Generative models to assist sampling: A tentative tutorial

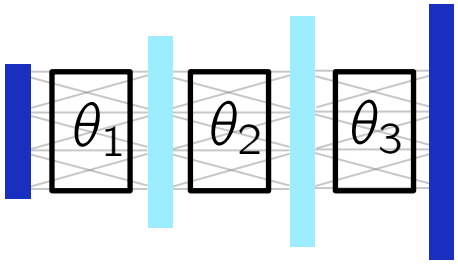
November 27-December 1st 2023
IAP, Paris / Flatiron institute, New York

Marylou Gabrié
CMAP, École Polytechnique

Deep generative models based on transport

▷ Deep latent generative models

$$T_\theta : \mathbb{R}^b \to \mathbb{R}^d$$

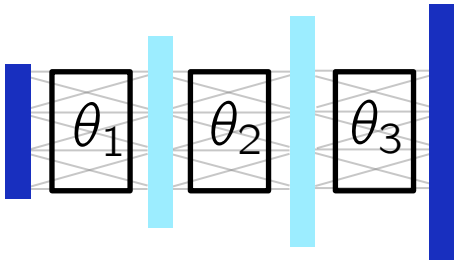
$z \sim \rho_{\mathrm{B}}(z)$  $x = T_\theta(z) \quad \sim \rho_\theta(x)$ "push-forward" distribution

["Normalizing flows" Tabak & V.-E. *Commun. Math. Sci.* 2010, Dinh et al *ICLR* 2017, Papamakarios et al. *JMLR* 2021, "Score based diffusion models" Song et al. *ICLR 2021,* "Stochastic interpolants [..]" Albergo et al. arxiv:2303.08797 *etc.*]

Deep generative models based on transport

▷ Deep latent generative models

$$T_\theta : \mathbb{R}^b \to \mathbb{R}^d$$

$z \sim \rho_B(z)$  $x = T_\theta(z) \quad \sim \rho_\theta(x)$ "push-forward" distribution

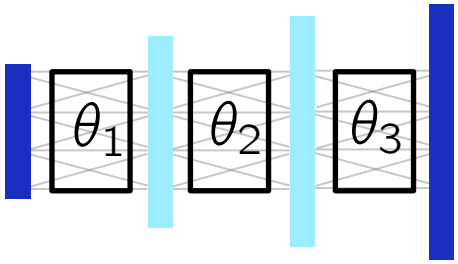
Song et al. *ICLR* 2021

["Normalizing flows" Tabak & V.-E. *Commun. Math. Sci.* 2010, Dinh et al *ICLR* 2017, Papamakarios et al. *JMLR* 2021, "Score based diffusion models" Song et al. *ICLR 2021,* "Stochastic interpolants [..]" Albergo et al. arxiv:2303.08797 *etc.*]

Deep generative models based on transport

▷ Deep latent generative models

$$T_\theta : \mathbb{R}^b \to \mathbb{R}^d$$

$z \sim \rho_B(z)$  $x = T_\theta(z) \quad \sim \rho_\theta(x)$ "push-forward" distribution

New dog picture for each new base variable!

Song et al. *ICLR* 2021

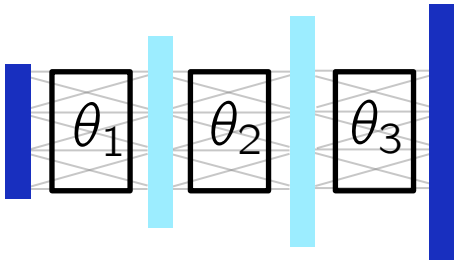
["Normalizing flows" Tabak & V.-E. *Commun. Math. Sci.* 2010, Dinh et al *ICLR* 2017, Papamakarios et al. *JMLR* 2021, "Score based diffusion models" Song et al. *ICLR 2021,* "Stochastic interpolants [..]" Albergo et al. arxiv:2303.08797 *etc.*]

Deep generative models based on transport

▷ Deep latent generative models

$$T_\theta : \mathbb{R}^b \to \mathbb{R}^d$$

$$z \sim \rho_B(z)$$



$$x = T_\theta(z) \quad \sim \rho_\theta(x)$$

"push-forward" distribution

New dog picture for each new base variable!

Song et al. *ICLR* 2021

▷ Invertible map/transport based models: $T_\theta : \Omega \mapsto \Omega$

(Normalizing flows (discrete time), Neural ODEs, Score-based diffusion models, etc.)

$$\rho_\theta(x) = \rho_B(T_\theta^{-1}(x)) \det \left| \nabla_x T_\theta^{-1} \right|$$

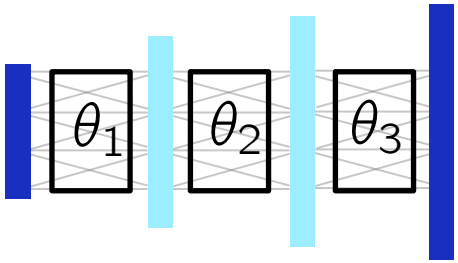
["Normalizing flows" Tabak & V.-E. *Commun. Math. Sci.* 2010, Dinh et al *ICLR* 2017, Papamakarios et al. *JMLR* 2021, "Score based diffusion models" Song et al. *ICLR 2021,* "Stochastic interpolants [..]" Albergo et al. arxiv:2303.08797 *etc.*]

Deep generative models based on transport

▷ Deep latent generative models

$$T_\theta : \mathbb{R}^b \to \mathbb{R}^d$$

$$z \sim \rho_B(z)$$



$$x = T_\theta(z) \quad \sim \rho_\theta(x)$$

"push-forward" distribution

New dog picture for each new base variable!

Song et al. *ICLR* 2021

▷ Invertible map/transport based models: $T_\theta : \Omega \mapsto \Omega$

(Normalizing flows (discrete time), Neural ODEs, Score-based diffusion models, etc.)

$$\rho_\theta(x) = \rho_B(T_\theta^{-1}(x)) \det \left| \nabla_x T_\theta^{-1} \right|$$

▷ Training

- From data samples $\{x_i\}_{i=1}^N$ assumed to be i.i.d
  - Maximum likelihood: minimize $L[\rho_\theta] = -\sum_{i=1}^N \log \rho_\theta(x_i)$
  - Score based objectives $L[\theta] = \sum_{i=1}^N \|\nabla_x \log \rho_\theta(x_i)\|^2 + 2\Delta_x \log \rho_\theta(x_i)$

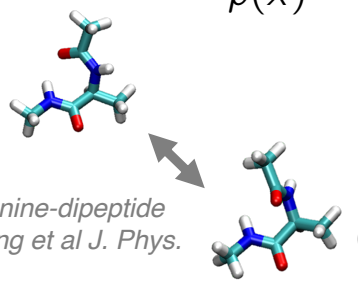
["Normalizing flows" Tabak & V.-E. *Commun. Math. Sci.* 2010, Dinh et al *ICLR* 2017, Papamakarios et al. *JMLR* 2021, "Score based diffusion models" Song et al. *ICLR 2021,* "Stochastic interpolants [..]" Albergo et al. arxiv:2303.08797 *etc.*]

High-dimensional inference

- A distribution of interest **known up to normalization** (posterior/Boltzmann)

ex: molecular configurations

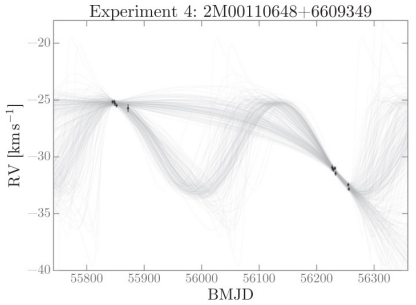
$$\rho(x) = \frac{1}{\mathcal{Z}_\beta} e^{-\beta U(x)}$$



Alanine-dipeptide
Jiang et al J. Phys.

ex: Bayesian model parameters

$$\rho(x|D) = \frac{1}{\mathcal{Z}_D} \rho(D|x)\rho(x)$$



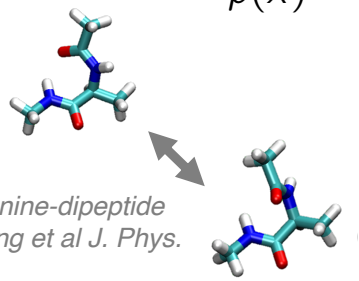
Price-Whelan et al. *The Astrophysical Journal* 2017

High-dimensional inference

▷ A distribution of interest **known up to normalization** (posterior/Boltzmann)

ex: molecular configurations

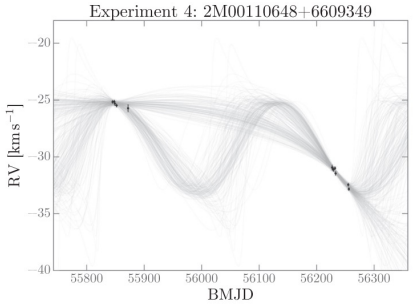
$$\rho(x) = \frac{1}{\mathcal{Z}_\beta} e^{-\beta U(x)}$$



Alanine-dipeptide
Jiang et al J. Phys.

ex: Bayesian model parameters

$$\rho(x|D) = \frac{1}{\mathcal{Z}_D} \rho(D|x)\rho(x)$$



Price-Whelan et al. *The Astrophysical Journal* 2017

▷ Direct inference is intractable $\mathbb{E}_\rho[f(x)] = \int_\Omega f(x)\rho(x)\mathrm{d}x$, costs $O(e^d)$ for $x \in \mathbb{R}^d$

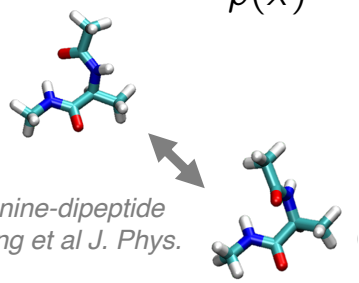
▷ Monte Carlo methods rely on samples: $\mathbb{E}_\rho[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$ for $x_i \sim \rho(x)$

High-dimensional inference

- A distribution of interest **known up to normalization** (posterior/Boltzmann)

ex: molecular configurations

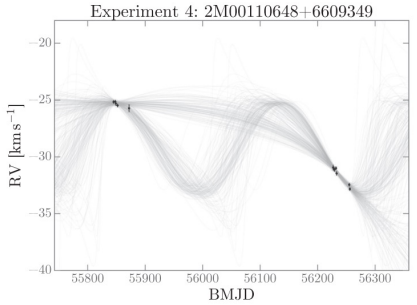
$$\rho(x) = \frac{1}{\mathcal{Z}_\beta} e^{-\beta U(x)}$$



Alanine-dipeptide
Jiang et al J. Phys.

ex: Bayesian model parameters

$$\rho(x|D) = \frac{1}{\mathcal{Z}_D} \rho(D|x)\rho(x)$$



Price-Whelan et al. *The Astrophysical Journal* 2017

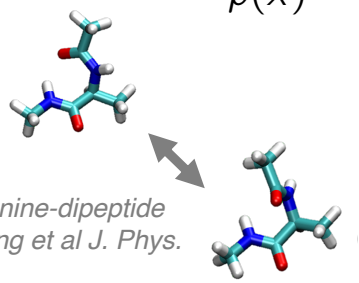
- Direct inference is intractable $\mathbb{E}_\rho[f(x)] = \int_\Omega f(x)\rho(x)\mathrm{d}x$, costs $O(e^d)$ for $x \in \mathbb{R}^d$
- Monte Carlo methods rely on samples: $\mathbb{E}_\rho[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$ for $x_i \sim \rho(x)$
- Sampling itself can be challenging (dimensionality, geometry, multimodality)

High-dimensional inference

- A distribution of interest **known up to normalization** (posterior/Boltzmann)

ex: molecular configurations

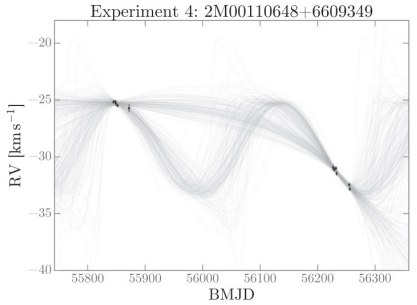
$$\rho(x) = \frac{1}{\mathcal{Z}_\beta} e^{-\beta U(x)}$$



Alanine-dipeptide
Jiang et al J. Phys.

ex: Bayesian model parameters

$$\rho(x|D) = \frac{1}{\mathcal{Z}_D} \rho(D|x)\rho(x)$$



Price-Whelan et al. *The Astrophysical Journal* 2017

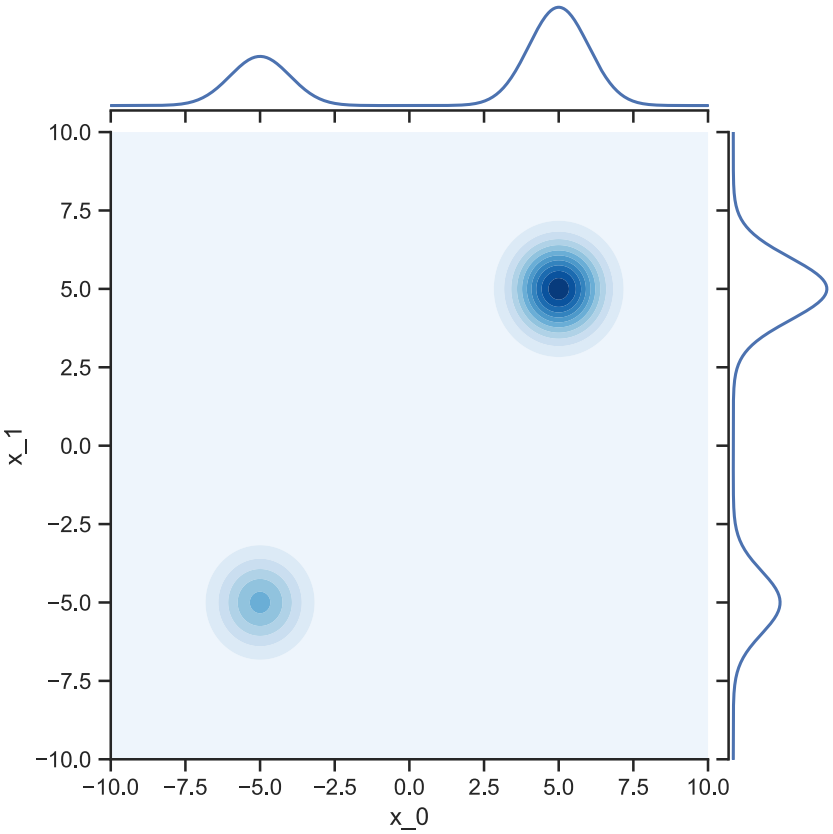
- Direct inference is intractable $\mathbb{E}_\rho[f(x)] = \int_\Omega f(x)\rho(x)\mathrm{d}x$, costs $O(e^d)$ for $x \in \mathbb{R}^d$
- Monte Carlo methods rely on samples: $\mathbb{E}_\rho[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$ for $x_i \sim \rho(x)$
- Sampling itself can be challenging (dimensionality, geometry, multimodality)
- **This talk:** How we use deep generative models to speed up sampling?

Consider simple samplers

- Importance sampling

  rely on a tractable proposal

$$\mathbb{E}_{\rho}[f(x)] = \int_{\Omega} f(x)\rho(x)\mathrm{d}x$$

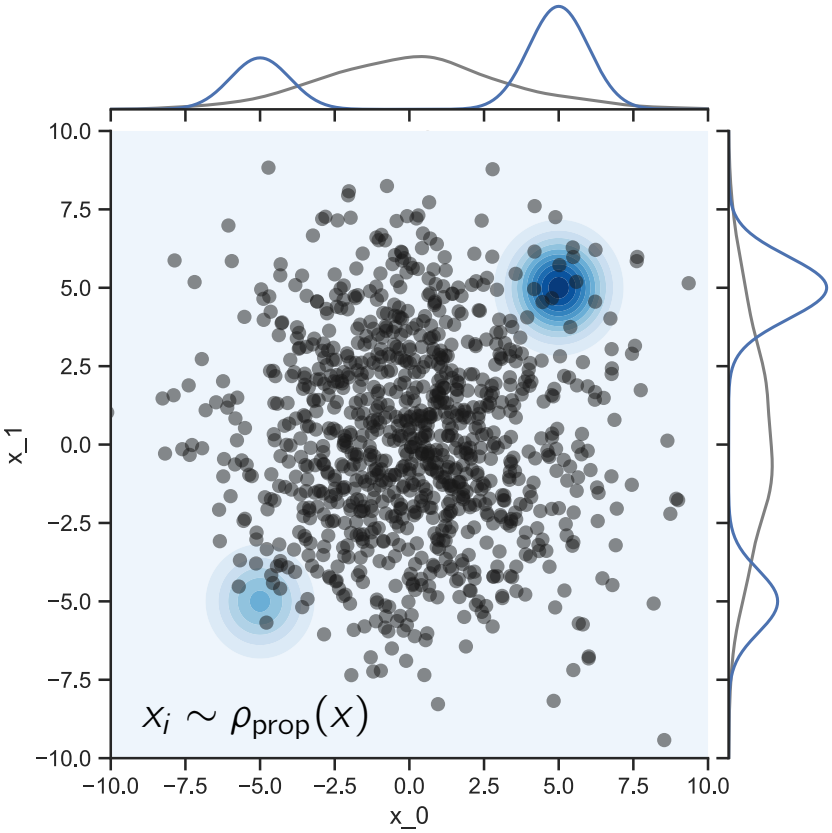


Consider simple samplers

- Importance sampling

  rely on a tractable proposal

$$\mathbb{E}_{\rho}[f(x)] = \int_{\Omega} f(x)\rho(x)\mathrm{d}x$$



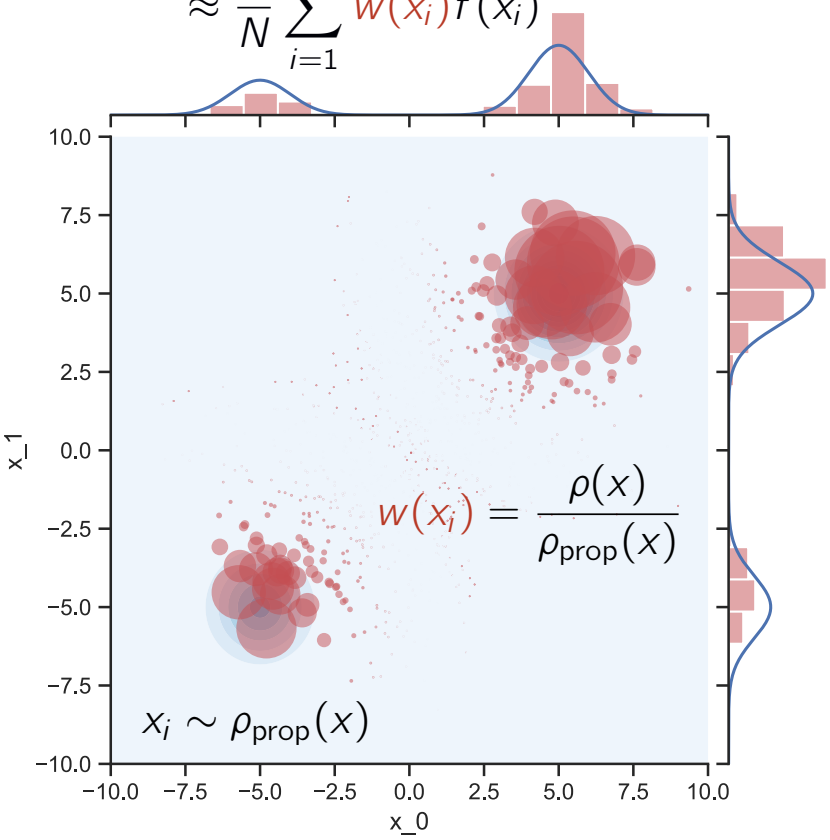
Consider simple samplers

▷ Importance sampling

rely on a tractable proposal

$$\mathbb{E}_\rho[f(x)] = \int_\Omega f(x)\rho(x)\mathrm{d}x$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i)$$



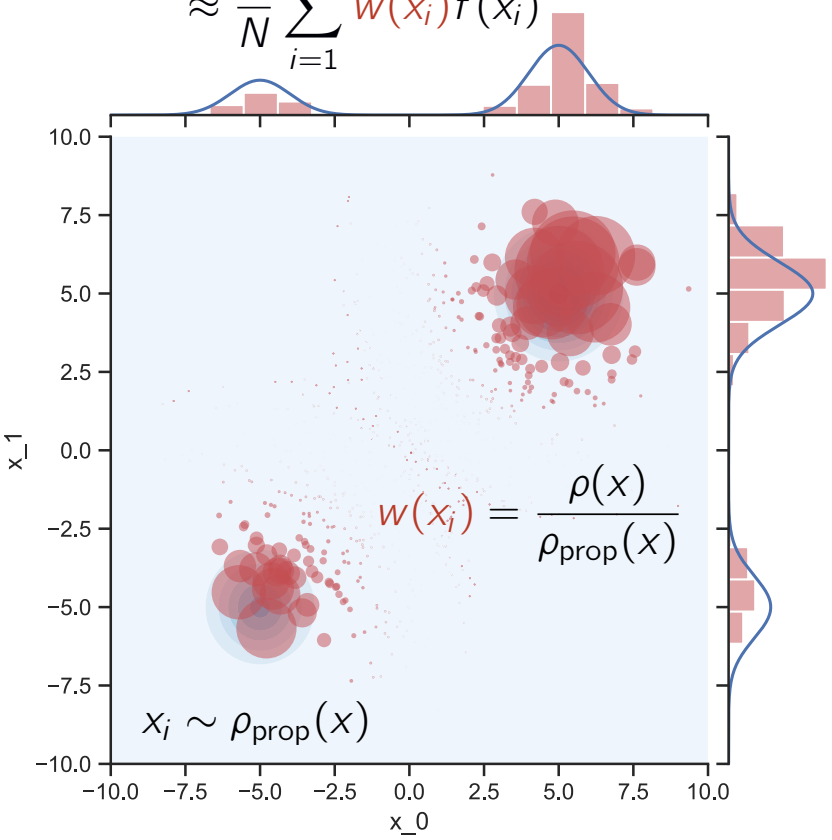
Consider simple samplers

▷ Importance sampling

rely on a tractable proposal

$$\mathbb{E}_\rho[f(x)] = \int_\Omega f(x)\rho(x)\mathrm{d}x$$

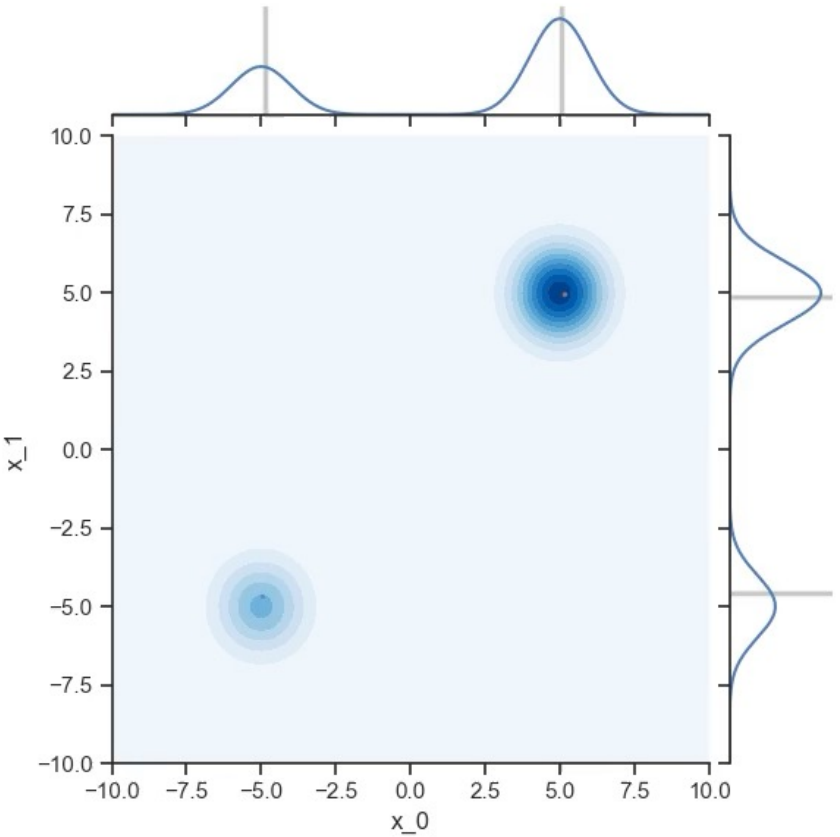
$$\approx \frac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i)$$



▷ Markov Chain Monte Carlo:
e.g. Metropolis Hastings

rely on local proposal

$$\rho_{\text{prop}}(x_{t+1}|x_t) = \mathcal{N}(x_t - dt\nabla U(x), \sqrt{2dt}I_d)$$



