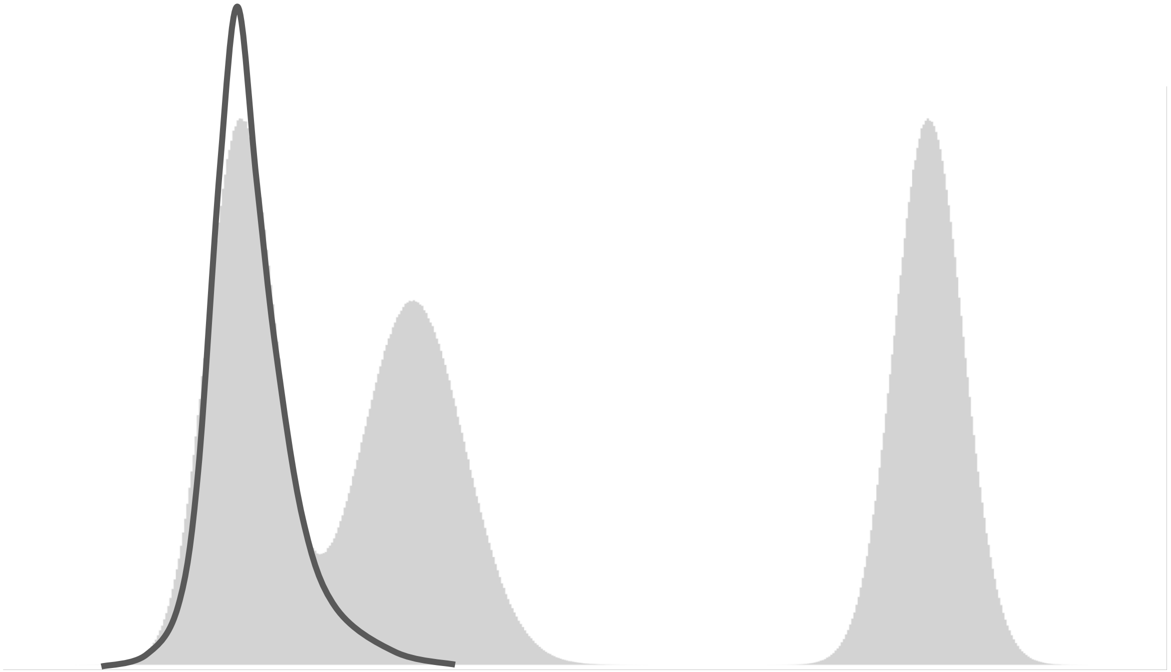
(1) Mode-seeking; (2) Intractability

Reverse KL Divergence



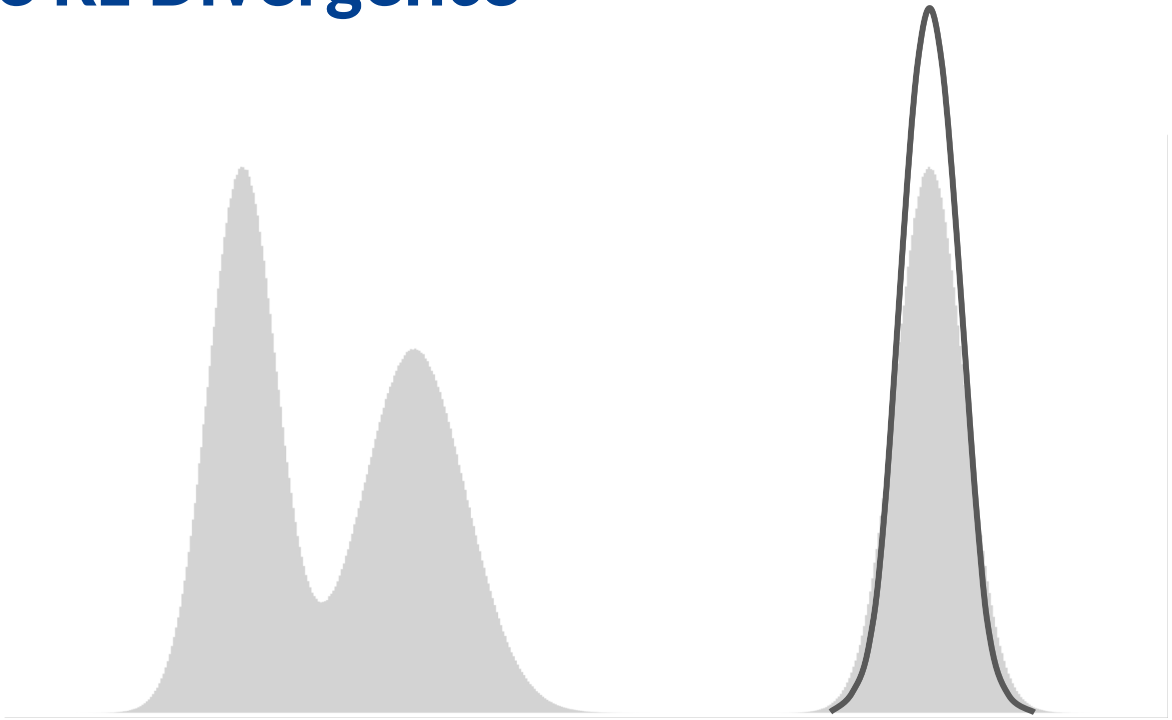
(1) Mode-seeking; (2) Intractability

Reverse KL Divergence



(1) Mode-seeking; (2) Intractability

Reverse KL Divergence

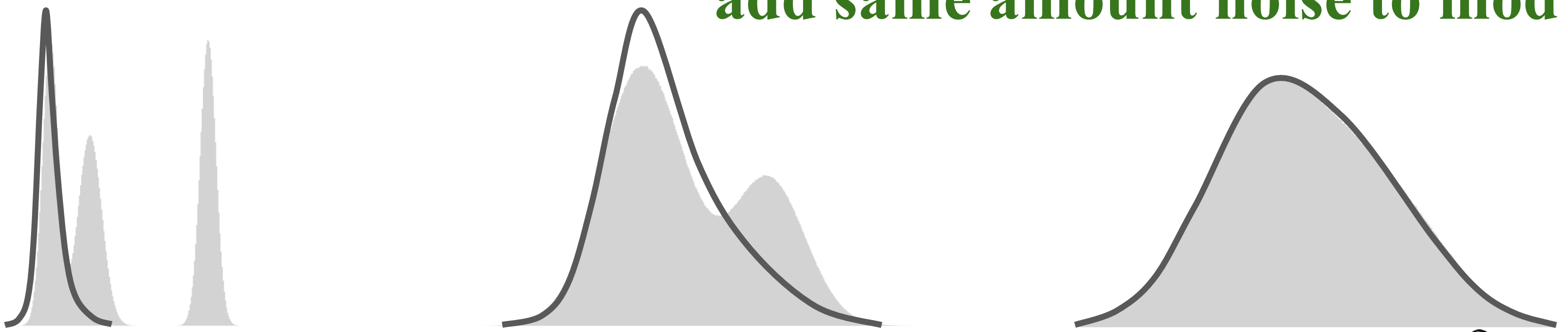


(1) Mode-seeking; (2) Intractability

How to Train Neural Samplers with Diffusion?

If trained with KL divergence… **model only learns noisy distribution!**

add same amount noise to model!



p $\quad$ $p * \mathcal{N}(0, \sigma_1^2)$ $\quad$ $p * \mathcal{N}(0, \sigma_2^2)$

?(1) Mode-seeking; **(2) Intractability**

Diffusive KL divergence

Define Gaussian noisy kernels $k_t(x_t|x) = \mathcal{N}(x_t|\alpha_t x, \sigma_t^2 I)$

$$\mathrm{DiKL}[p_\theta||p] := \sum_{t=1}^{T} w(t) D_{\mathrm{KL}}[p_\theta * k_t || p * k_t]$$