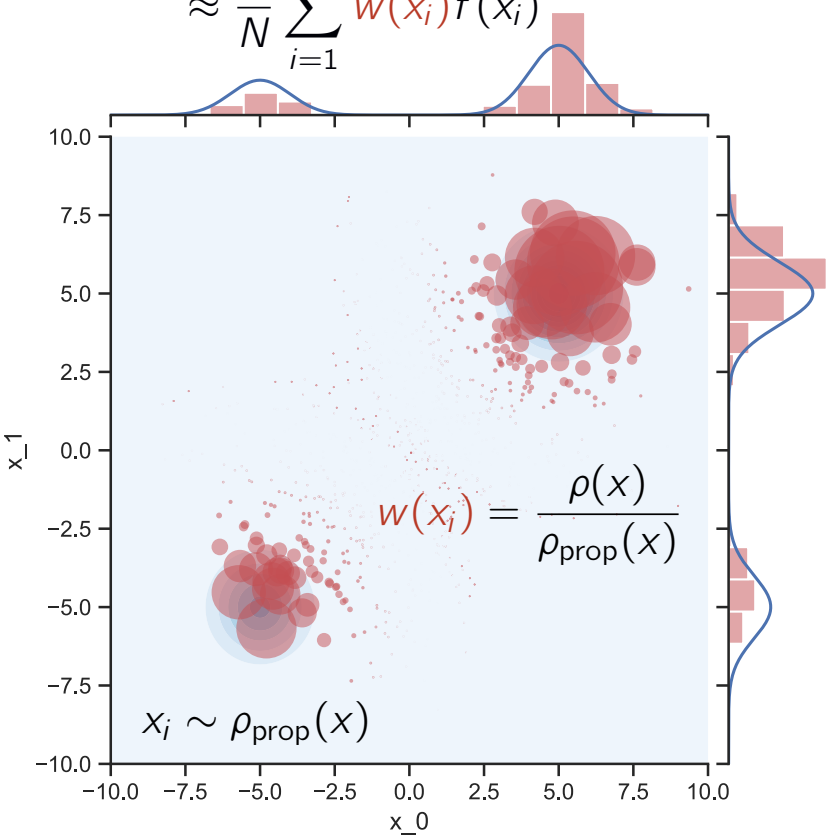
Consider simple samplers

▷ Importance sampling

rely on a tractable proposal

$$\mathbb{E}_\rho[f(x)] = \int_\Omega f(x)\rho(x)\mathrm{d}x$$

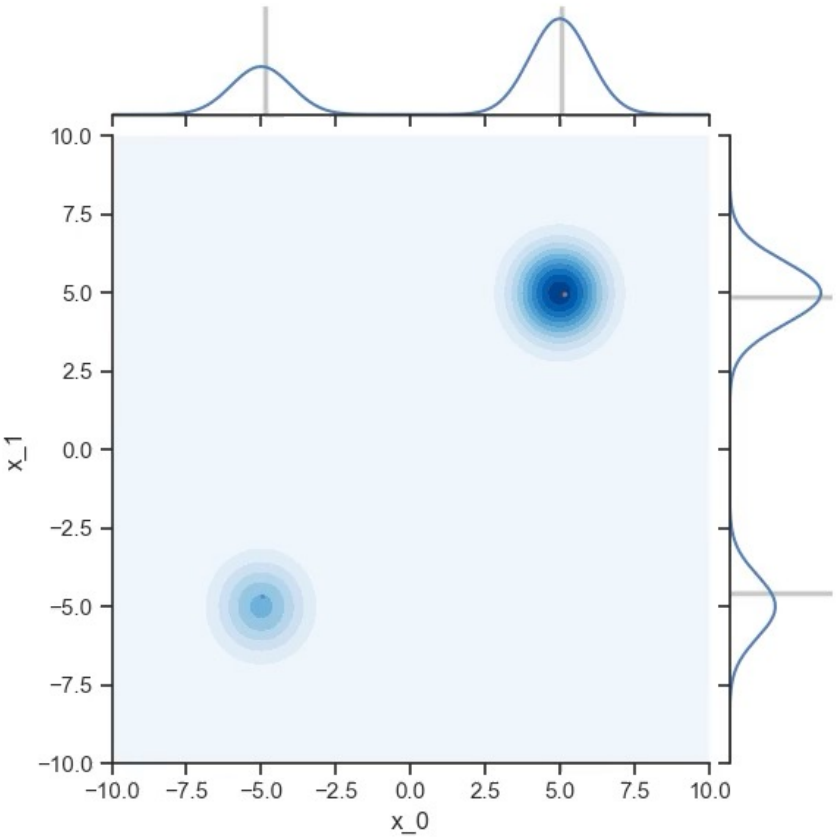
$$\approx \frac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i)$$



▷ Markov Chain Monte Carlo:
e.g. Metropolis Hastings

rely on local proposal

$$\rho_{\text{prop}}(x_{t+1}|x_t) = \mathcal{N}(x_t - dt\nabla U(x), \sqrt{2dt}I_d)$$

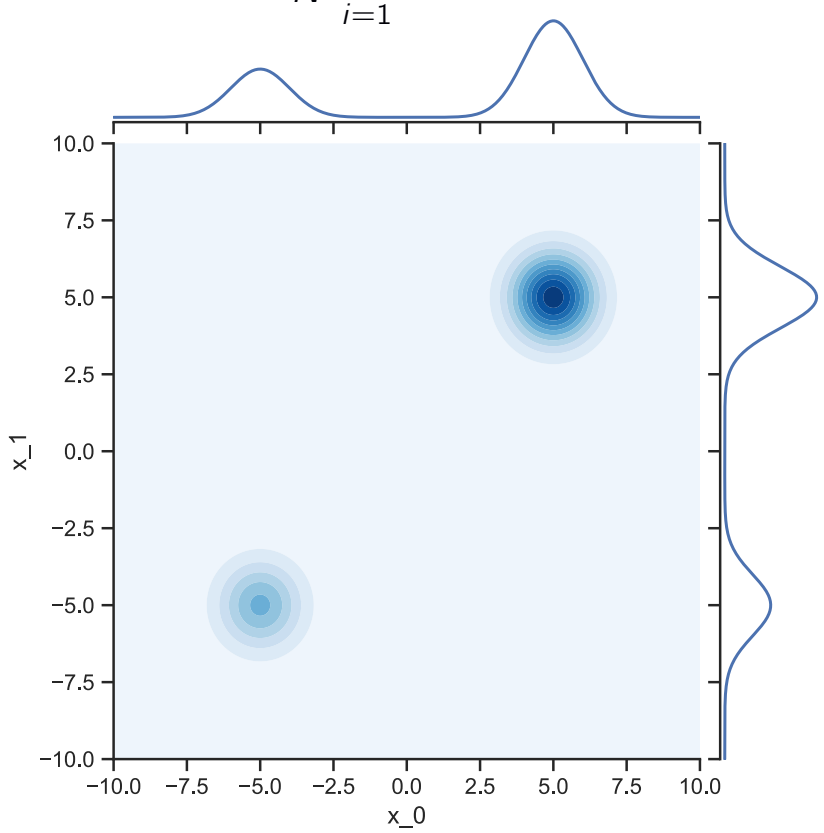


Suppose you can train a model $\rho_\theta(x) \approx \rho(x)$, what do you gain?

▷ Importance sampling

*rely on **adapated** tractable proposal!*

$$\mathbb{E}_\rho[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i)$$

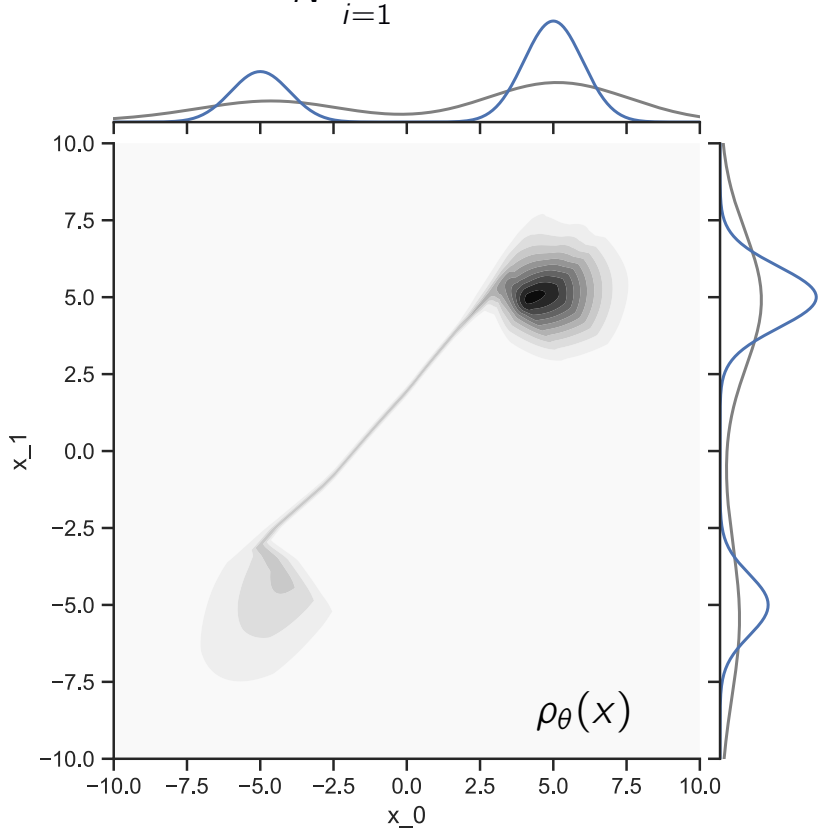


Suppose you can train a model $\rho_\theta(x) \approx \rho(x)$, what do you gain?

▷ Importance sampling

*rely on **adapated** tractable proposal!*

$$\mathbb{E}_\rho[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i)$$

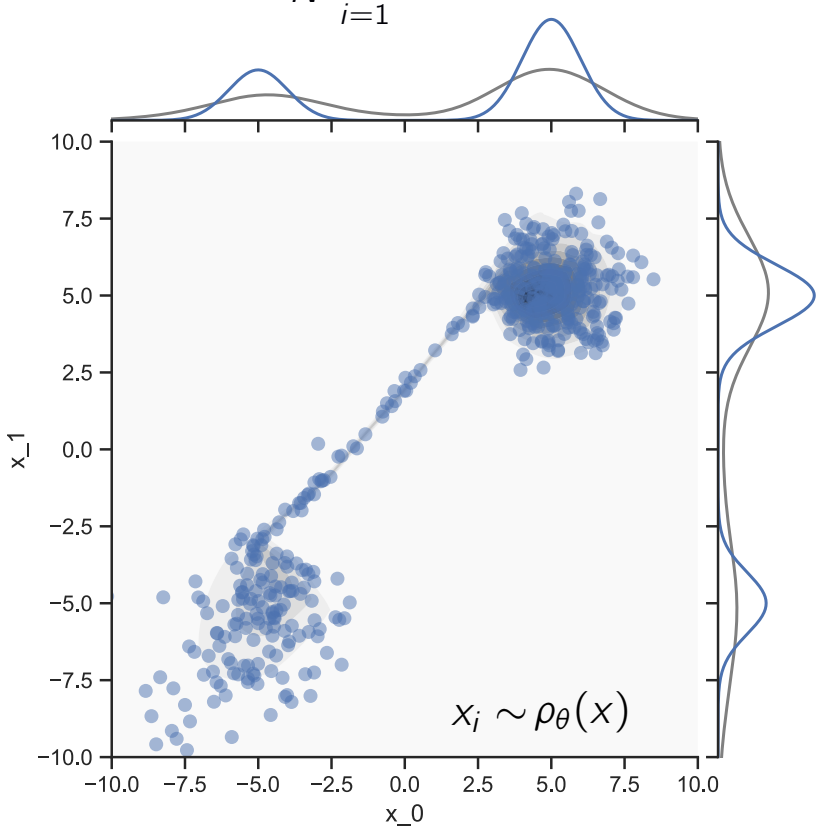


Suppose you can train a model $\rho_\theta(x) \approx \rho(x)$, what do you gain?

▷ Importance sampling

*rely on **adapated** tractable proposal!*

$$\mathbb{E}_\rho[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i)$$

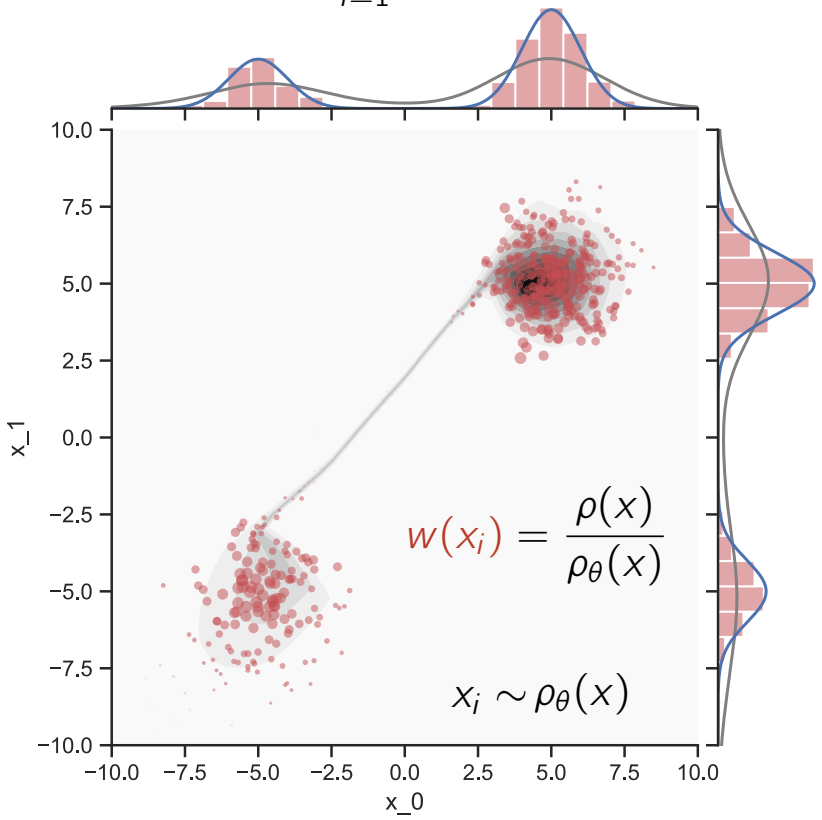


Suppose you can train a model $\rho_\theta(x) \approx \rho(x)$, what do you gain?

▷ Importance sampling

*rely on **adapated** tractable proposal!*

$$\mathbb{E}_\rho[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i)$$

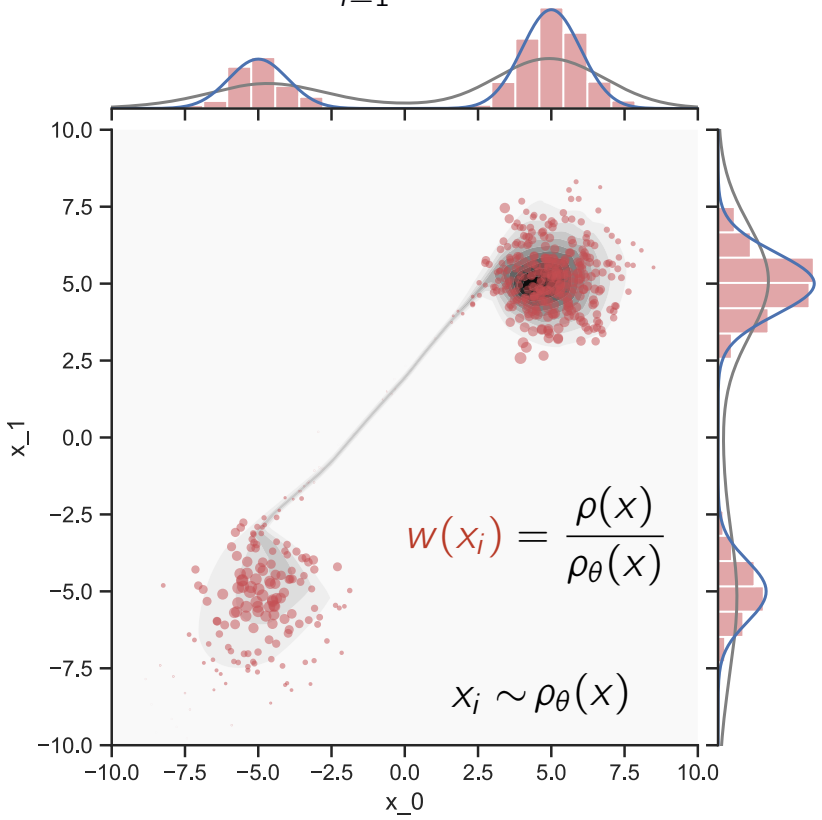


Suppose you can train a model $\rho_\theta(x) \approx \rho(x)$, what do you gain?

▷ Importance sampling

*rely on **adapated** tractable proposal!*

$$\mathbb{E}_\rho[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i)$$



▷ Markov Chain Monte Carlo: e.g. Metropolis Hastings

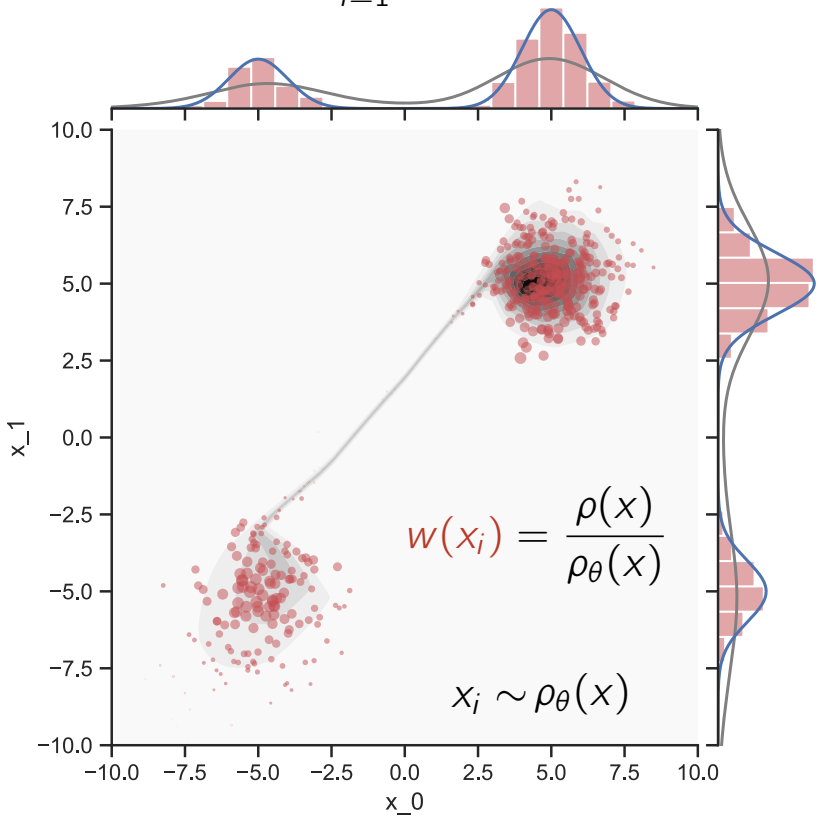
*rely on **global** proposal!*

Suppose you can train a model $\rho_\theta(x) \approx \rho(x)$, what do you gain?

▷ Importance sampling

*rely on **adapated** tractable proposal!*

$$\mathbb{E}_\rho[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i)$$



▷ Markov Chain Monte Carlo: e.g. Metropolis Hastings

*rely on **global** proposal!*

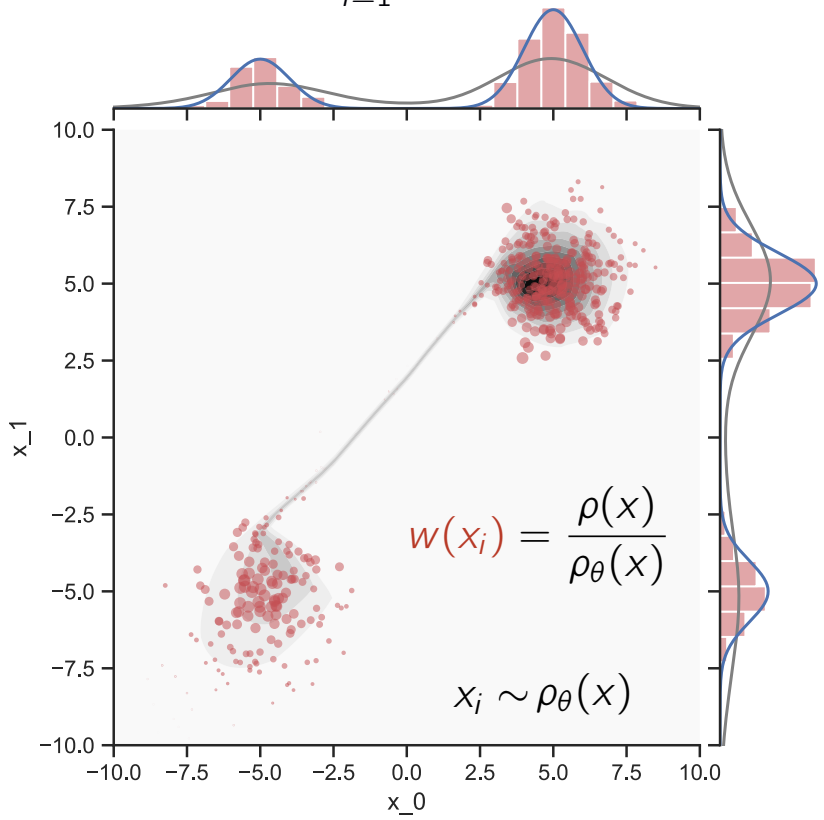
$$\rho_{\text{prop}}(x_{t+1}|x_t) = \rho_\theta(x_{t+1})$$

Suppose you can train a model $\rho_\theta(x) \approx \rho(x)$, what do you gain?

▷ Importance sampling

*rely on **adapated** tractable proposal!*

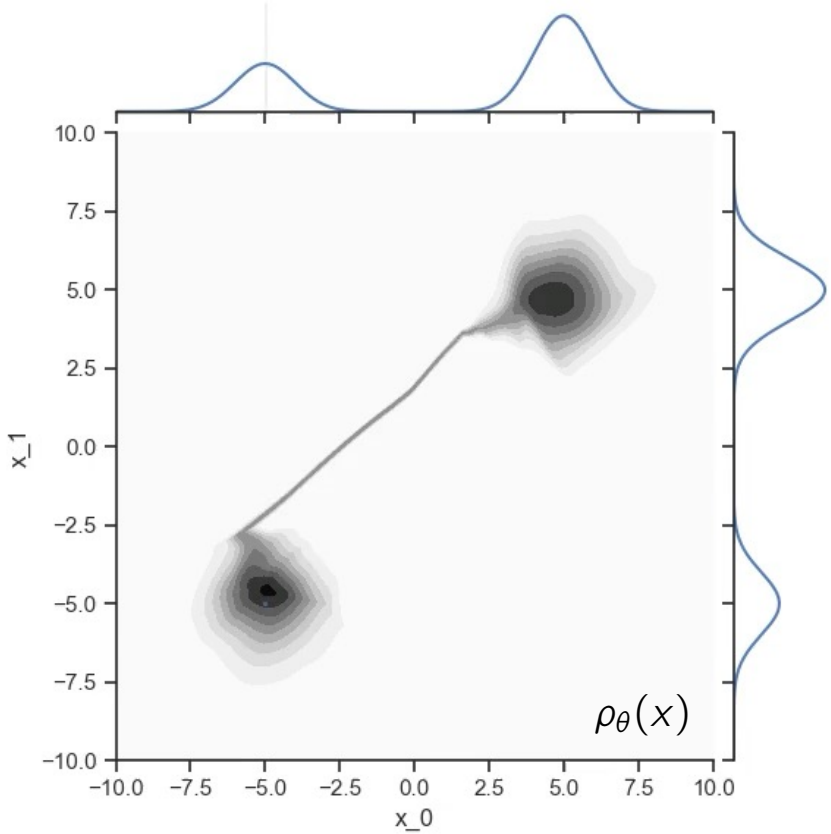
$$\mathbb{E}_\rho[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i)$$



▷ Markov Chain Monte Carlo:
e.g. Metropolis Hastings

*rely on **global** proposal!*

$$\rho_{\text{prop}}(x_{t+1}|x_t) = \rho_\theta(x_{t+1})$$

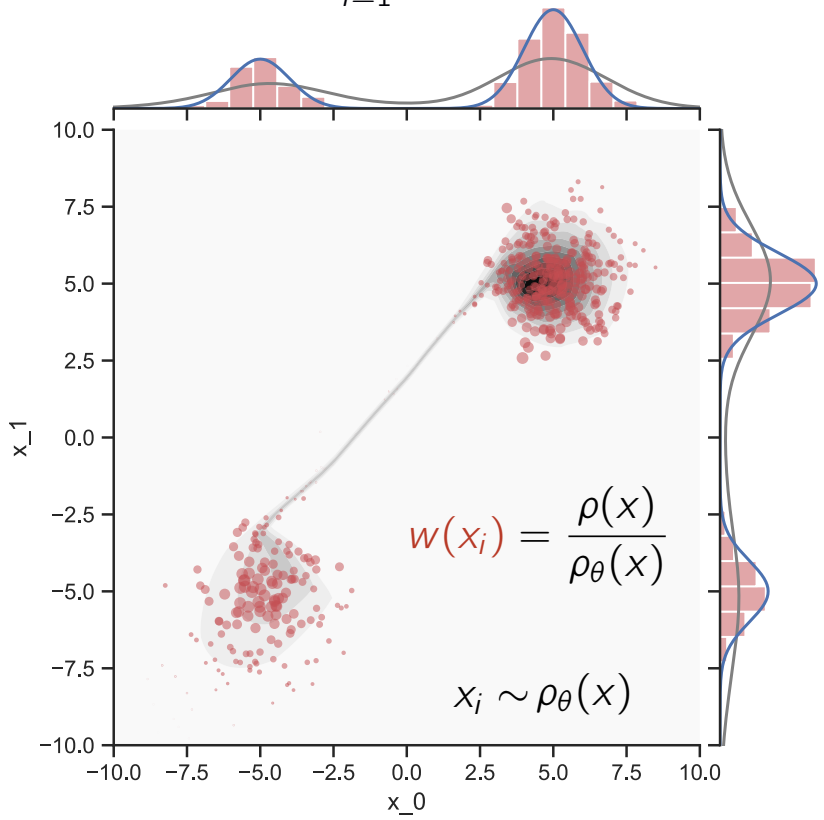


Suppose you can train a model $\rho_\theta(x) \approx \rho(x)$, what do you gain?