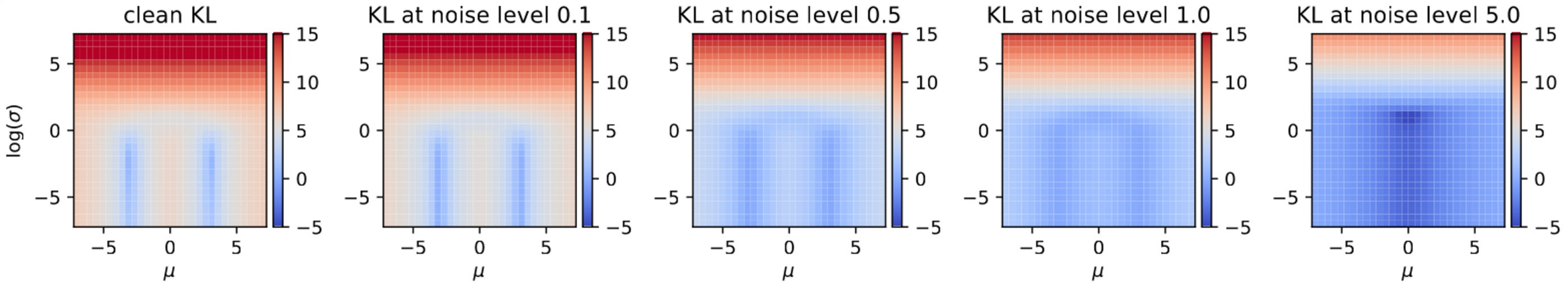
$$\mathrm{DiKL}[p||q] = 0 \Leftrightarrow p = q$$

? **(1) Mode-seeking;** **(2) Intractability**

Diffusive KL divergence

Why does it avoid mode-seeking?

Model: 1D Gaussian
Target: 1D Mixture of 2 Gaussians



- Low noise levels: refine around local mode.
- High noise levels: explore modes.

✓ **(1) Mode-seeking;** **(2) Intractability**

Gradient Estimation for Diffusive KL divergence

$$D_{\mathrm{KL}}[p_\theta * k_t || p * k_t]$$

p : target density
$\tilde{p}$: unnormalized target density
p_θ : model density
$p_{\theta,t}$: $p_\theta * k_t$
p_t : $p * k_t$

✓ (1) Mode-seeking; **(2) Intractability**

Gradient Estimation for Diffusive KL divergence

$$D_{\mathrm{KL}}[p_\theta * k_t || p * k_t] = \int p_{\theta,t}(x_t) \log \frac{p_{\theta,t}(x_t)}{p_t(x_t)} dx_t$$

✓(1) Mode-seeking; **(2) Intractability**

Gradient Estimation for Diffusive KL divergence

$$\nabla_\theta D_{\mathrm{KL}}[p_\theta * k_t || p * k_t] = \int p_{\theta,t}(x_t)(\nabla_{x_t} \log p_{\theta,t}(x_t) - \nabla_{x_t} \log p_t(x_t))\frac{\partial x_t}{\partial \theta} dx_t$$

✓ (1) Mode-seeking; **(2) Intractability**

Gradient Estimation for Diffusive KL divergence

$$\nabla_\theta D_{\mathrm{KL}}[p_\theta * k_t || p * k_t] = \int p_{\theta,t}(x_t)(\nabla_{x_t} \log p_{\theta,t}(x_t) - \nabla_{x_t} \log p_t(x_t))\frac{\partial x_t}{\partial \theta} dx_t$$

✓ (1) Mode-seeking; **(2) Intractability**

Gradient Estimation for Diffusive KL divergence

$$\nabla_\theta D_{\mathrm{KL}}[p_\theta * k_t || p * k_t] = \int p_{\theta,t}(x_t)(\nabla_{x_t} \log p_{\theta,t}(x_t) - \nabla_{x_t} \log p_t(x_t)) \frac{\partial x_t}{\partial \theta} dx_t$$

$$x_t = \alpha_t f_\theta(z) + \sigma_t \epsilon$$

auto-diff (VJP) by torch, jax, etc…

✓(1) Mode-seeking; **(2) Intractability**

Gradient Estimation for Diffusive KL divergence

$$\nabla_\theta D_{\mathrm{KL}}[p_\theta * k_t || p * k_t] = \int p_{\theta,t}(x_t)\left(\nabla_{x_t} \log p_{\theta,t}(x_t) - \nabla_{x_t} \log p_t(x_t)\right)\frac{\partial x_t}{\partial \theta} dx_t$$

- do not know model density p_θ
- but can easily generate samples from model p_θ

How to estimate this noisy score given only model samples?

Train a diffusion model to approximate model score!

$$\min_\phi \iint \|s_\phi(x_t, t) - \nabla_{x_t} \log k_t(x_t|x)\|^2 p_\theta(x) k_t(x_t|x) dx_t dx$$

✓ (1) Mode-seeking; **(2) Intractability**

Gradient Estimation for Diffusive KL divergence

$$\nabla_\theta D_{\mathrm{KL}}[p_\theta * k_t || p * k_t] = \int p_{\theta,t}(x_t)(\nabla_{x_t} \log p_{\theta,t}(x_t) - \nabla_{x_t} \log p_t(x_t)) \frac{\partial x_t}{\partial \theta} dx_t$$

- know p (up to some normalization constant)

Score Identity:

can be sampled **available**

$$\nabla_{x_t} \log p_t(x_t) = \int p(x|x_t)(\alpha_t(x + \nabla_x \log p(x)) - x_t)\, dx$$

$p(x|x_t) \propto \tilde{p}(x)k_t(x_t|x)$ As before, use MALA/HMC/AIS, …

(1) Mode-seeking; (2) Intractability

Training Neural Sampler with Diffusive KL divergence

$$\nabla_\theta D_{\mathrm{KL}}[p_\theta * k_t || p * k_t] = \int p_{\theta,t}(x_t)(\nabla_{x_t} \log p_{\theta,t}(x_t) - \nabla_{x_t} \log p_t(x_t))\frac{\partial x_t}{\partial \theta} dx_t$$

(1) Mode-covering:

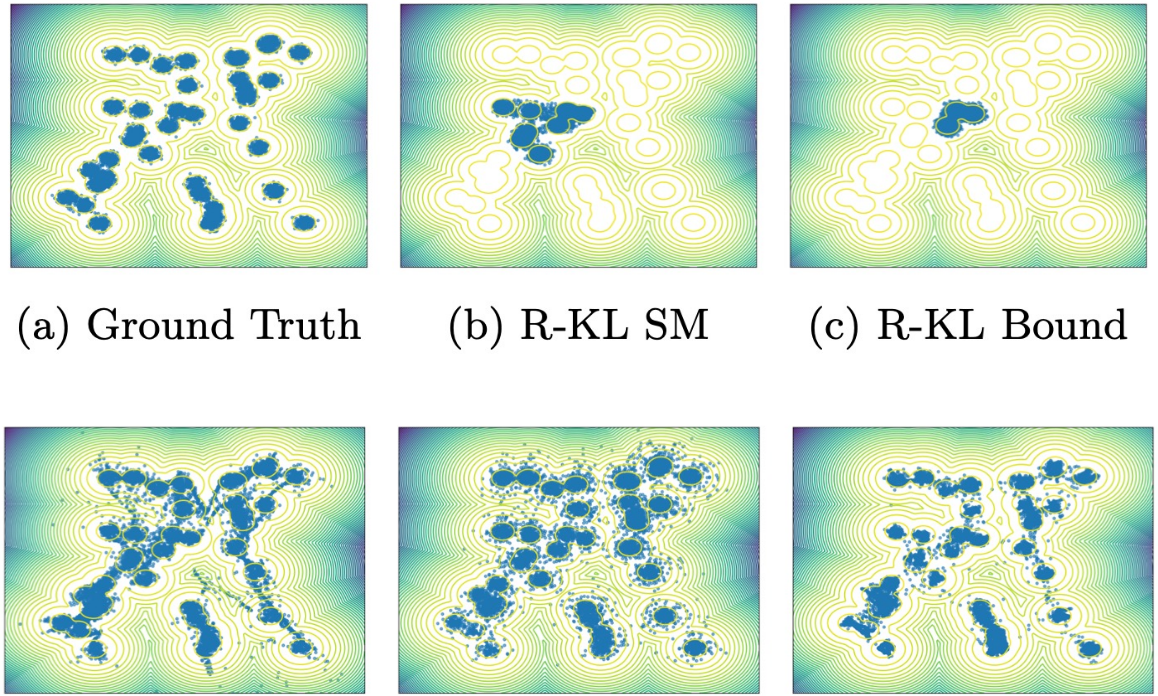
- match KL divergence at different noise levels