▷ Importance sampling

*rely on **adapated** tractable proposal!*

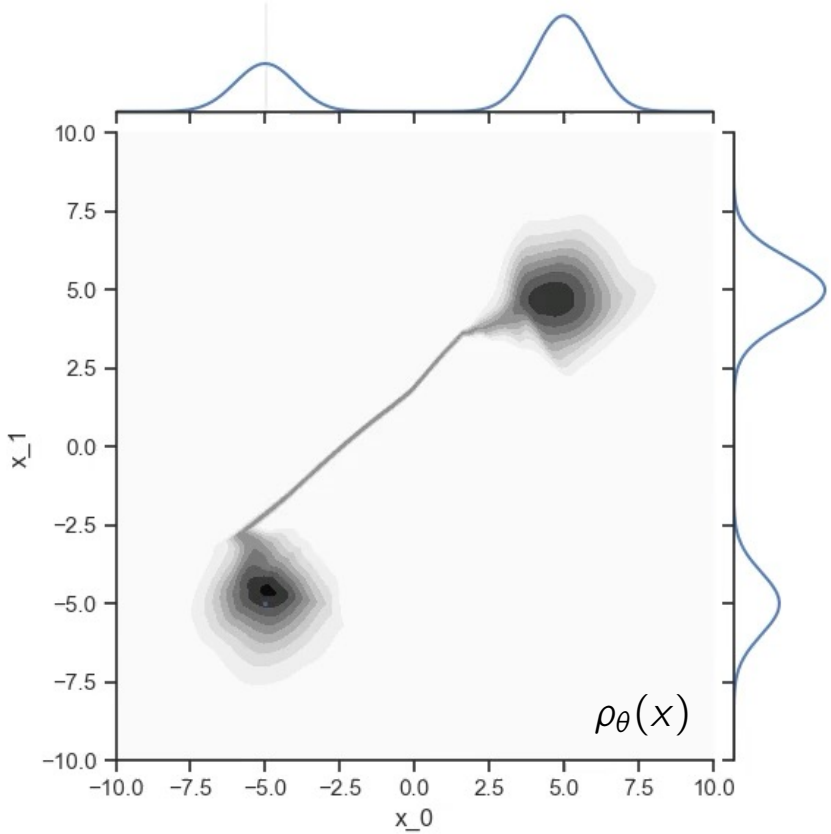
$$\mathbb{E}_\rho[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i)$$



▷ Markov Chain Monte Carlo:
e.g. Metropolis Hastings

*rely on **global** proposal!*

$$\rho_{\text{prop}}(x_{t+1}|x_t) = \rho_\theta(x_{t+1})$$



Suppose you can train a model $\rho_\theta(x) \approx \rho_*(x)$, what do you gain?

Suppose you can train a model $\rho_\theta(x) \approx \rho_*(x)$, what do you gain?

$\triangleright$ A lot!

Suppose you can train a model $\rho_\theta(x) \approx \rho_*(x)$, what do you gain?

▷A lot!

Suppose you can train a model $\rho_\theta(x) \approx \rho_*(x)$, what do you gain?

▷ A lot!

First idea: variational inference “on steroids”:

▷ A data-free learning objective: the (reverse) Kullback-Leibler divergence $D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$

$$D_{\mathrm{KL}}(\rho_\theta \| \rho_*) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_\theta(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \frac{\rho_B(z_i) \det |\nabla_{z_i} T_\theta|}{\rho_*(T_\theta(z_i))} \qquad z_i \sim \rho_B(z)$$

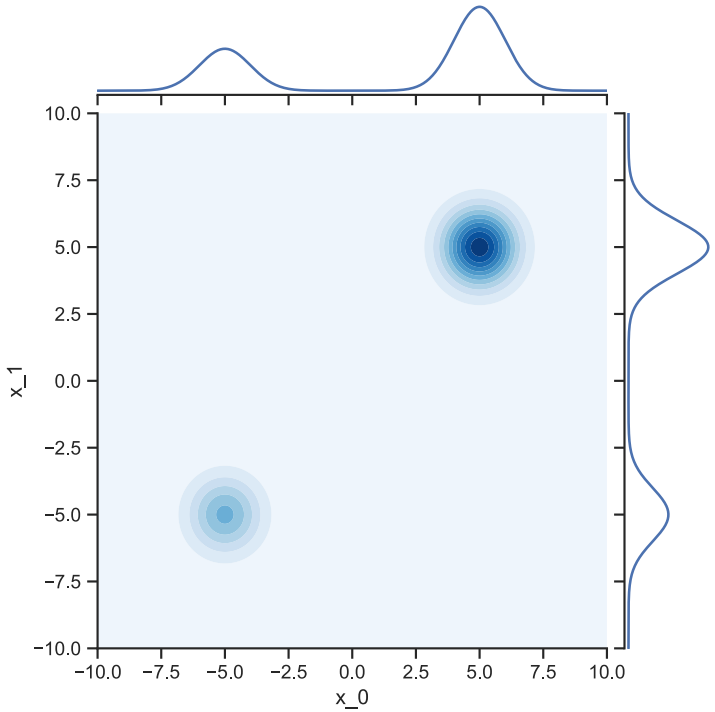
Weiss, P. (1907). L’hypothèse du champ moléculaire et la propriété ferromagnétique.
Wainwright, M. J., & Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference.
Rezende & Mohamed, (2015). Variational inference with normalizing flows
Wu et al. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks.
Albergo et al (2019). Flow-based generative models for Markov chain Monte Carlo in lattice field theory.
Li, Shuo-Hui, and Lei Wang. “Neural Network Renormalization Group.” *PRL 2018*

First idea: variational inference “on steroids”:

▷ A data-free learning objective: the (reverse) Kullback-Leibler divergence $D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$

$$D_{\mathrm{KL}}(\rho_\theta \| \rho_*) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_\theta(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \frac{\rho_B(z_i) \det |\nabla_{z_i} T_\theta|}{\rho_*(T_\theta(z_i))} \qquad z_i \sim \rho_B(z)$$

$\rho_*(x)$



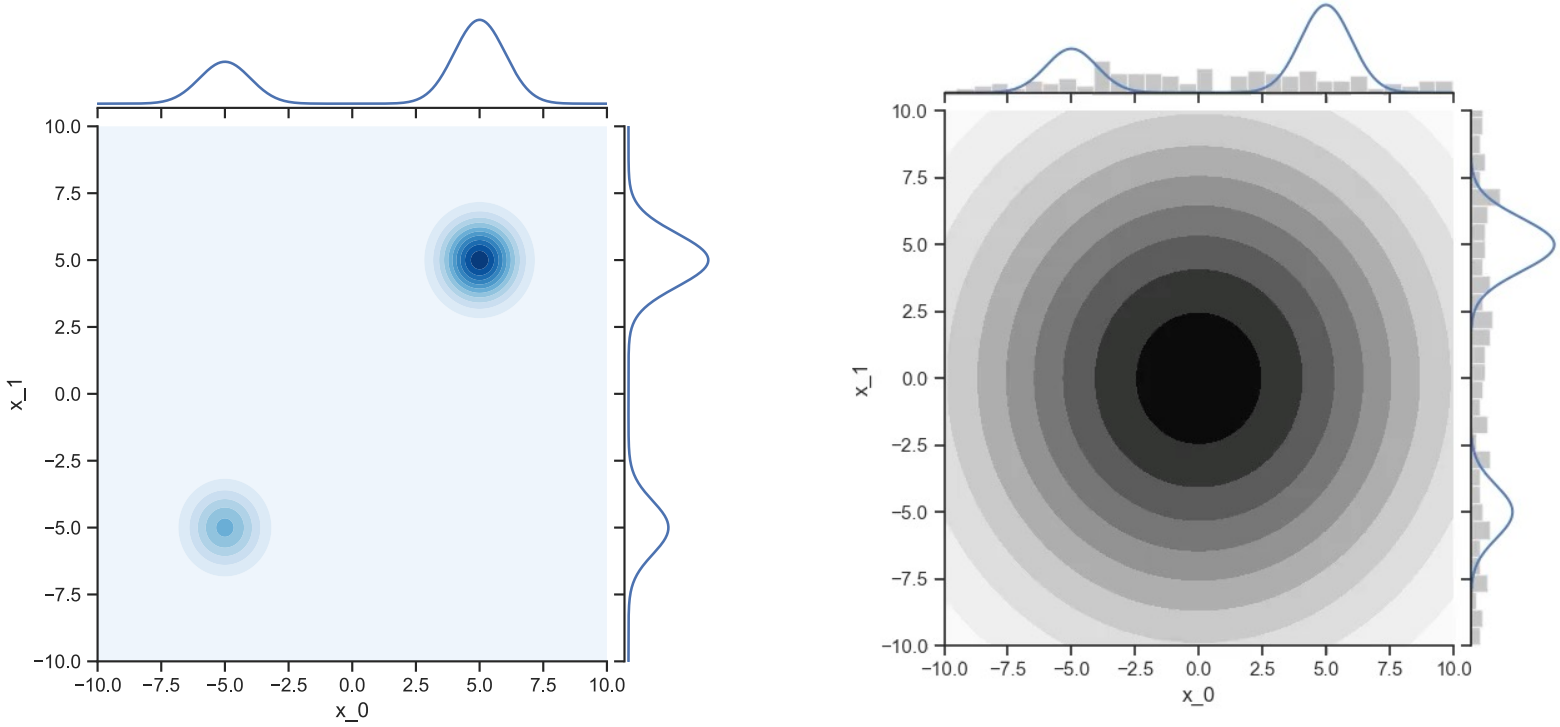
Weiss, P. (1907). L’hypothèse du champ moléculaire et la propriété ferromagnétique.
Wainwright, M. J., & Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference.
Rezende & Mohamed, (2015). Variational inference with normalizing flows
Wu et al. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks.
Albergo et al (2019). Flow-based generative models for Markov chain Monte Carlo in lattice field theory.
Li, Shuo-Hui, and Lei Wang. “Neural Network Renormalization Group.” *PRL 2018*

First idea: variational inference “on steroids”:

▷ A data-free learning objective: the (reverse) Kullback-Leibler divergence $D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$

$$D_{\mathrm{KL}}(\rho_\theta \| \rho_*) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_\theta(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \frac{\rho_B(z_i) \det |\nabla_{z_i} T_\theta|}{\rho_*(T_\theta(z_i))} \qquad z_i \sim \rho_B(z)$$

$\rho_*(x)$ $\qquad$ $\rho_\theta(x)$



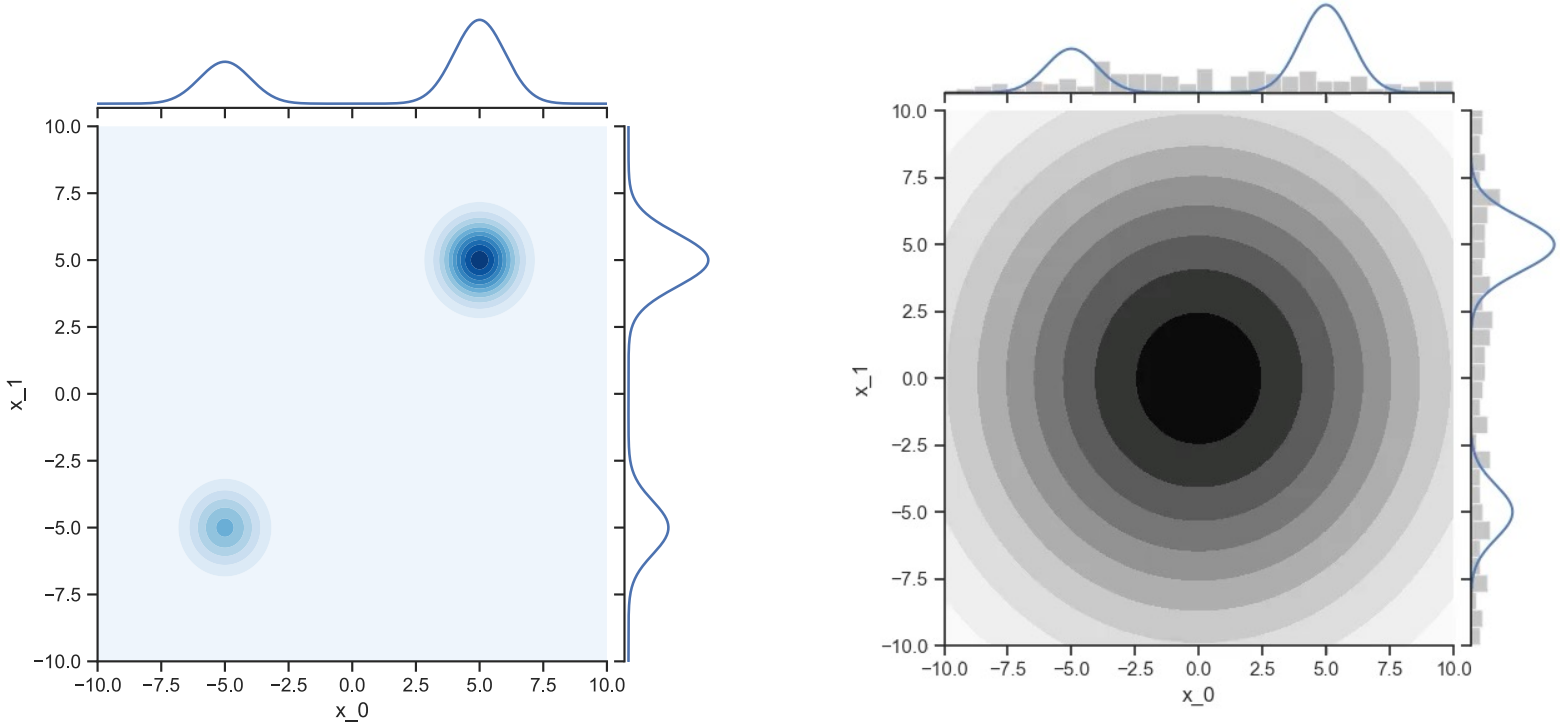
Weiss, P. (1907). L’hypothèse du champ moléculaire et la propriété ferromagnétique.
Wainwright, M. J., & Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference.
Rezende & Mohamed, (2015). Variational inference with normalizing flows
Wu et al. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks.
Albergo et al (2019). Flow-based generative models for Markov chain Monte Carlo in lattice field theory.
Li, Shuo-Hui, and Lei Wang. “Neural Network Renormalization Group.” *PRL 2018*

First idea: variational inference “on steroids”:

▷ A data-free learning objective: the (reverse) Kullback-Leibler divergence $D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$

$$D_{\mathrm{KL}}(\rho_\theta \| \rho_*) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_\theta(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \frac{\rho_B(z_i) \det |\nabla_{z_i} T_\theta|}{\rho_*(T_\theta(z_i))} \qquad z_i \sim \rho_B(z)$$

$\rho_*(x)$ $\qquad$ $\rho_\theta(x)$



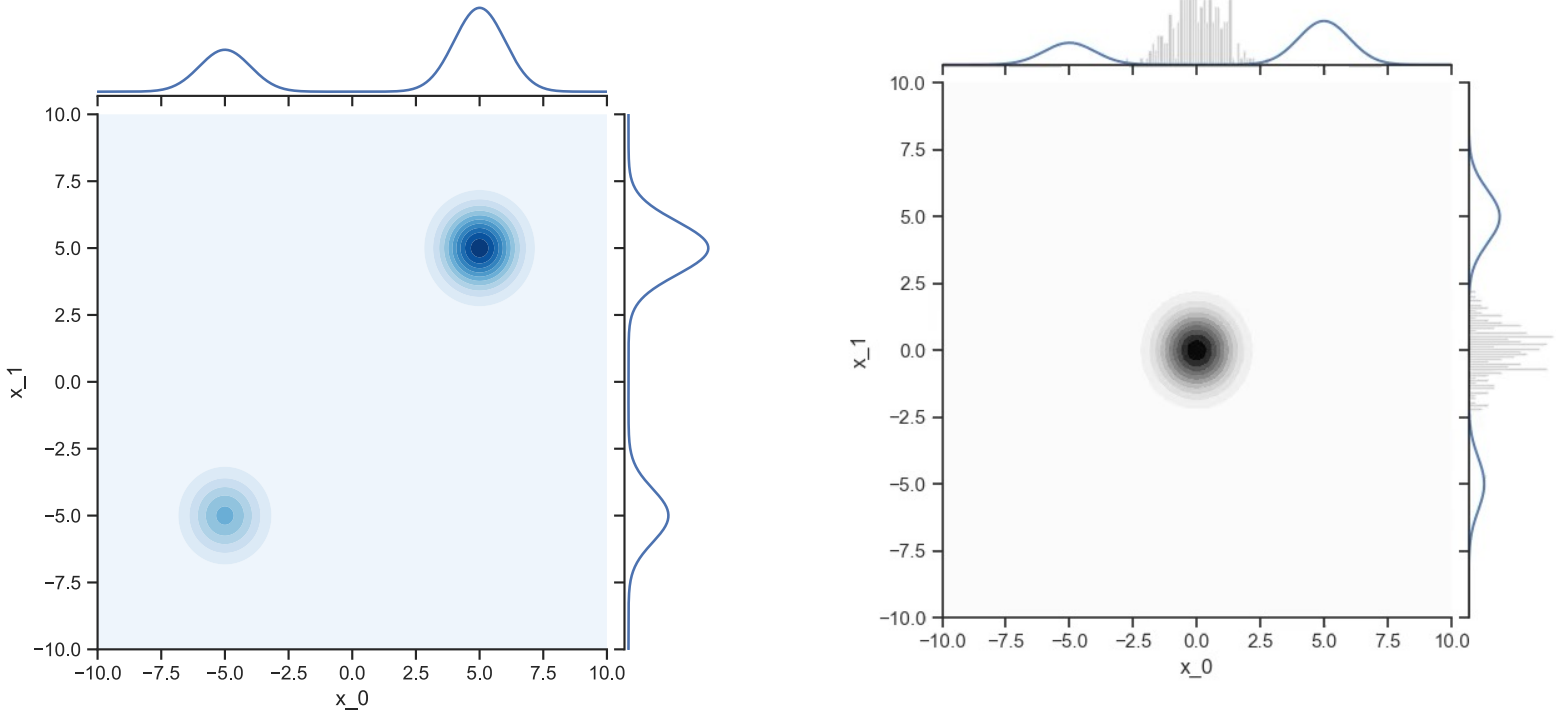
Weiss, P. (1907). L’hypothèse du champ moléculaire et la propriété ferromagnétique.
Wainwright, M. J., & Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference.
Rezende & Mohamed, (2015). Variational inference with normalizing flows
Wu et al. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks.
Albergo et al (2019). Flow-based generative models for Markov chain Monte Carlo in lattice field theory.
Li, Shuo-Hui, and Lei Wang. “Neural Network Renormalization Group.” *PRL 2018*

First idea: variational inference “on steroids”:

▷ A data-free learning objective: the (reverse) Kullback-Leibler divergence $D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$

$$D_{\mathrm{KL}}(\rho_\theta \| \rho_*) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_\theta(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \frac{\rho_B(z_i) \det |\nabla_{z_i} T_\theta|}{\rho_*(T_\theta(z_i))} \qquad z_i \sim \rho_B(z)$$

$\rho_*(x)$ $\qquad$ $\rho_\theta(x)$



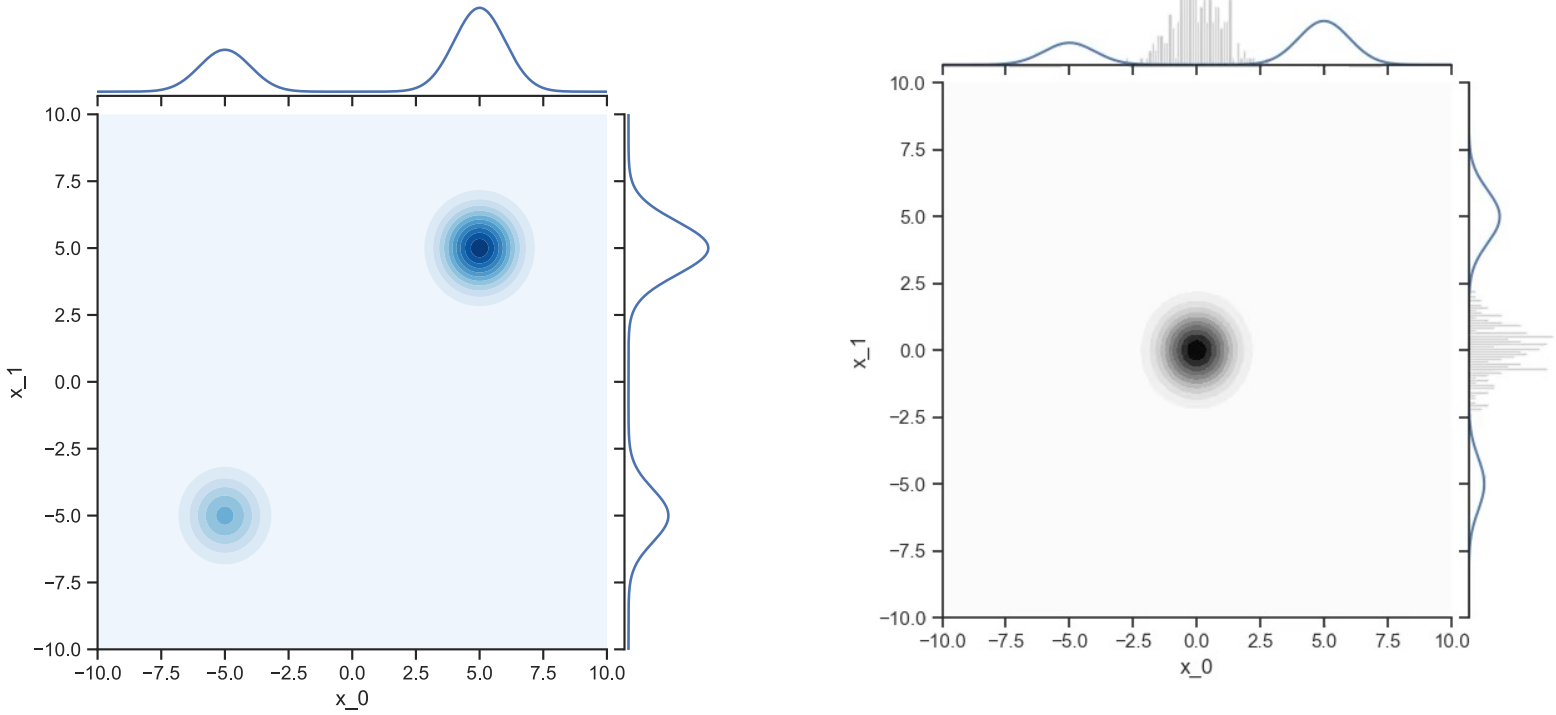
Weiss, P. (1907). L’hypothèse du champ moléculaire et la propriété ferromagnétique.
Wainwright, M. J., & Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference.
Rezende & Mohamed, (2015). Variational inference with normalizing flows
Wu et al. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks.
Albergo et al (2019). Flow-based generative models for Markov chain Monte Carlo in lattice field theory.
Li, Shuo-Hui, and Lei Wang. “Neural Network Renormalization Group.” *PRL 2018*

First idea: variational inference “on steroids”:

▷ A data-free learning objective: the (reverse) Kullback-Leibler divergence $D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$

$$D_{\mathrm{KL}}(\rho_\theta \| \rho_*) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_\theta(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \frac{\rho_B(z_i) \det |\nabla_{z_i} T_\theta|}{\rho_*(T_\theta(z_i))} \qquad z_i \sim \rho_B(z)$$