(2) Tractable:

- estimate noisy model score by training a diffusion model
- estimate noisy target score by score identity with Monte Carlo

Expectation-Maximization (EM) Style Model Training!

Results: Mixture of 40 Gaussians

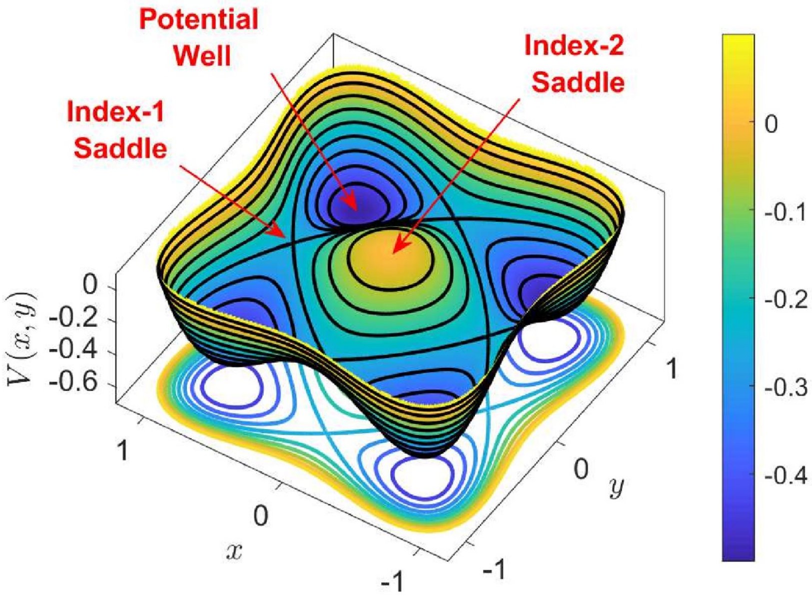


(a) Ground Truth (b) R-KL SM (c) R-KL Bound

(d) FAB (e) iDEM (f) DiKL (ours)

Results: Many Well 32 Potential Energy

Highly multi-modal: 2^{32} modes in total obtained by stacking double well 32 times.