$\rho_*(x)$ $\qquad$ $\rho_\theta(x)$



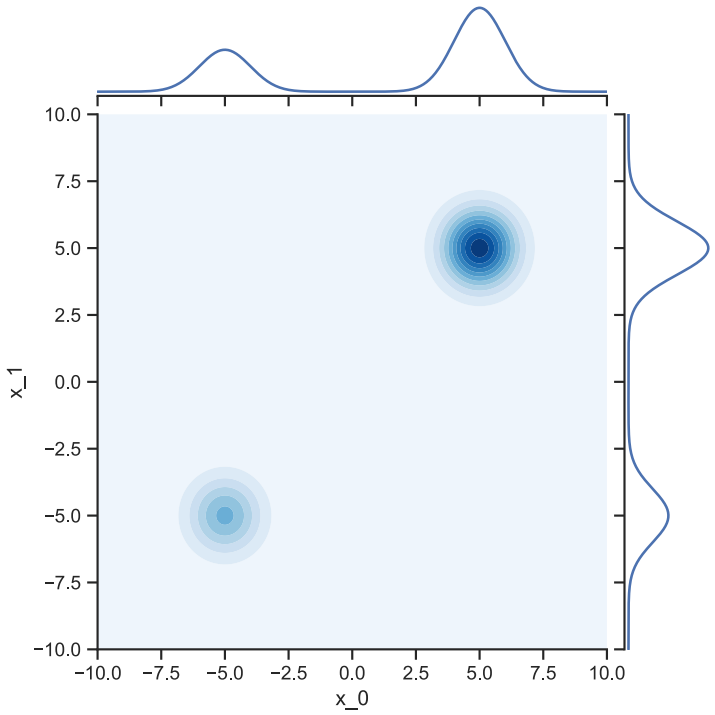
Weiss, P. (1907). L’hypothèse du champ moléculaire et la propriété ferromagnétique.
Wainwright, M. J., & Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference.
Rezende & Mohamed, (2015). Variational inference with normalizing flows
Wu et al. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks.
Albergo et al (2019). Flow-based generative models for Markov chain Monte Carlo in lattice field theory.
Li, Shuo-Hui, and Lei Wang. “Neural Network Renormalization Group.” *PRL 2018*

First idea: variational inference “on steroids”: Caution: prone to mode-collapse

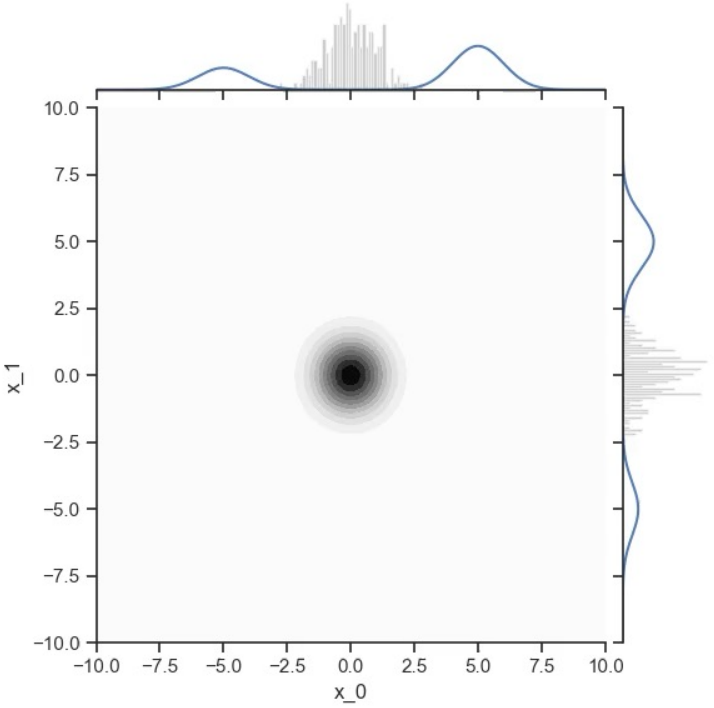
▷ A data-free learning objective: the (reverse) Kullback-Leibler divergence $D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$

$$D_{\mathrm{KL}}(\rho_\theta \| \rho_*) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_\theta(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \frac{\rho_B(z_i) \det |\nabla_{z_i} T_\theta|}{\rho_*(T_\theta(z_i))} \qquad z_i \sim \rho_B(z)$$

$\rho_*(x)$



$\rho_\theta(x)$



Weiss, P. (1907). L’hypothèse du champ moléculaire et la propriété ferromagnétique.
Wainwright, M. J., & Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference.
Rezende & Mohamed, (2015). Variational inference with normalizing flows
Wu et al. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks.
Albergo et al (2019). Flow-based generative models for Markov chain Monte Carlo in lattice field theory.
Li, Shuo-Hui, and Lei Wang. “Neural Network Renormalization Group.” *PRL 2018*

Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

▷ Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_* \| \rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_*(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

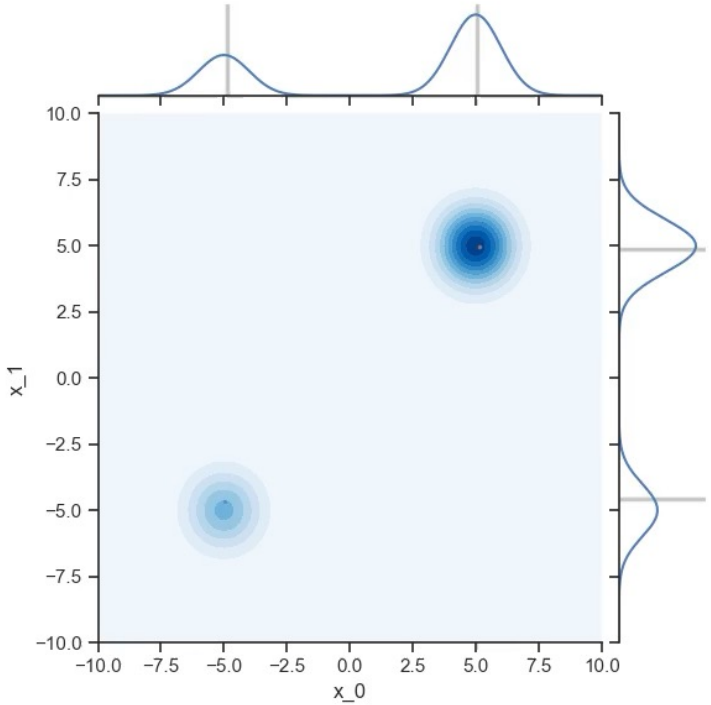
Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

- ▷ Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_*\|\rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_*(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

- ▷ Use an auxillary simple sampler to create data $x^i_{t+1} \sim \pi_{\mathrm{local}}(x^i_{t+1}|x^i_t)$



[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

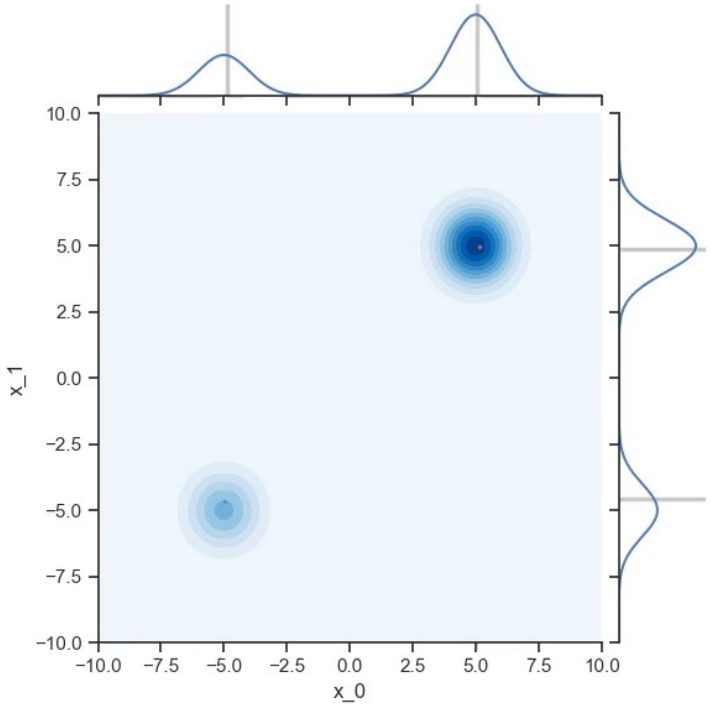
Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

- ▷ Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_*\|\rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_*(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

- ▷ Use an auxillary simple sampler to create data $x^i_{t+1} \sim \pi_{\mathrm{local}}(x^i_{t+1}|x^i_t)$



[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

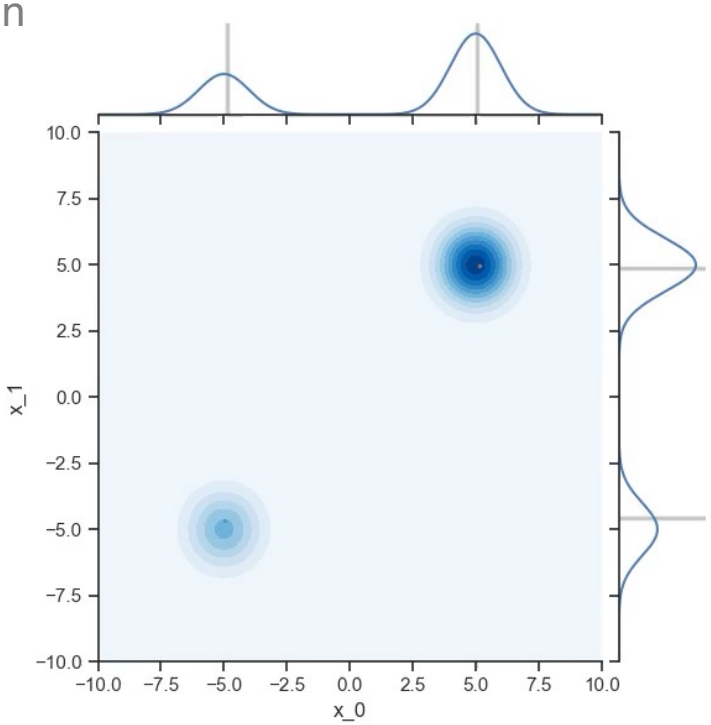
Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

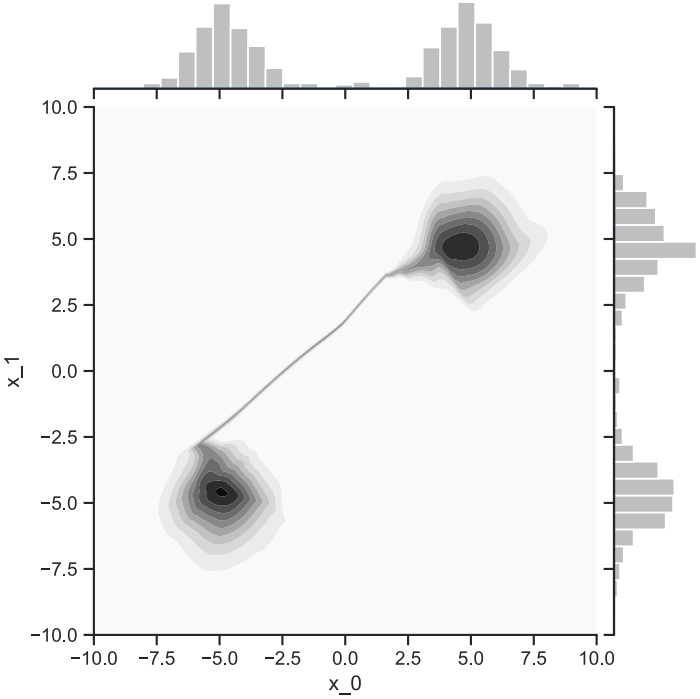
With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

- ▷ Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_* \| \rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_*(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

- ▷ Use an auxillary simple sampler to create data $x^i_{t+1} \sim \pi_{\mathrm{local}}(x^i_{t+1} | x^i_t)$
- ▷ Train





[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

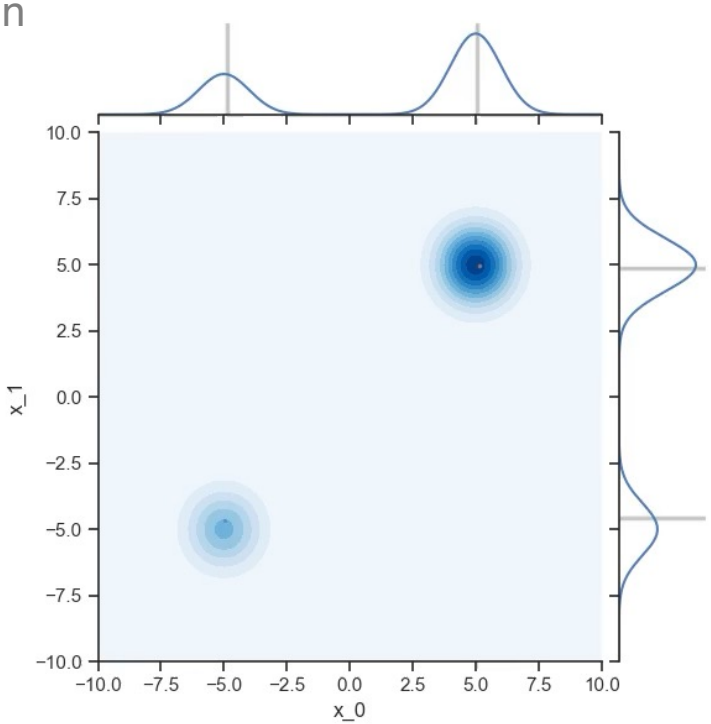
Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

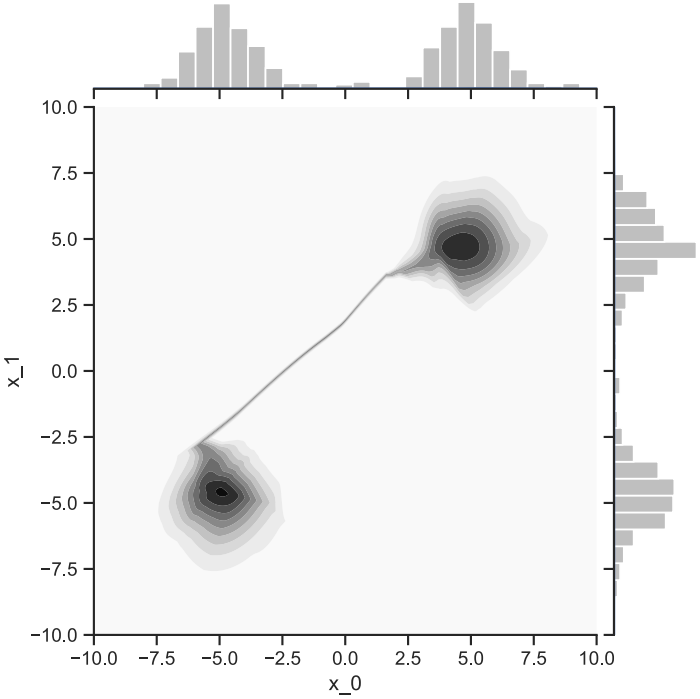
With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

- ▷ Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_*\|\rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)}\rho_*(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

- ▷ Use an auxillary simple sampler to create data $x^i_{t+1} \sim \pi_{\mathrm{local}}(x^i_{t+1}|x^i_t)$
- ▷ Train





- ▷ Add non-local steps relying on flow proposals $x \sim \rho_\theta(x_i)$

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

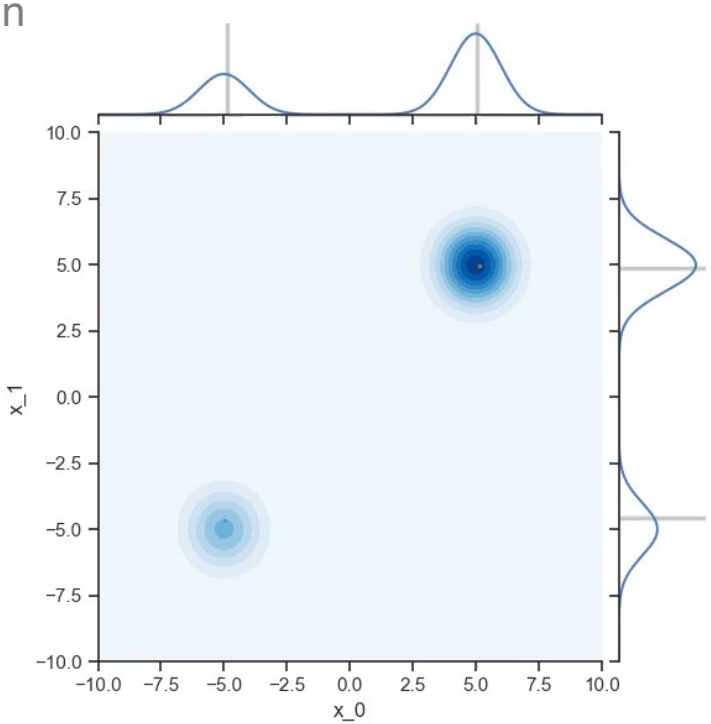
Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

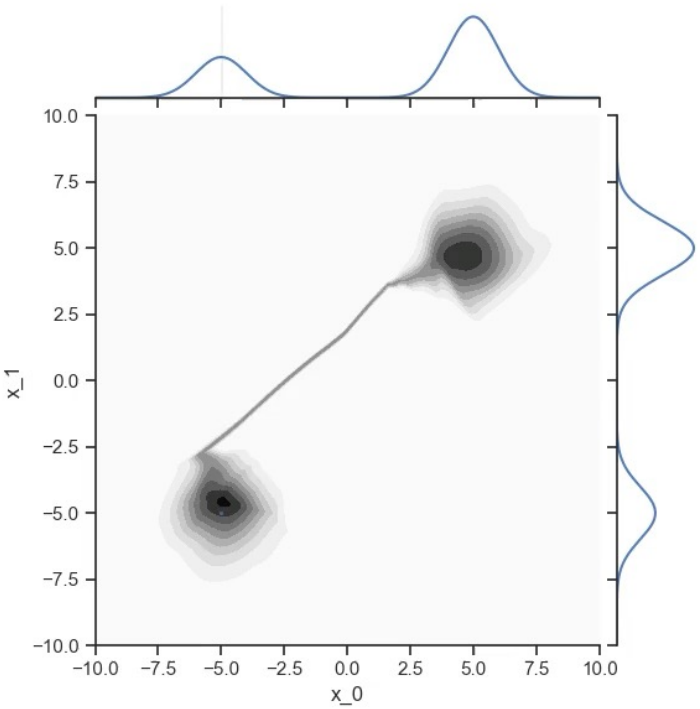
With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

- Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_* \| \rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_*(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

- Use an auxillary simple sampler to create data $x^i_{t+1} \sim \pi_{\mathrm{local}}(x^i_{t+1} | x^i_t)$
- Train





- Add non-local steps relying on flow proposals $x \sim \rho_\theta(x_i)$

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

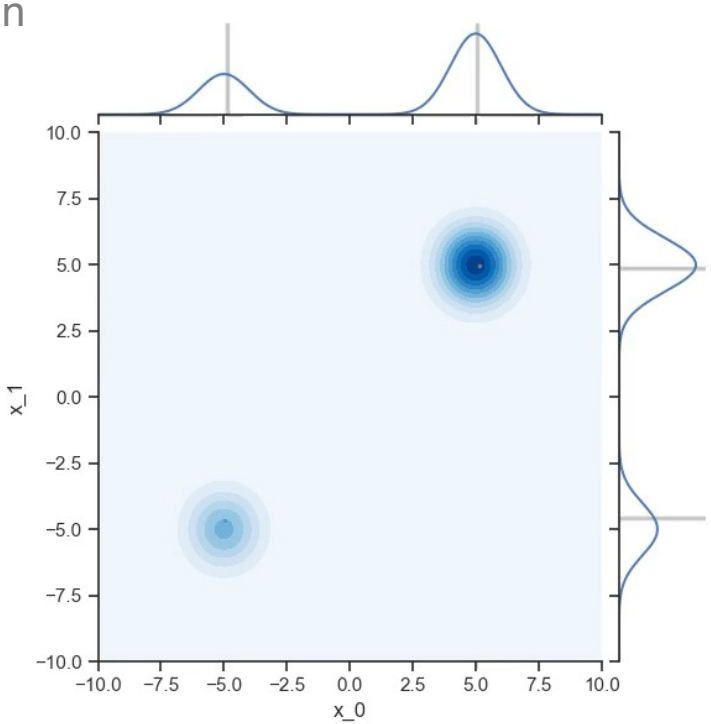
Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

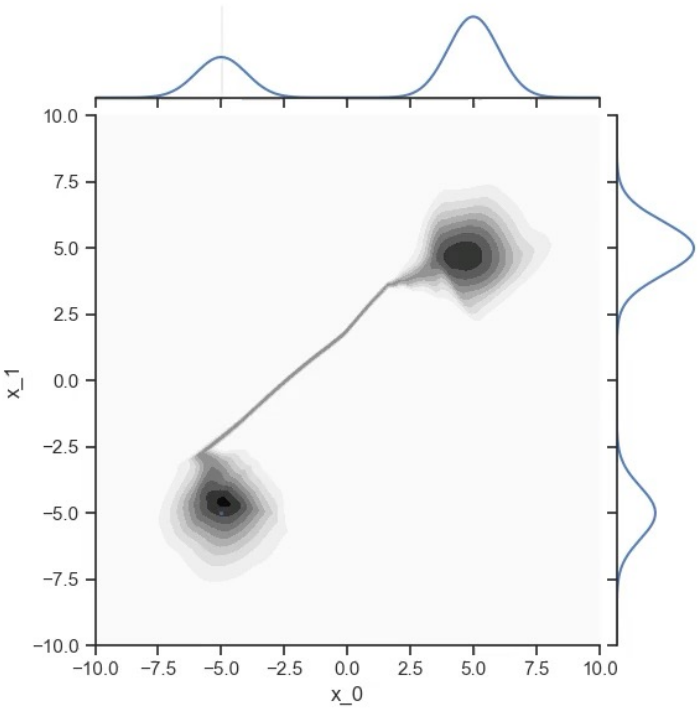
With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

- Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_* \| \rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_*(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

- Use an auxillary simple sampler to create data $x^i_{t+1} \sim \pi_{\mathrm{local}}(x^i_{t+1} | x^i_t)$
- Train





- Add non-local steps relying on flow proposals $x \sim \rho_\theta(x_i)$

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

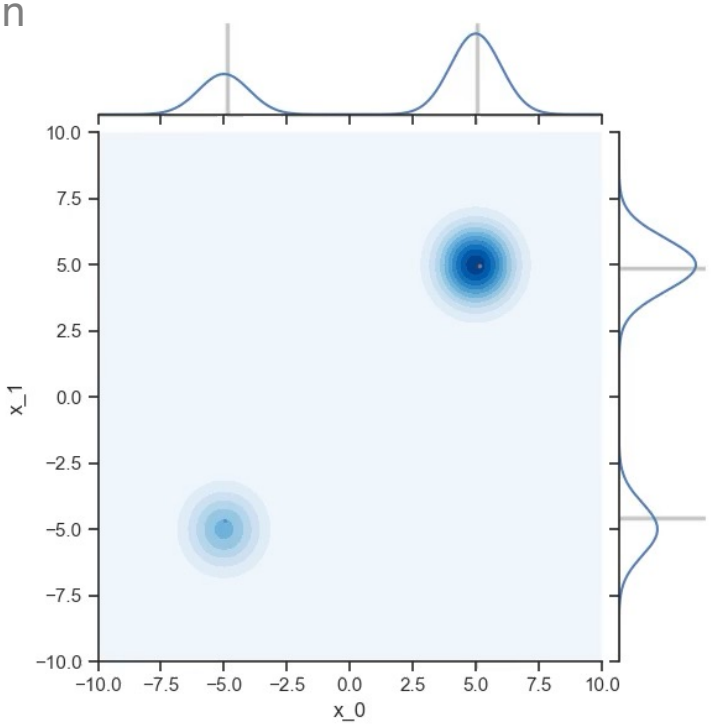
Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

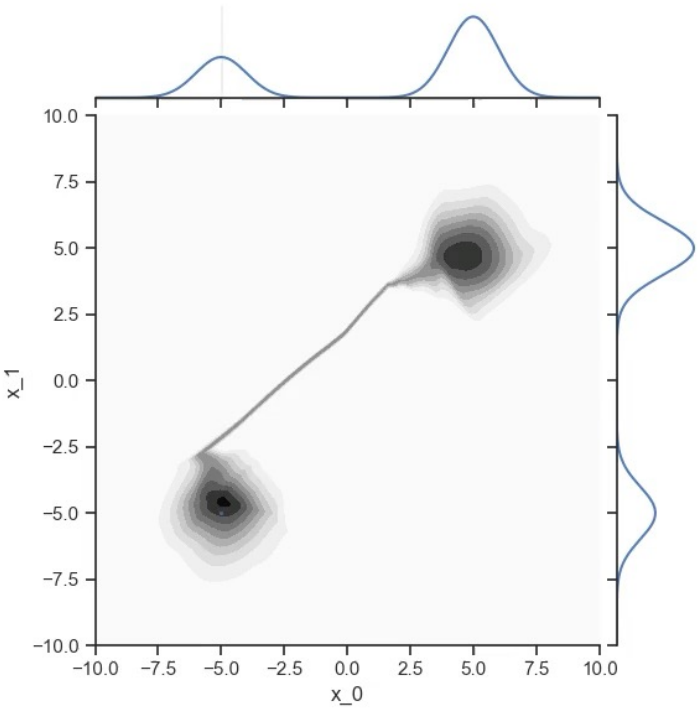
With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