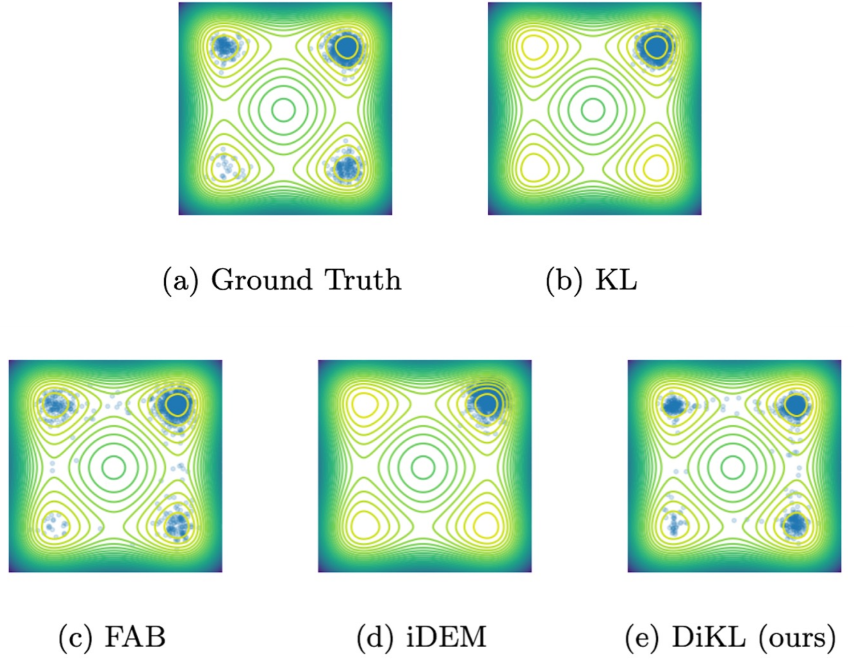
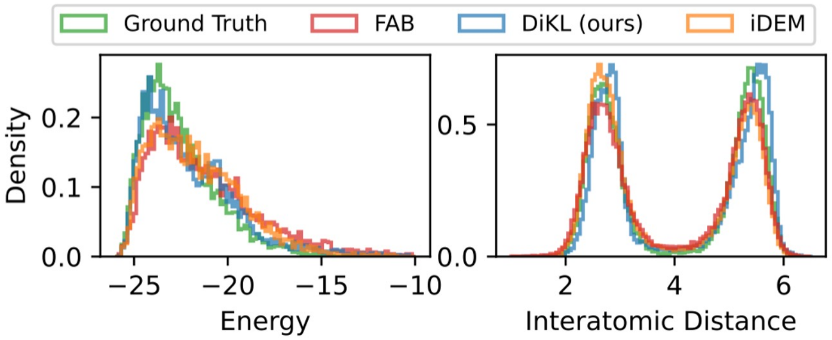


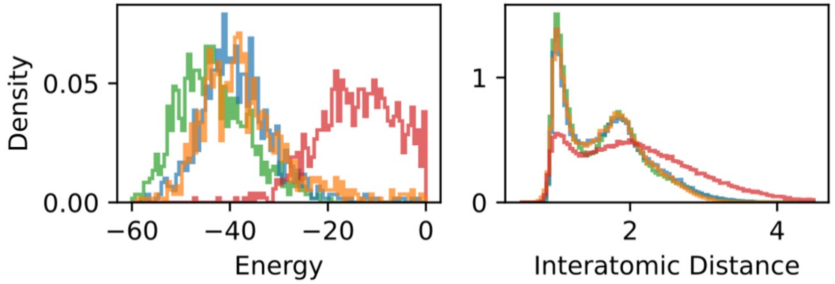
Image from https://www.chemicalreactions.io/act2/four_well_morse/four_well_morse-jekyll.html

Results: n-body Systems

Double-Well-4



Lennard-Jones-13



		FAB	**iDEM**	**DiKL (ours)**
Training	MW-32	3.5h	3.5h	**2.5h**
	DW-4	4.5h	4.5h	**0.8h**
	LJ-13	21.5h	6.5h	**3h**
Batch Sampling (1,000 samples)	MW-32	**0.01s**	7.2s	**0.01s**
	DW-4	-	2.6s	**0.01s**
	LJ-13	-	19.7s	**0.02s**

Bottleneck of Diffusion-Inspired Samplers

sampling from $p(x|x_{\text{noise}})$

- **Unsatisfactory:** denoising posterior sampling could still be hard
- **Inevitable:** no data is available to train a denoiser network

Reference and Collabroators

- **Diffusive Gibbs Sampling**
  Wenlin Chen*, Mingtian Zhang*, Brooks Paige, José Miguel Hernández-Lobato, David Barber
  International Conference on Machine Learning (ICML), 2024.

- **Training Neural Samplers with Reverse Diffusive KL Divergence**
  Jiajun He*, Wenlin Chen*, Mingtian Zhang*, David Barber, José Miguel Hernández-Lobato
  International Conference on Artificial Intelligence and Statistics (AISTATS), 2025.


Jiajun He


Mingtian Zhang


Brooks Paige


David Barber


Miguel Hernández-Lobato