- ▷ Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_*\|\rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)}\rho_*(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

- ▷ Use an auxillary simple sampler to create data $x^i_{t+1} \sim \pi_{\mathrm{local}}(x^i_{t+1}|x^i_t)$
- ▷ Train





- ▷ Add non-local steps relying on flow proposals $x \sim \rho_\theta(x_i)$
- ▷ Repeat and converge

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

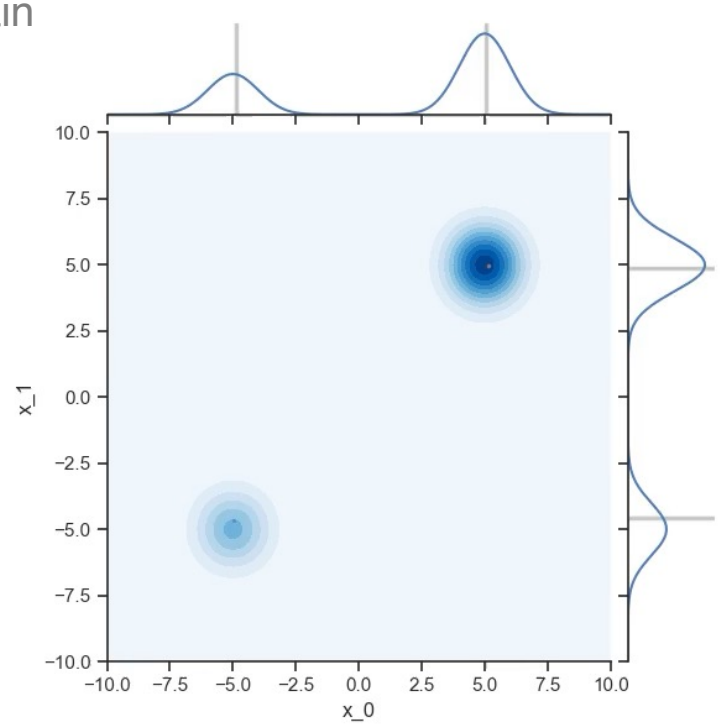
Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

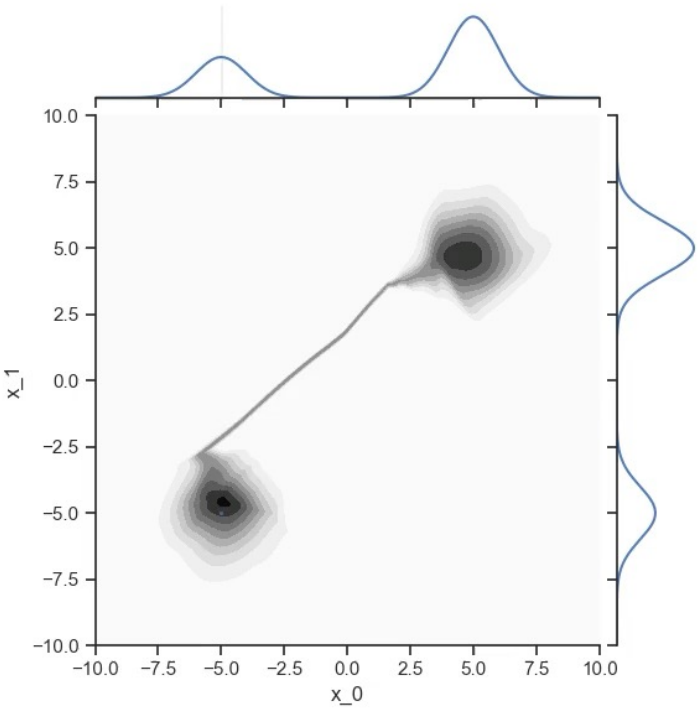
With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

- ▷ Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_* \| \rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_*(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

- ▷ Use an auxillary simple sampler to create data $x^i_{t+1} \sim \pi_{\mathrm{local}}(x^i_{t+1} | x^i_t)$
- ▷ Train





- ▷ Add non-local steps relying on flow proposals $x \sim \rho_\theta(x_i)$ *Global-Local algorithm!*
- ▷ Repeat and converge

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

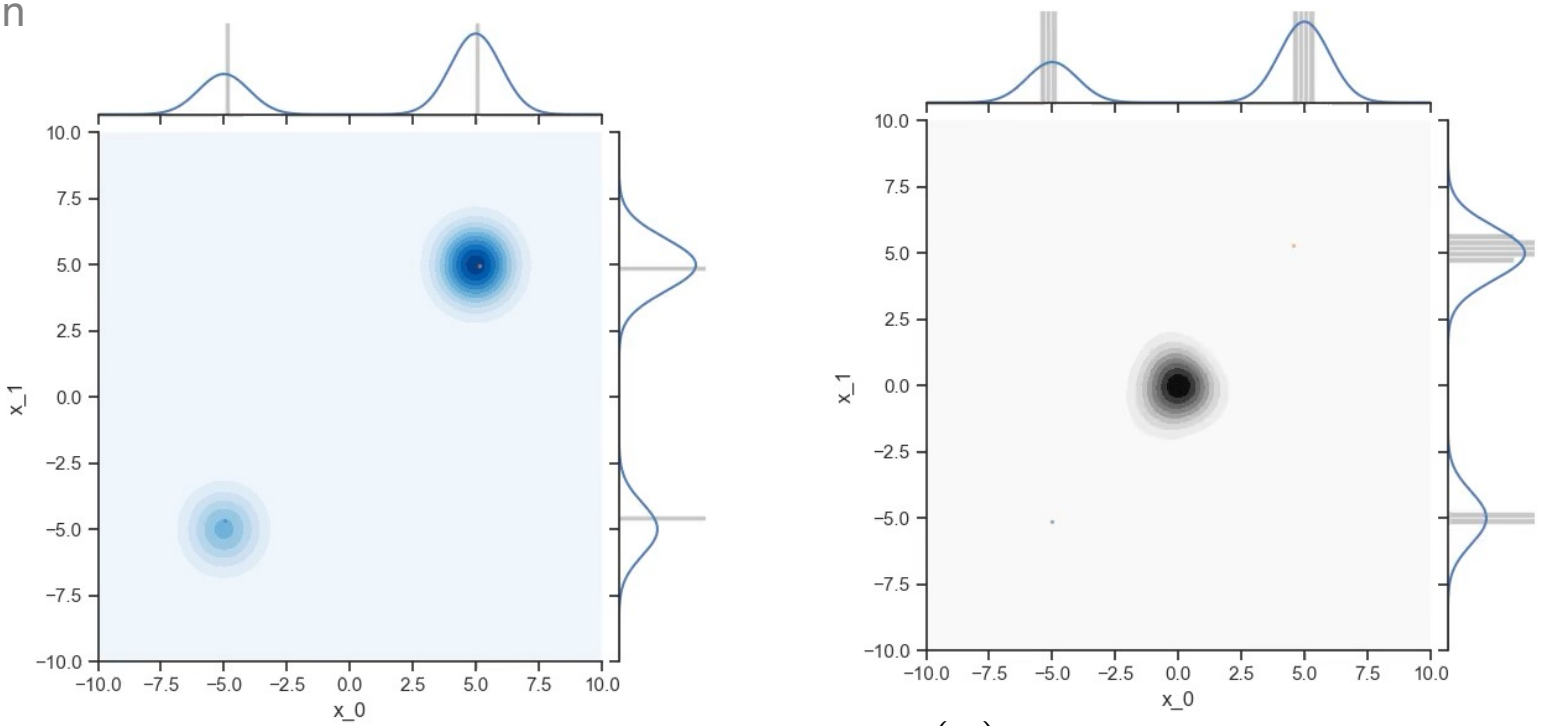
Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

- ▷ Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_* \| \rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_*(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

- ▷ Use an auxillary simple sampler to create data $x^i_{t+1} \sim \pi_{\mathrm{local}}(x^i_{t+1} | x^i_t)$
- ▷ Train



- ▷ Add non-local steps relying on flow proposals $x \sim \rho_\theta(x_i)$

Global-Local algorithm!

- ▷ Repeat and converge

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

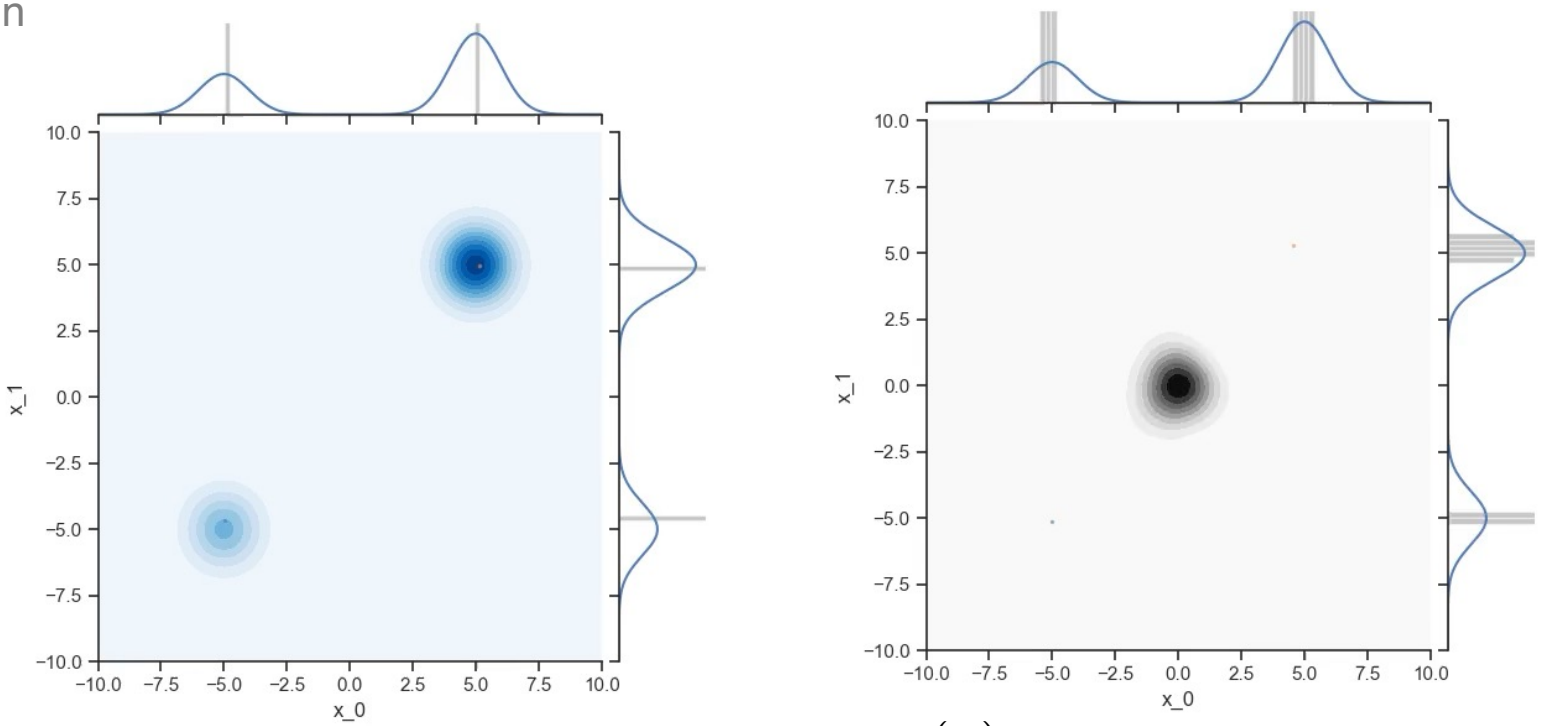
Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

- ▷ Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_* \| \rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_*(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

- ▷ Use an auxillary simple sampler to create data $x^i_{t+1} \sim \pi_{\mathrm{local}}(x^i_{t+1} | x^i_t)$
- ▷ Train



- ▷ Add non-local steps relying on flow proposals $x \sim \rho_\theta(x_i)$

Global-Local algorithm!

- ▷ Repeat and converge

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

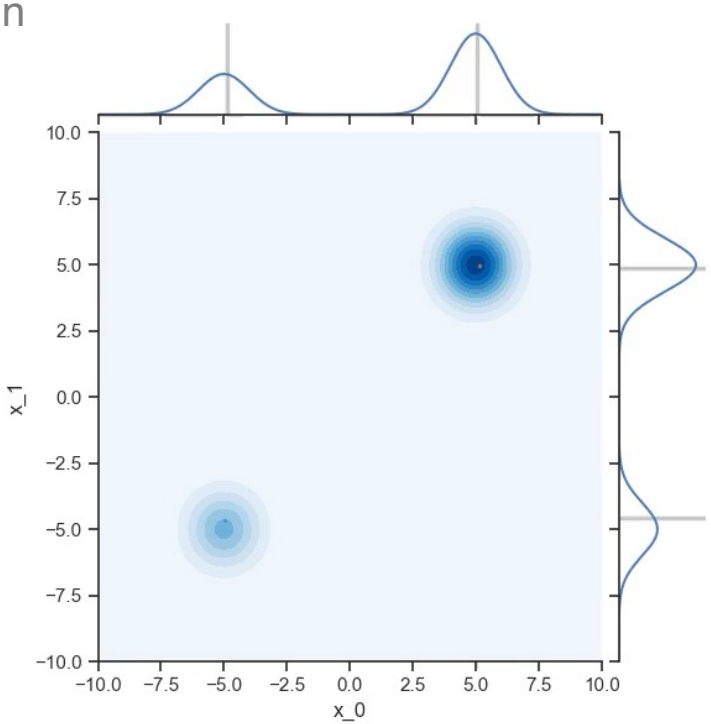
Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

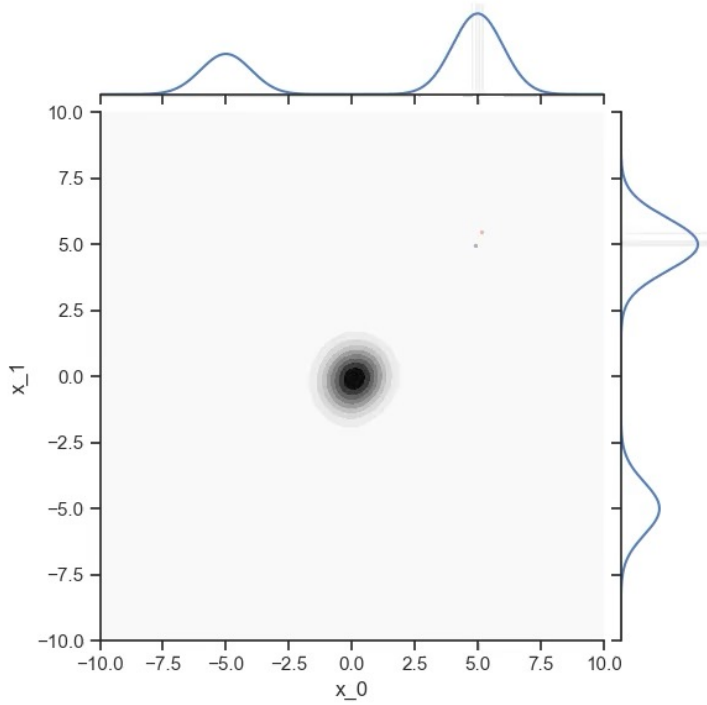
With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

- ▷ Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_* \| \rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_*(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

- ▷ Use an auxillary simple sampler to create data $x^i_{t+1} \sim \pi_{\mathrm{local}}(x^i_{t+1} | x^i_t)$
- ▷ Train





- ▷ Add non-local steps relying on flow proposals $x \sim \rho_\theta(x_i)$ *Global-Local algorithm!*
- ▷ Repeat and converge

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

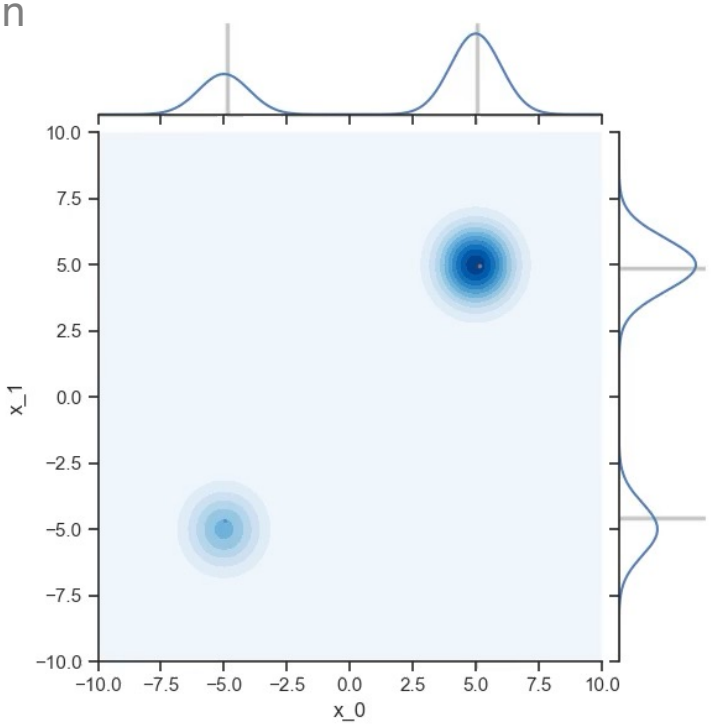
Second idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

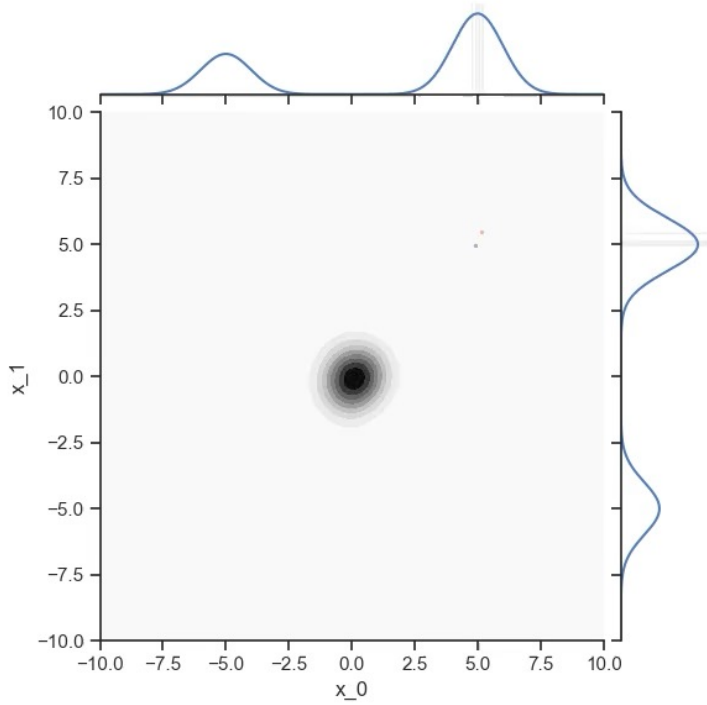
With Eric Vanden-Eijnden (Courant NYU) & Grant Rotskoff (Stanford)

- ▷ Maximum likelihood objective (a.k.a forward KL)

$$D_{\mathrm{KL}}(\rho_*\|\rho_\theta) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)}\rho_*(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} \log \rho_\theta(x_i) + \mathrm{Cst} \qquad x \sim \rho_*(x_i)$$

- ▷ Use an auxillary simple sampler to create data $x^i_{t+1} \sim \pi_{\mathrm{local}}(x^i_{t+1}|x^i_t)$
- ▷ Train





- ▷ Add non-local steps relying on flow proposals $x \sim \rho_\theta(x_i)$ *Global-Local algorithm!*
- ▷ Repeat and converge

No free lunch!

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

Third idea: Joining forces with annealing/SMC

Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks- B. McNaughton et al *PRE* 2020
(Continuously Repeated) Annealed Flow Transport Monte Carlo. Arbel, Matthews et al (2021 & 2022)
Karamanis et al . 'Accelerating Astronomical and Cosmological Inference with Preconditioned Monte Carlo. (2022)

Third idea: Joining forces with annealing/SMC

- ▷ Annealing to create progressively dataset of training

$$\rho_{\beta *}(x) = e^{-\beta U_*(x)}/Z$$

Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks- B. McNaughton et al *PRE* 2020
(Continuously Repeated) Annealed Flow Transport Monte Carlo. Arbel, Matthews et al (2021 & 2022)
Karamanis et al . 'Accelerating Astronomical and Cosmological Inference with Preconditioned Monte Carlo. (2022)

Third idea: Joining forces with annealing/SMC

- ▷ Annealing to create progressively dataset of training

$$\rho_{\beta *}(x) = e^{-\beta U_*(x)}/Z$$

From high-temperature repeat:

- Use $\rho_{\theta_{k-1}}^{\beta_{k-1}}(x) = \rho_p(x)$ in MCMC to sample $\rho_*^{\beta_k}(x)$
- Use $x_i \sim \rho_*^{\beta_k}(x_i)$ as data to train $\rho_{\theta_{k-1}}^{\beta_{k-1}}(x)$

Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks- B. McNaughton et al *PRE* 2020
(Continuously Repeated) Annealed Flow Transport Monte Carlo. Arbel, Matthews et al (2021 & 2022)
Karamanis et al . 'Accelerating Astronomical and Cosmological Inference with Preconditioned Monte Carlo. (2022)

Third idea: Joining forces with annealing/SMC

- Annealing to create progressively dataset of training

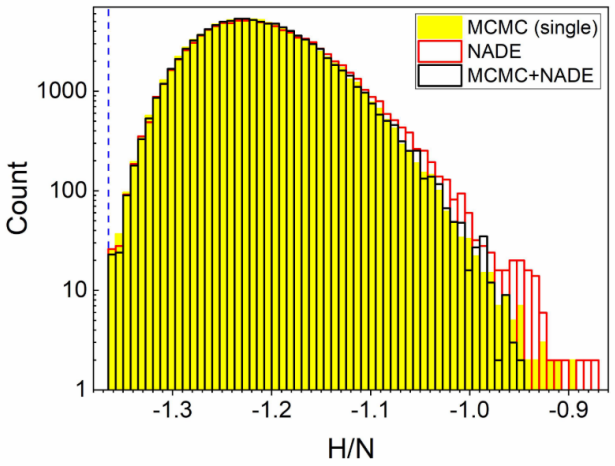
$$\rho_{\beta *}(x) = e^{-\beta U_*(x)}/Z$$

From high-temperature repeat:

- Use $\rho_{\theta_{k-1}}^{\beta_{k-1}}(x) = \rho_p(x)$ in MCMC to sample $\rho_*^{\beta_k}(x)$
- Use $x_i \sim \rho_*^{\beta_k}(x_i)$ as data to train $\rho_{\theta_{k-1}}^{\beta_{k-1}}(x)$

S. Pilati *PRE* 2020

2d – *Edwards Anderson*



Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks- B. McNaughton et al *PRE* 2020
(Continuously Repeated) Annealed Flow Transport Monte Carlo. Arbel, Matthews et al (2021 & 2022)
Karamanis et al . 'Accelerating Astronomical and Cosmological Inference with Preconditioned Monte Carlo. (2022)

Third idea: Joining forces with annealing/SMC

- Annealing to create progressively dataset of training

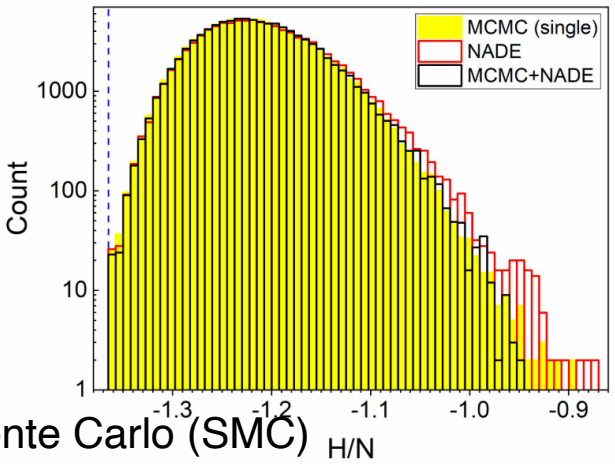
$$\rho_{\beta *}(x) = e^{-\beta U_*(x)}/Z$$

From high-temperature repeat:

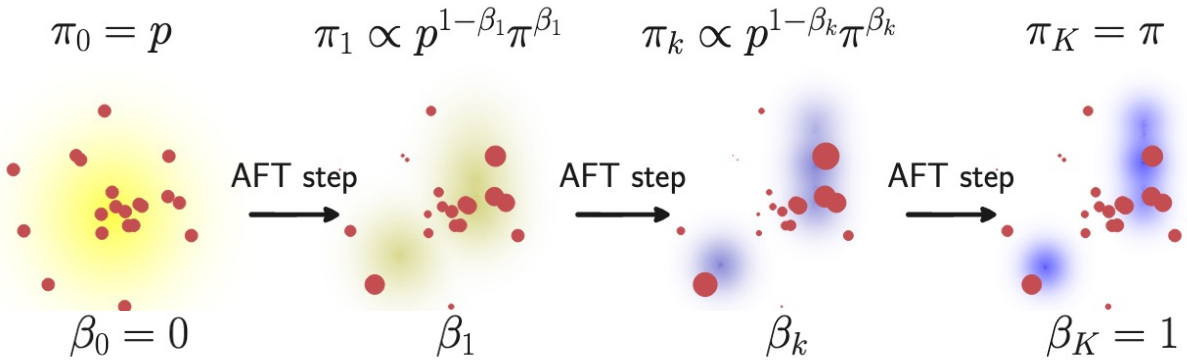
- Use $\rho_{\theta_{k-1}}^{\beta_{k-1}}(x) = \rho_p(x)$ in MCMC to sample $\rho_*^{\beta_k}(x)$
- Use $x_i \sim \rho_*^{\beta_k}(x_i)$ as data to train $\rho_{\theta_{k-1}}^{\beta_{k-1}}(x)$

S. Pilati *PRE* 2020

2d – *Edwards Anderson*



- (Continuously Repeated) Annealed Flow Transport
  - Add flow transport maps within steps of sequencial Monte Carlo (SMC)



Credit M. Arbel

Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks- B. McNaughton et al *PRE* 2020
(Continuously Repeated) Annealed Flow Transport Monte Carlo. Arbel, Matthews et al (2021 & 2022)
Karamanis et al . 'Accelerating Astronomical and Cosmological Inference with Preconditioned Monte Carlo. (2022)

Third idea: Joining forces with annealing/SMC

- Annealing to create progressively dataset of training

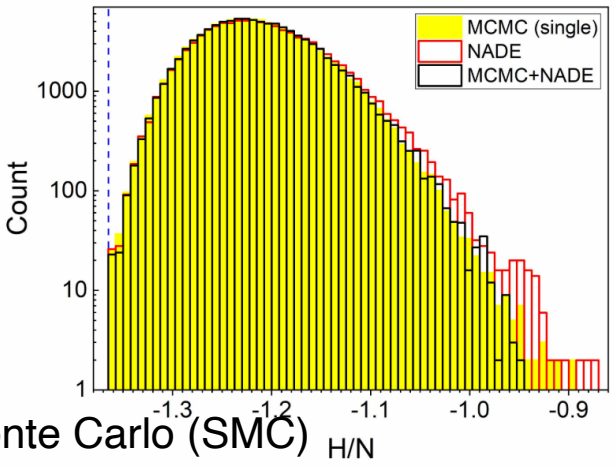
$$\rho_{\beta *}(x) = e^{-\beta U_*(x)}/Z$$

From high-temperature repeat:

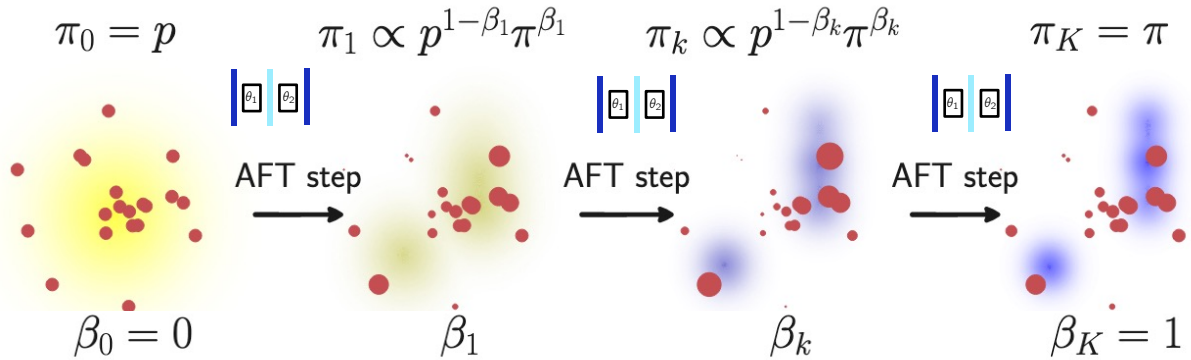
- Use $\rho_{\theta_{k-1}}^{\beta_{k-1}}(x) = \rho_p(x)$ in MCMC to sample $\rho_*^{\beta_k}(x)$
- Use $x_i \sim \rho_*^{\beta_k}(x_i)$ as data to train $\rho_{\theta_{k-1}}^{\beta_{k-1}}(x)$

S. Pilati *PRE* 2020

2d – *Edwards Anderson*



- (Continuously Repeated) Annealed Flow Transport
  - Add flow transport maps within steps of sequencial Monte Carlo (SMC)



Credit M. Arbel

Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks- B. McNaughton et al *PRE* 2020
(Continuously Repeated) Annealed Flow Transport Monte Carlo. Arbel, Matthews et al (2021 & 2022)
Karamanis et al . 'Accelerating Astronomical and Cosmological Inference with Preconditioned Monte Carlo. (2022)

Third idea: Joining forces with annealing/SMC

- Annealing to create progressively dataset of training

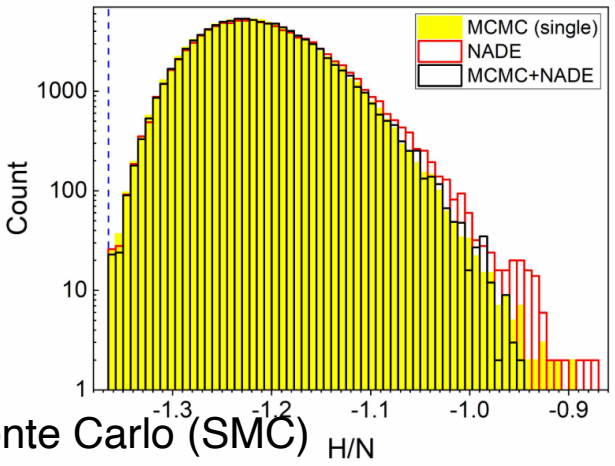
$$\rho_{\beta *}(x) = e^{-\beta U_*(x)}/Z$$

From high-temperature repeat:

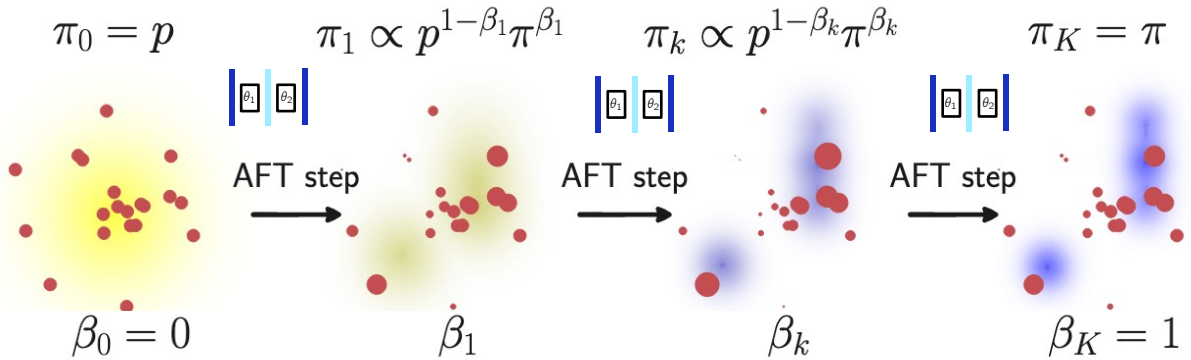
  - Use $\rho_{\theta_{k-1}}^{\beta_{k-1}}(x) = \rho_p(x)$ in MCMC to sample $\rho_*^{\beta_k}(x)$
  - Use $x_i \sim \rho_*^{\beta_k}(x_i)$ as data to train $\rho_{\theta_{k-1}}^{\beta_{k-1}}(x)$

S. Pilati *PRE* 2020

2d – *Edwards Anderson*



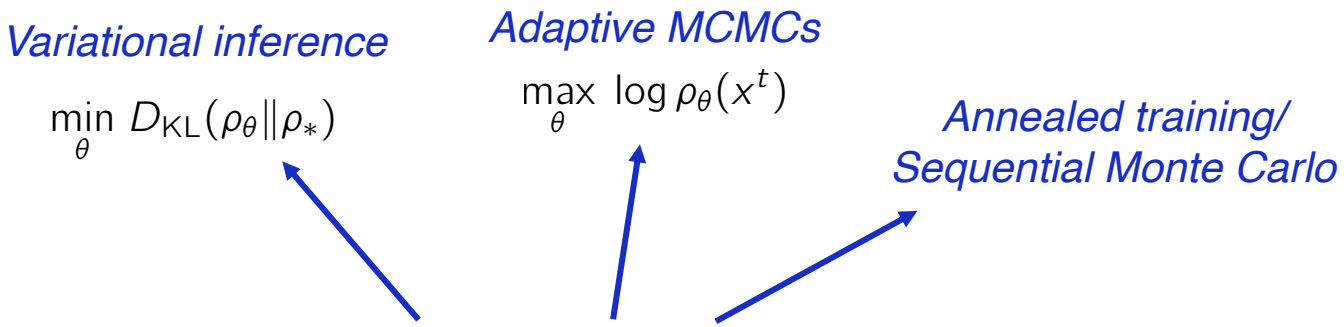
- (Continuously Repeated) Annealed Flow Transport
  - Add flow transport maps within steps of sequencial Monte Carlo (SMC)



Credit M. Arbel

- Preconditioned SMC (PMC)

Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks- B. McNaughton et al *PRE* 2020
(Continuously Repeated) Annealed Flow Transport Monte Carlo. Arbel, Matthews et al (2021 & 2022)
Karamanis et al . 'Accelerating Astronomical and Cosmological Inference with Preconditioned Monte Carlo. (2022)



Suppose you can train a model $\rho_\theta(x) \approx \rho_*(x)$, what do you gain?

▷A lot!

▷Examples

Example 1/2: Bayesian inference of Gravitational Waves parameters

Kaze W. K. Wong (CCA)

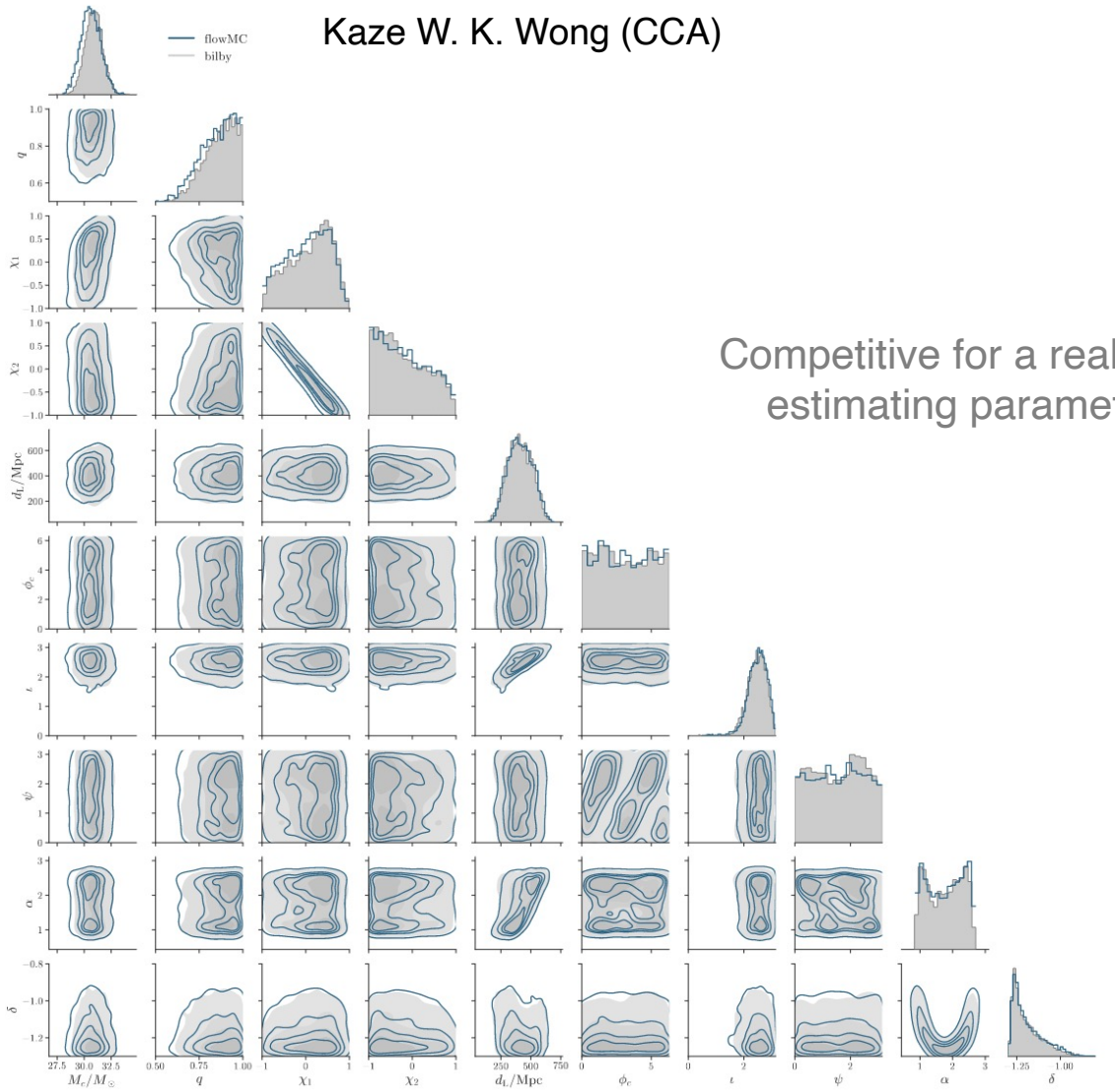


Figure 2. GW150914 posterior computed by our code (blue) and BILBY (gray).

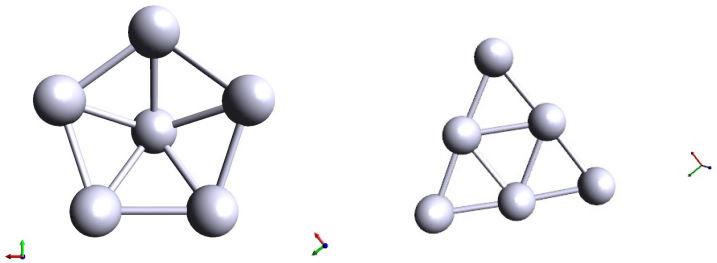
Competitive for a real-time application of estimating parameters of gravitational wave

Fast gravitational wave parameter estimation without compromises - Kaze W. K. Wong et al. arxiv.org/2302.05333

Example 2/2: Sampling metastable silver clusters

With Ana Molina Tarboda, Olga Lopez-Acevedo (Universidad de Antioquia), Pilar Cossio (Flatiron Institute)

- Target density given by Density Functional Theory: 2 metastable isomers

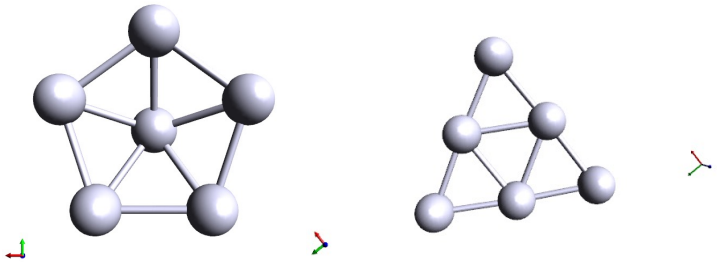


[in preparation]

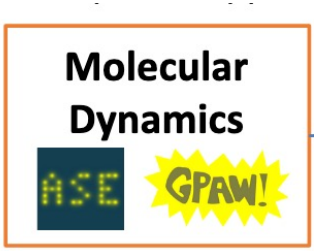
Example 2/2: Sampling metastable silver clusters

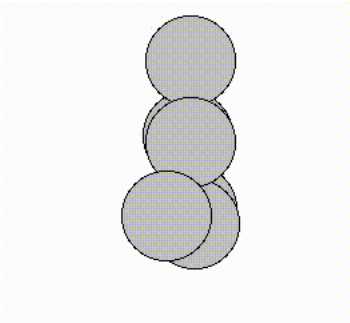
With Ana Molina Tarboda, Olga Lopez-Acevedo (Universidad de Antioquia), Pilar Cossio (Flatiron Institute)

- Target density given by Density Functional Theory: 2 metastable isomers



- Local sampler: Molecular Dynamics



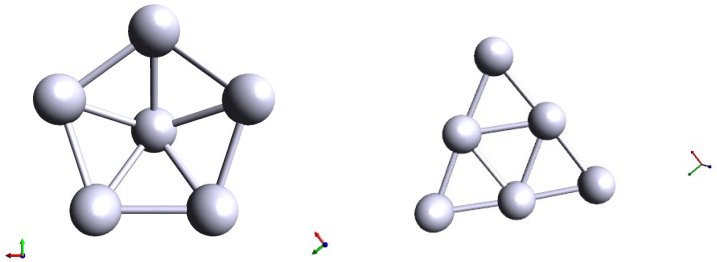


[in preparation]

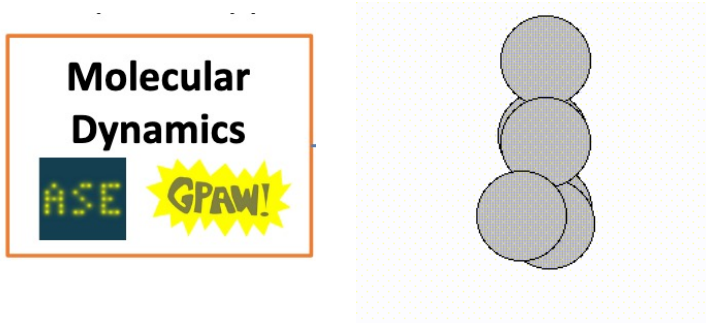
Example 2/2: Sampling metastable silver clusters

With Ana Molina Tarboda, Olga Lopez-Acevedo (Universidad de Antioquia), Pilar Cossio (Flatiron Institute)

- Target density given by Density Functional Theory: 2 metastable isomers

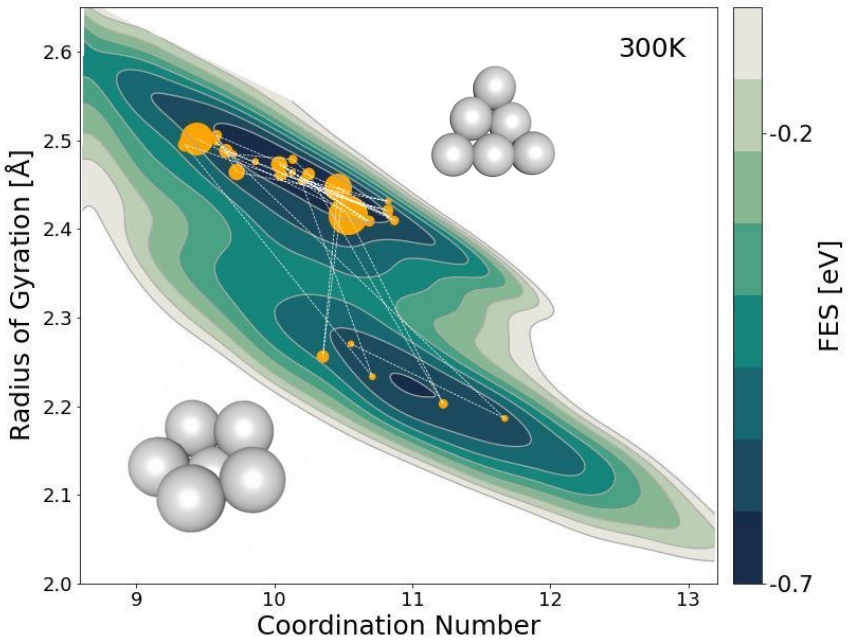


- Local sampler: Molecular Dynamics

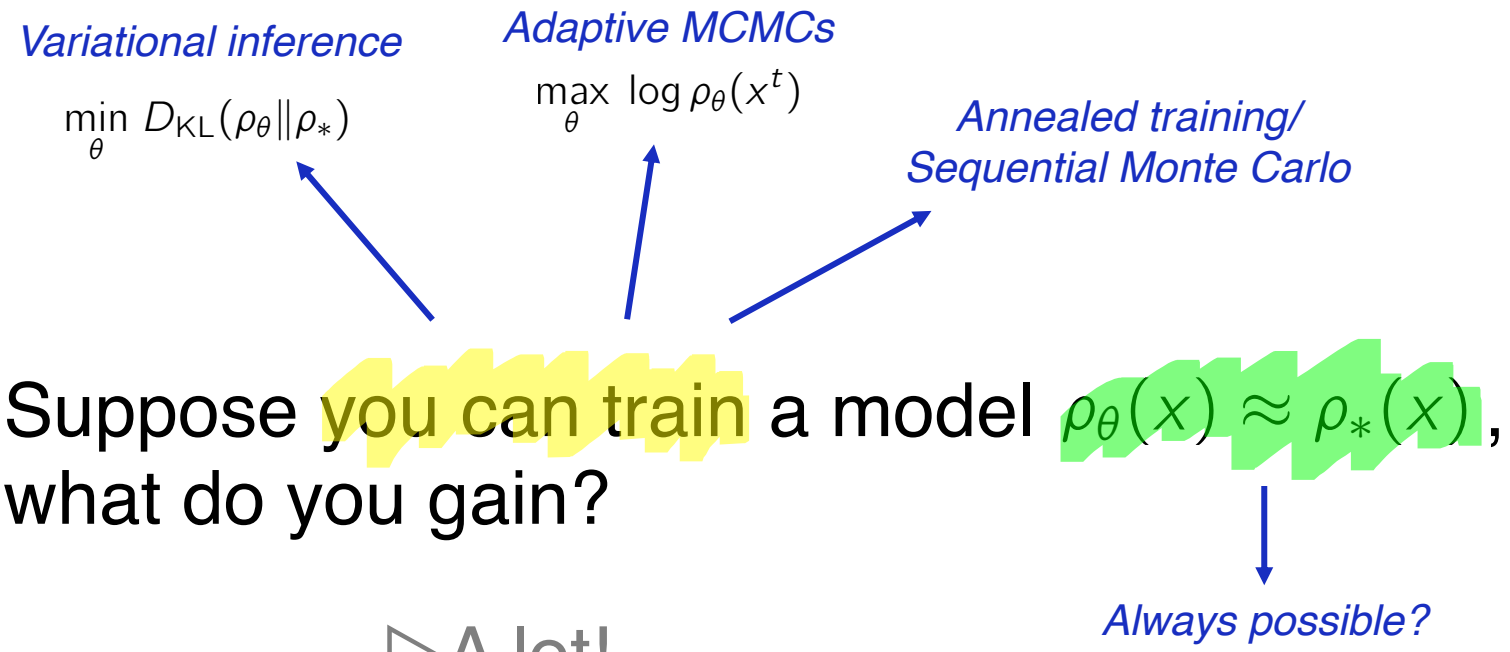


- Adaptive MCMC jumping between isomers
  - Using here a mixture of flows

2 dimension projection of
the free energy surface



[in preparation]


Variational inference
$\min_\theta D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$
Adaptive MCMCs
$\max_\theta \log \rho_\theta(x^t)$
Annealed training/
Sequential Monte Carlo
Suppose you can train a model $\rho_\theta(x) \approx \rho_*(x)$,
what do you gain?
Always possible?
▷A lot!

Is the model good enough to be useful? Is the sampling converged?

Is the model good enough to be useful? Is the sampling converged?

- Key observables:
  - Neural Importance Sampling

$$w_i = \frac{\rho_*(x_i)/\rho_\theta(x_i)}{\sum_{i=1}^{N} \rho_*(x_i)/\rho_\theta(x_i)}$$

Participation ratio

$$\mathrm{PR} = \frac{(\sum_{i=1}^{N} w_i)^2}{N \sum_{i=1}^{N} w_i^2} \in [0, 1]$$

Is the model good enough to be useful? Is the sampling converged?

- Key observables:
  - Neural Importance Sampling

$$w_i = \frac{\rho_*(x_i)/\rho_\theta(x_i)}{\sum_{i=1}^N \rho_*(x_i)/\rho_\theta(x_i)}$$

  Participation ratio

$$\mathrm{PR} = \frac{(\sum_{i=1}^N w_i)^2}{N \sum_{i=1}^N w_i^2} \in [0, 1]$$

  - Independent Metropolis Hastings:
    Acceptance rate of the proposal

$$\mathrm{acc}(x^i_{t+1}|x^i_t) = \min\left[1, \frac{\rho_*(x^i_{t+1})\rho_\theta(x^i_t)}{\rho_*(x^i_t)\rho_\theta(x^i_{t+1})}\right]$$

Is the model good enough to be useful? Is the sampling converged?

- Key observables:
  - Neural Importance Sampling $w_i = \frac{\rho_*(x_i)/\rho_\theta(x_i)}{\sum_{i=1}^N \rho_*(x_i)/\rho_\theta(x_i)}$

    Participation ratio $\mathrm{PR} = \frac{(\sum_{i=1}^N w_i)^2}{N \sum_{i=1}^N w_i^2} \in [0, 1]$
  - Independent Metropolis Hastings:

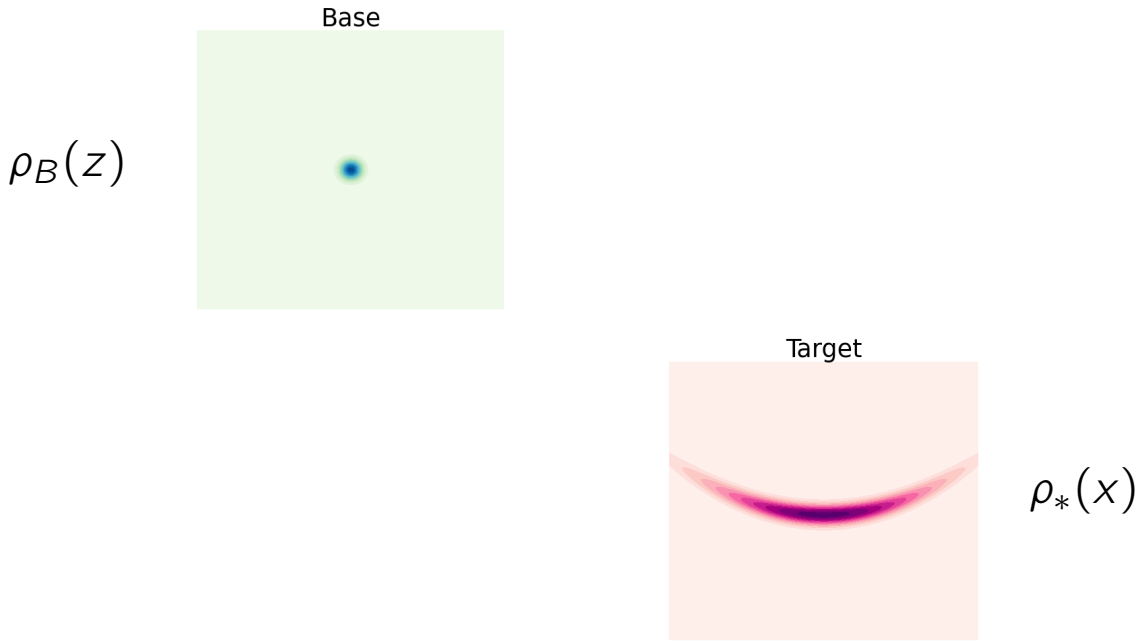
    Acceptance rate of the proposal $\mathrm{acc}(x^i_{t+1}|x^i_t) = \min\left[1, \frac{\rho_*(x^i_{t+1})\rho_\theta(x^i_t)}{\rho_*(x^i_t)\rho_\theta(x^i_{t+1})}\right]$
- What is going to make the method fail? Discrepancies $\rho_\theta(x) \approx \rho_*(x)$
  - Posteriors/Targets too intricate to be represented
  - Dimensionality …

Is the model good enough to be useful? Is the sampling converged?

- Key observables:
  - Neural Importance Sampling $w_i = \frac{\rho_*(x_i)/\rho_\theta(x_i)}{\sum_{i=1}^N \rho_*(x_i)/\rho_\theta(x_i)}$

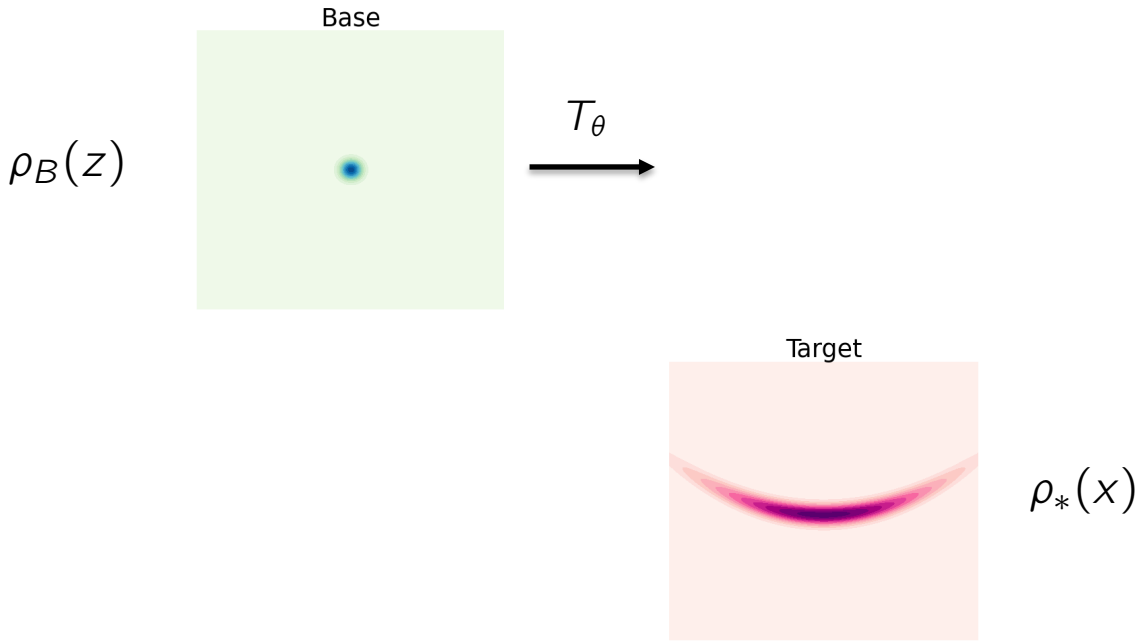
    Participation ratio $\mathrm{PR} = \frac{(\sum_{i=1}^N w_i)^2}{N \sum_{i=1}^N w_i^2} \in [0, 1]$
  - Independent Metropolis Hastings:
    Acceptance rate of the proposal $\mathrm{acc}(x^i_{t+1}|x^i_t) = \min\left[1, \frac{\rho_*(x^i_{t+1})\rho_\theta(x^i_t)}{\rho_*(x^i_t)\rho_\theta(x^i_{t+1})}\right]$
- What is going to make the method fail? Discrepancies $\rho_\theta(x) \approx \rho_*(x)$
  - Posteriors/Targets too intricate to be represented
  - Dimensionality …
- Can we be less ambitious but still take advantage of the generative models?

Pre-conditioning local samplers with flows



[Li & Wang *PRL 2018,* Wu et al. *PRL 2019,* Noé et al. *Science 2019,* Hoffman et al *arXiv:1903.03704]*

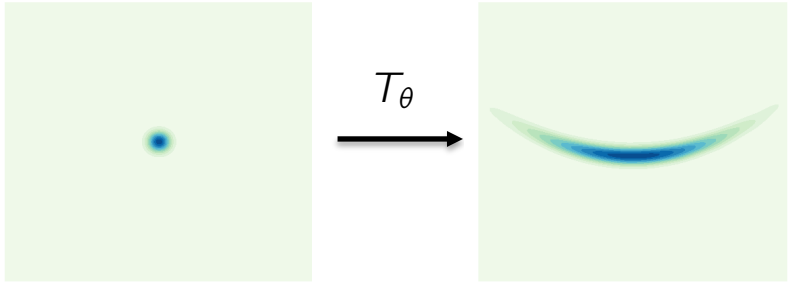
Pre-conditioning local samplers with flows



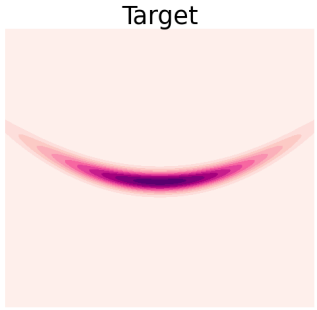
[Li & Wang *PRL 2018,* Wu et al. *PRL 2019,* Noé et al. *Science 2019,* Hoffman et al *arXiv:1903.03704]*

Pre-conditioning local samplers with flows

$\rho_B(z)$



$$\rho_\theta(x) = \rho_B(T_\theta^{-1}(x)) \left|\nabla_z T_\theta^{-1}(x)\right|$$

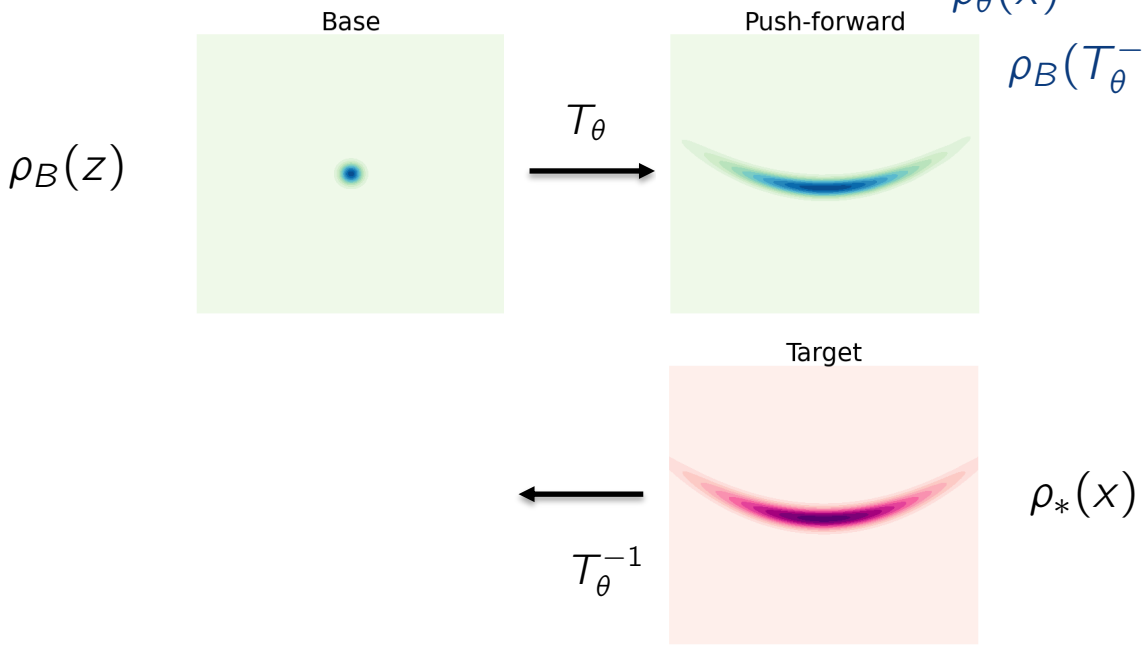


$\rho_*(x)$

[Li & Wang *PRL 2018,* Wu et al. *PRL 2019,* Noé et al. *Science 2019,* Hoffman et al *arXiv:1903.03704]*

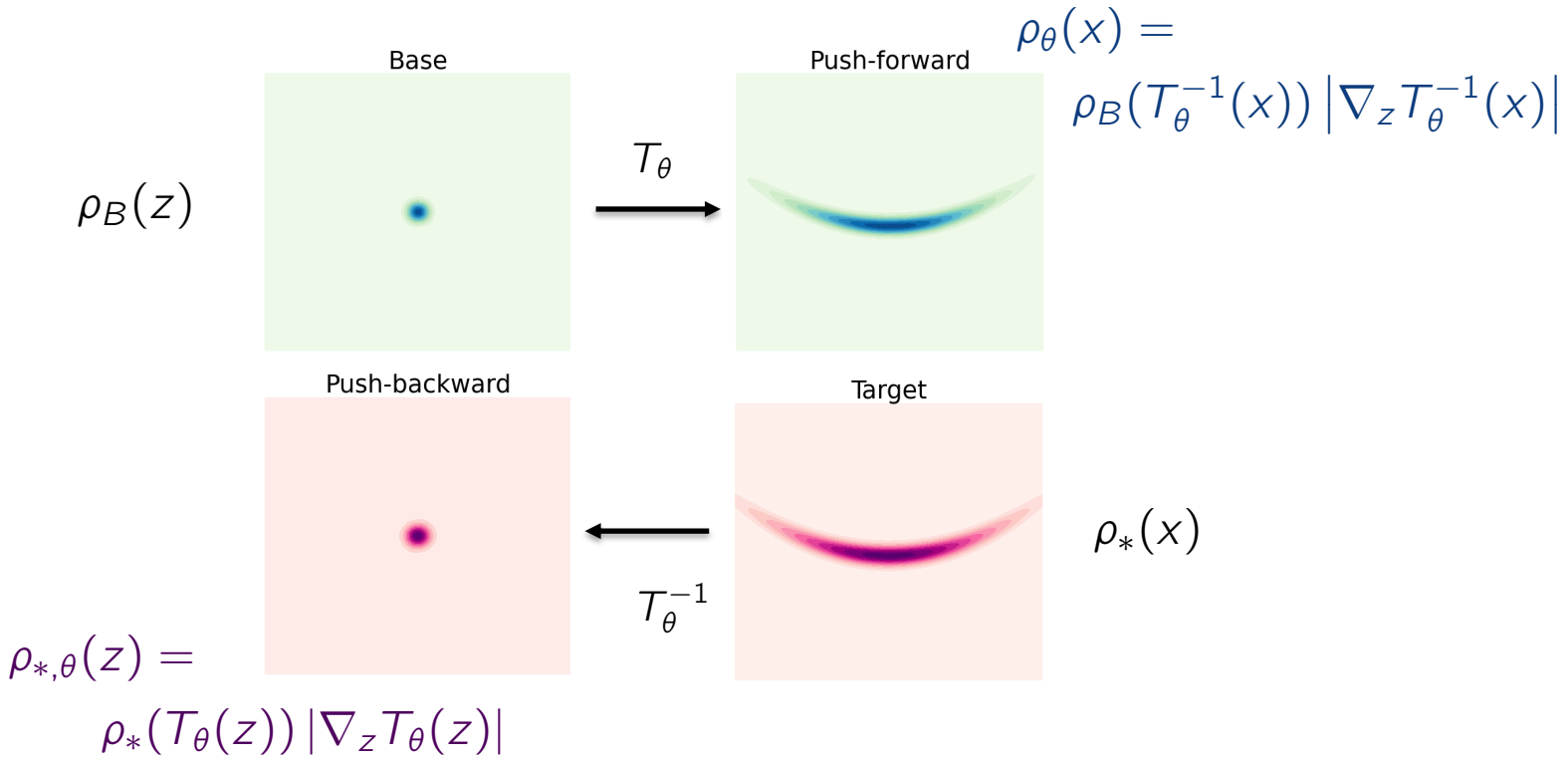
Pre-conditioning local samplers with flows

$$\rho_\theta(x) = \rho_B(T_\theta^{-1}(x)) \left|\nabla_z T_\theta^{-1}(x)\right|$$



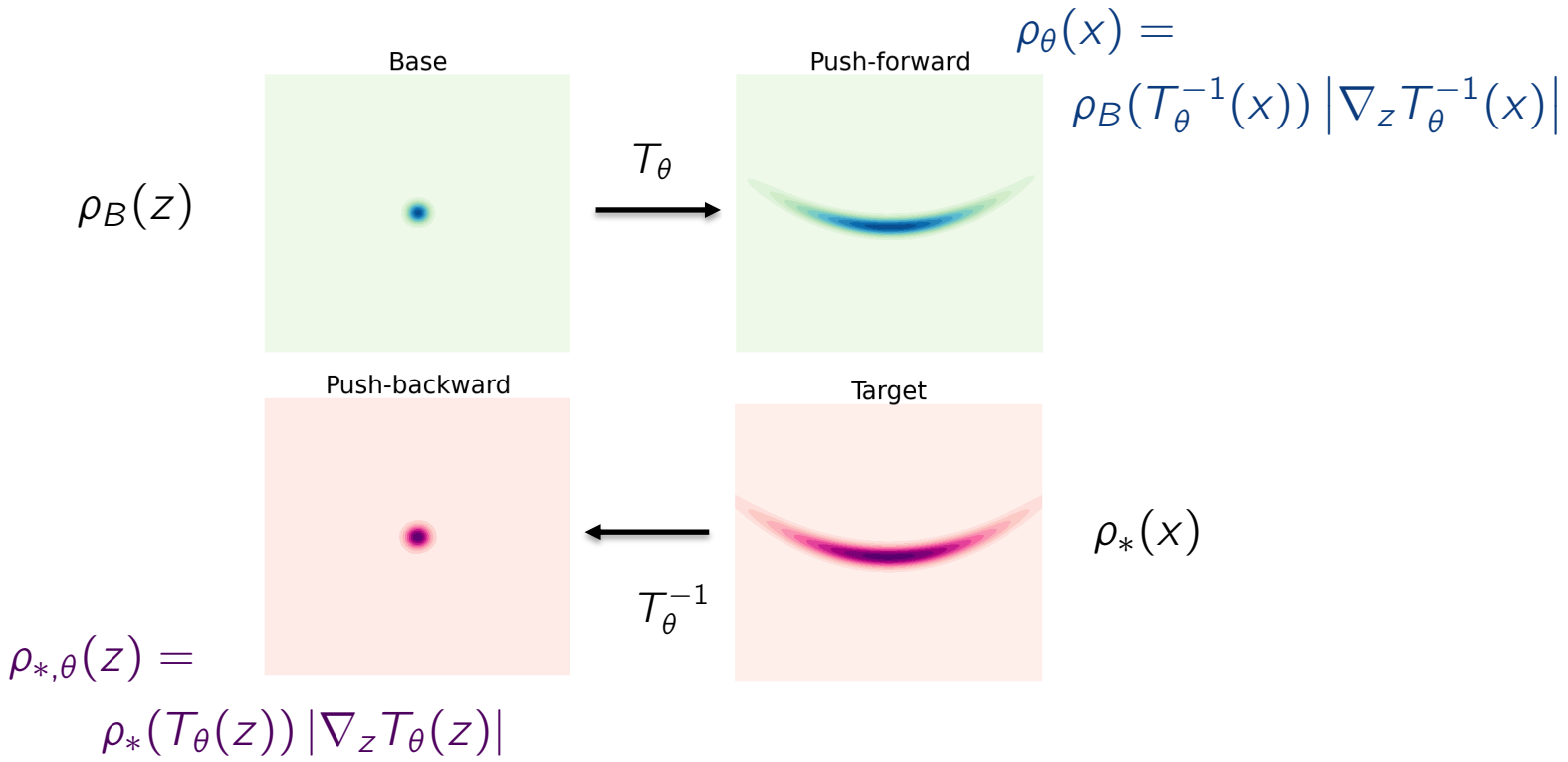
[Li & Wang *PRL 2018,* Wu et al. *PRL 2019,* Noé et al. *Science 2019,* Hoffman et al *arXiv:1903.03704]*

Pre-conditioning local samplers with flows



[Li & Wang *PRL 2018,* Wu et al. *PRL 2019,* Noé et al. *Science 2019,* Hoffman et al *arXiv:1903.03704]*

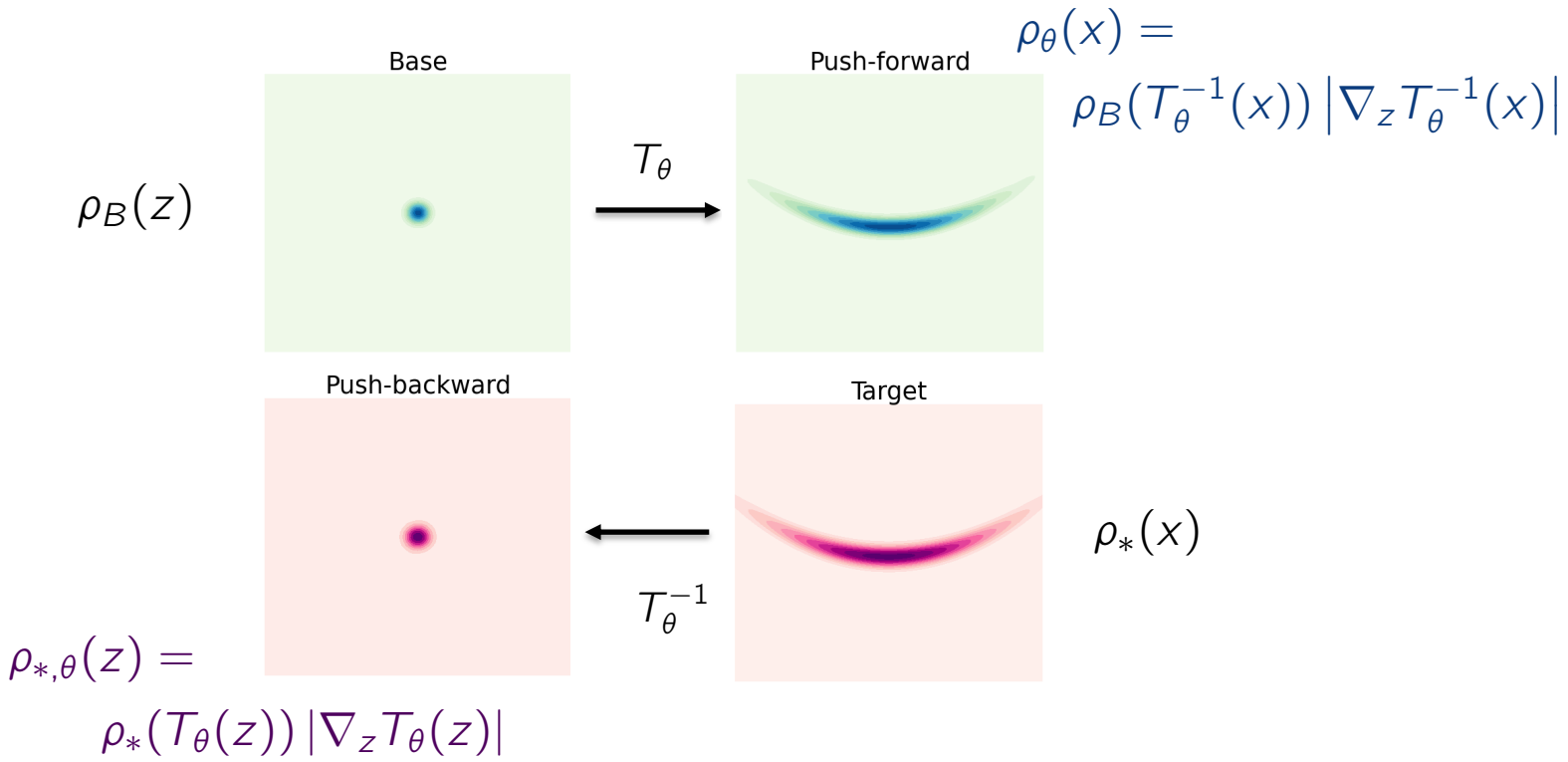
Pre-conditioning local samplers with flows



- ▷ "Neutra-MCMC": Run a gradient-based local sampler (HMC, MALA) in the latent space

[Li & Wang *PRL 2018,* Wu et al. *PRL 2019,* Noé et al. *Science 2019,* Hoffman et al *arXiv:1903.03704]*

Pre-conditioning local samplers with flows



- ▷ "Neutra-MCMC": Run a gradient-based local sampler (HMC, MALA) in the latent space
- ▷ Why? Local samplers are more robust in high-dimension, and are greatly helped by preconditionning

[Li & Wang *PRL 2018,* Wu et al. *PRL 2019,* Noé et al. *Science 2019,* Hoffman et al *arXiv:1903.03704]*

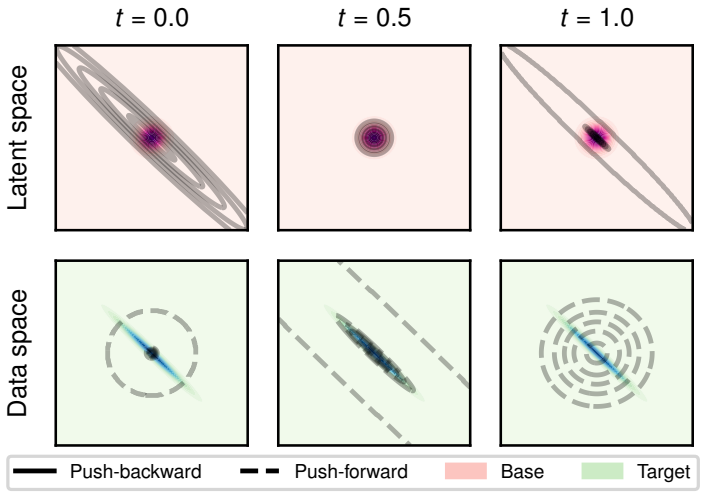
Flow proposals vs latent local jumps 1/2

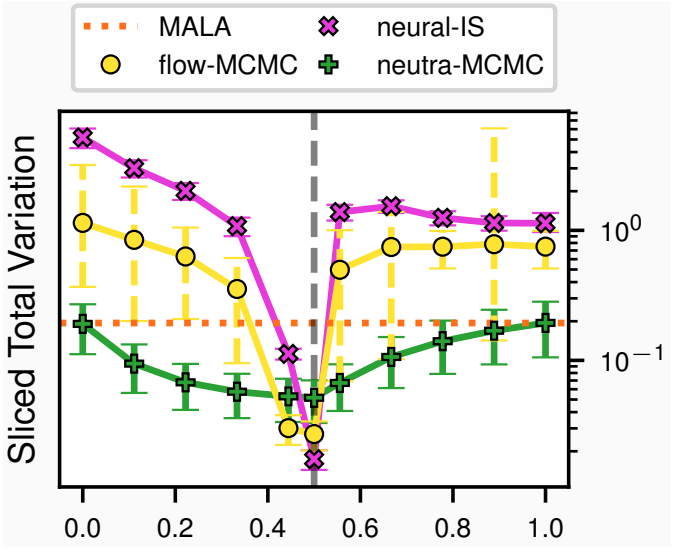
With Louis Grenioux & Christoph Schönle (École Polytechnique) [Grenioux, Durmus, Moulines & MG, *ICML 2023*]

Flow proposals vs latent local jumps 1/2

With Louis Grenioux & Christoph Schönle (École Polytechnique) [Grenioux, Durmus, Moulines & MG, *ICML 2023*]

▷ Preconditioned gradient-based sampling typically more robust to poor flows

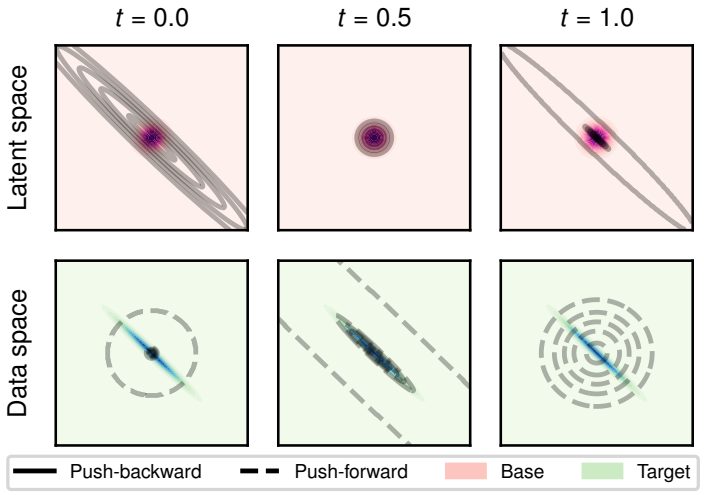


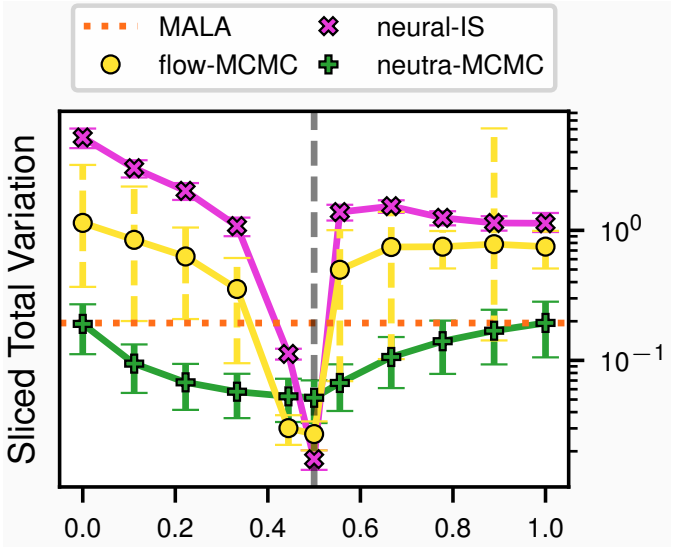


Flow proposals vs latent local jumps 1/2

With Louis Grenioux & Christoph Schönle (École Polytechnique) [Grenioux, Durmus, Moulines & MG, *ICML 2023*]

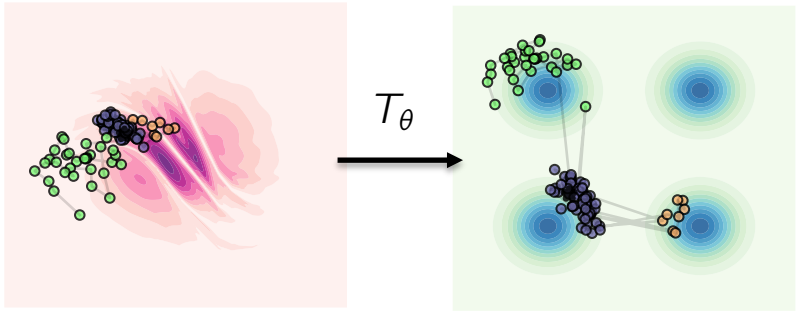
▷ Preconditioned gradient-based sampling typically more robust to poor flows





▷ But learned transport maps do not eras multimodality

NeutraMALA

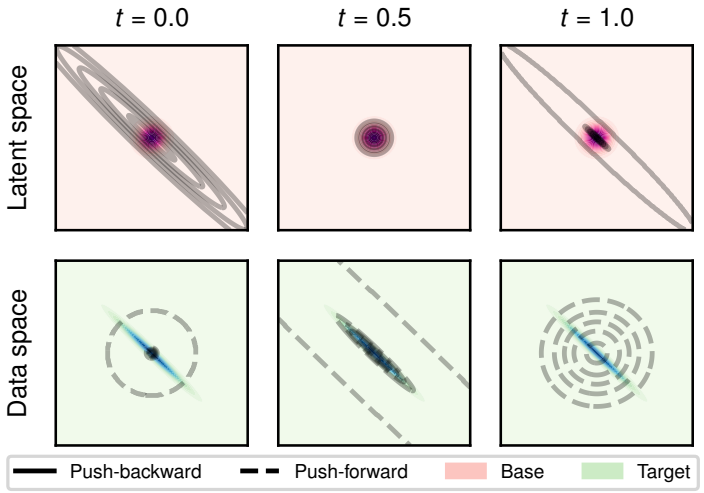


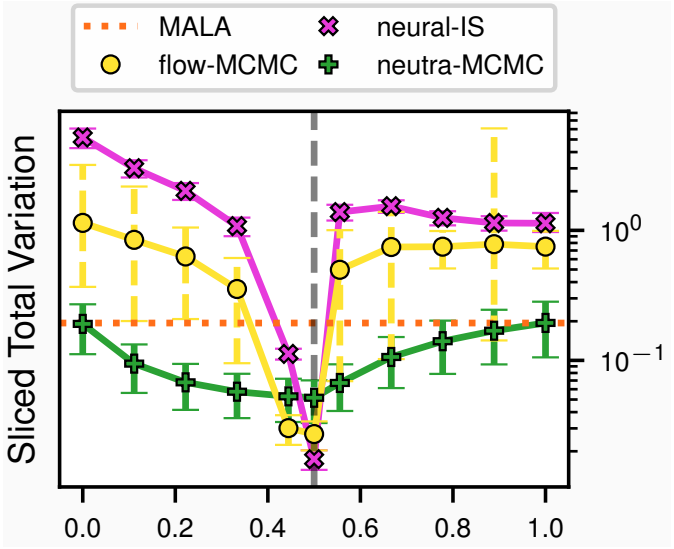
[Cornish et al. ICML 2020. Salmona et al. Neurips 2022.]

Flow proposals vs latent local jumps 1/2

With Louis Grenioux & Christoph Schönle (École Polytechnique) [Grenioux, Durmus, Moulines & MG, *ICML 2023*]

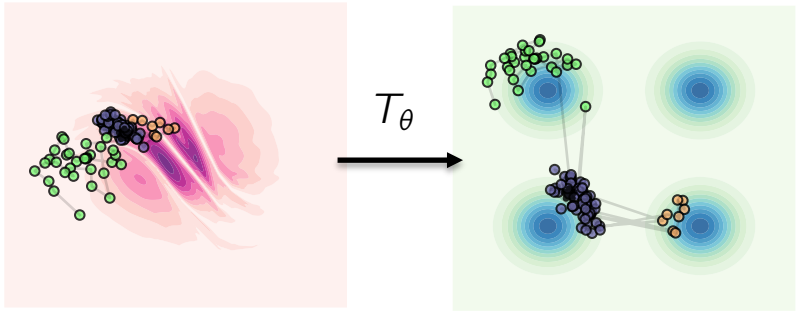
▷ Preconditioned gradient-based sampling typically more robust to poor flows



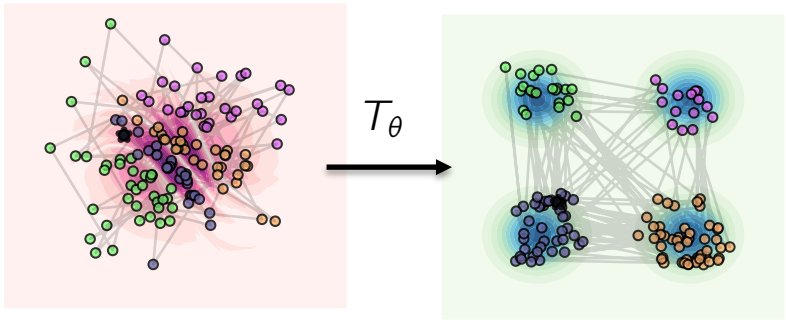


▷ But learned transport maps do not eras multimodality

NeutraMALA



FlowMC



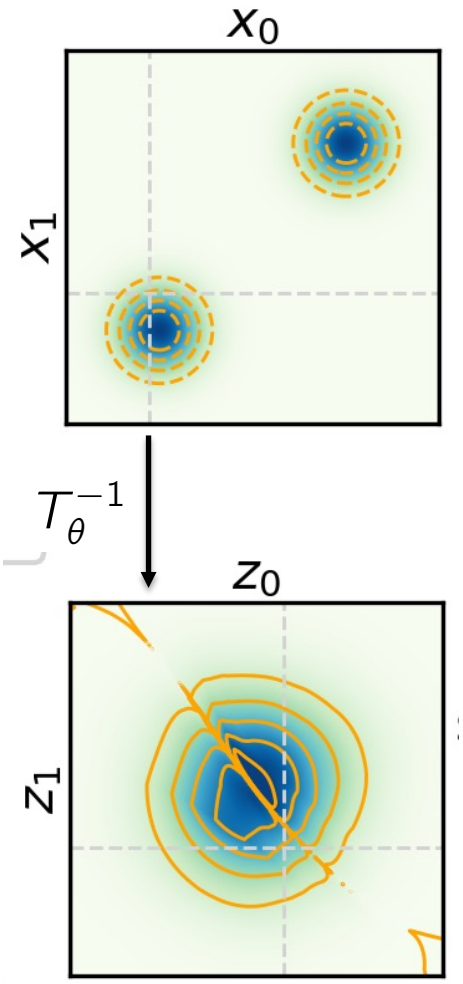
[Cornish et al. ICML 2020. Salmona et al. Neurips 2022.]

Flow proposals vs latent ~~local~~ partial jumps 2/2

With Louis Grenioux & Christoph Schönle (École Polytechnique)

[Schönle & MG, *NeurIPS Workshop 2023*]

▷ Partial updates in latent space allow mixing while improving acceptance: GflowMC

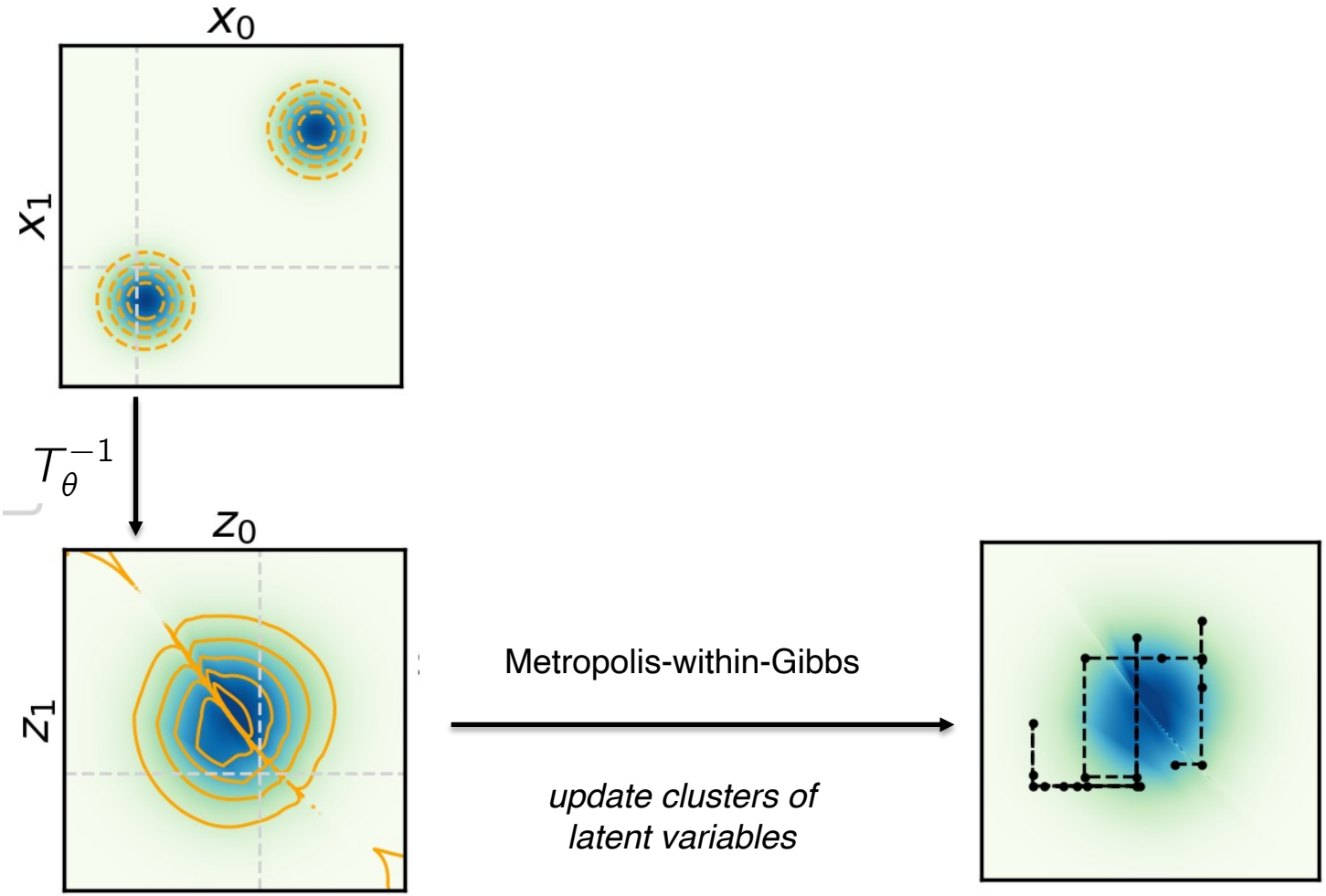


Flow proposals vs latent ~~local~~ partial jumps 2/2

With Louis Grenioux & Christoph Schönle (École Polytechnique)

[Schönle & MG, *NeurIPS Workshop 2023*]

▷ Partial updates in latent space allow mixing while improving acceptance: GflowMC

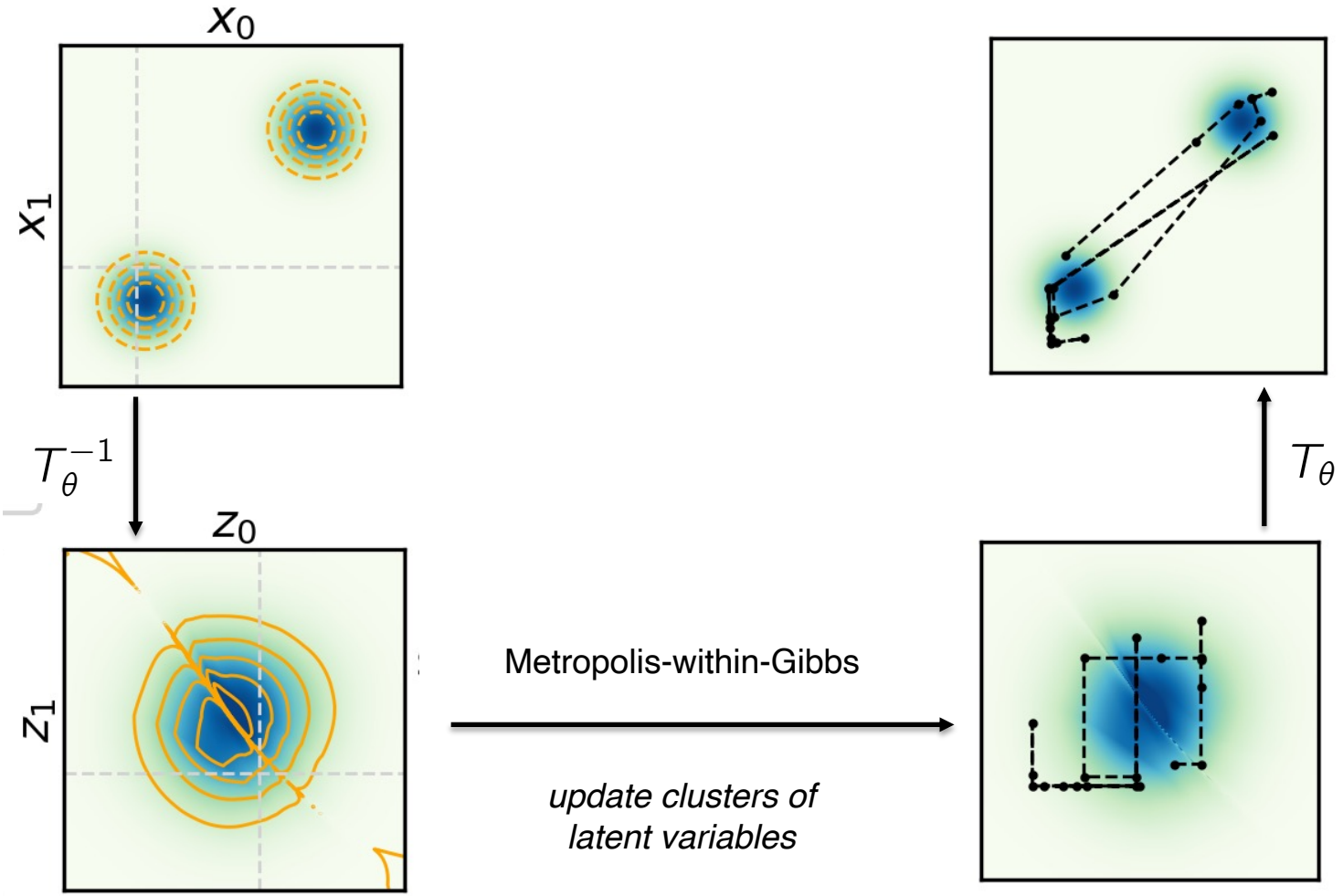


Flow proposals vs latent ~~local~~ partial jumps 2/2

With Louis Grenioux & Christoph Schönle (École Polytechnique)

[Schönle & MG, *NeurIPS Workshop 2023*]

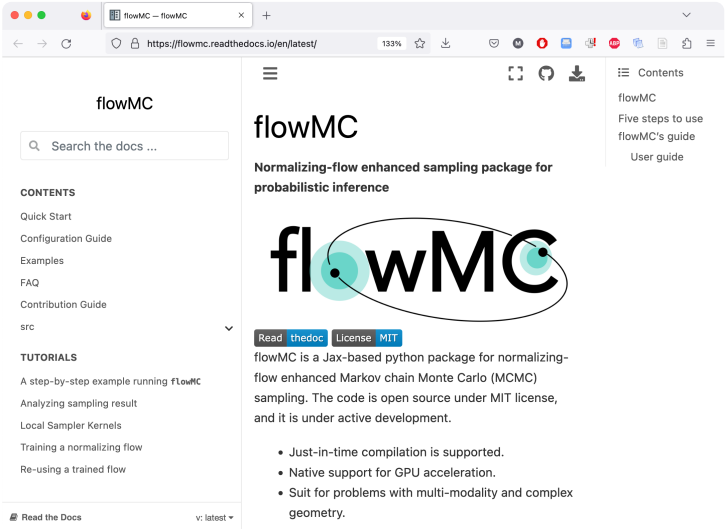
▷ Partial updates in latent space allow mixing while improving acceptance: GflowMC



2 Python packages I am aware off

- flowMC
  Wong et al. 2022
  (adaptive MCMC)

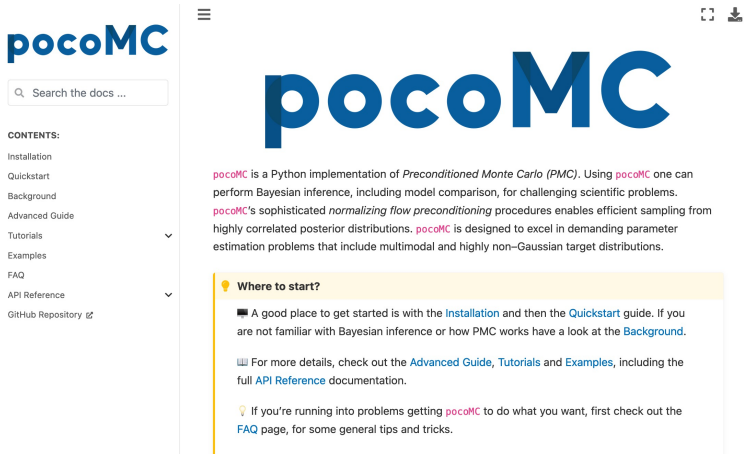
```
> pip install flowmc
```



- pocoMC
  Karamanis et al. 2022
  (preconditioned SMC)

```
> pip install pocomc
```

PyTorch



Perspectives & Conclusions

- ▷ Progess in generative modelling suggests a road to machine learning enhance samplers.
- ▷ Reaching the level of training accuracy required is not trivial for complex systems.
- ▷ Adaptations are possible for continuous time generative models, but are costly.
- ▷ There is no free lunch in finding the modes.
- ▷ These methods appear to be well-suited for Bayesian inference problems in moderate dimension (~100)!

▷Thank you

Collaborators:
Grant Rotskoff (Stanford), Éric Vanden-Eijnden (Courant Institute, NYU)
Louis Grenioux, Christoph Schönle, Alain Durmus & Eric Moulines (École Polytechnique)
Pilar Cossio (Flatiron, CCM), Olga Lopez Acevedo & Ana Molina Taborda (Universidad de Antioquia)
Kaze Wong & Dan Foreman-Mackey (Flatiron, CCA)

Based on

- M. Gabrié, G. M. Rotskof, and E. Vanden-Eijnden, ‘Efficient Bayesian Sampling Using Normalizing Flows to Assist Markov Chain Monte Carlo Methods’, *ICML workshop 2021*
- M. Gabrié, G. M. Rotskoff, and E. Vanden-Eijnden, ‘Adaptive Monte Carlo augmented with normalizing flows’, *PNAS 2022*
- J. A. Brofos, M. Gabrié, M. A. Brubaker, and R. R. Lederman, ‘Adaptation of the Independent Metropolis-Hastings Sampler with Normalizing Flow Proposals’. *AISTAT 2022*
- S. Samsonov, E. Lagutin, M. Gabrié, A. Durmus, A. Naumov, and E. Moulines, ‘Local-Global MCMC kernels: the best of both worlds’, in *Neural Information Processing Systems*, 2022.
- K. W. K. Wong, M. Gabrié, and D. Foreman-Mackey, ‘flowMC: Normalizing-flow enhanced sampling package for probabilistic inference in Jax’. Accepted at *Journal of Open Science Software 2023*
- L. Grenioux, A. Durmus, É. Moulines, and M. Gabrié, ‘On Sampling with Approximate Transport Maps’. ICML 2023
- L. Grenioux, É. Moulines, and M. Gabrié, ‘Balanced Training of Energy Based Models’, SPIGM Workshop, ICML 2023
- C. Schönle and M. Gabrié, ‘Optimizing Markov Chain Monte Carlo Convergence with Normalizing Flows and Gibbs Sampling’. AI for Science Workshop, NeurIPS 2023

One key idea is physics informed learning: for instance system tailored base measure

- Φ^4 model: Retain local couplings of the action

$$U(\Phi) = \sum_{i \in \Lambda} \big[-\beta(\phi_{i+e_1}\phi_i + \phi_{i+e_2}\phi_i) + \phi_i^2 + \lambda(\phi_i^2 - 1)^2 \big]$$

[MG, Rotskoff & Vanden-Eijnden – *PNAS 2022*]
Cf Grant's talk

One key idea is physics informed learning: for instance system tailored base measure

▷ Φ^4 model: Retain local couplings of the action

$$U(\Phi) = \sum_{i \in \Lambda} \big[-\beta(\phi_{i+e_1}\phi_i + \phi_{i+e_2}\phi_i) + \phi_i^2 + \lambda(\phi_i^2 - 1)^2 \big]$$

coupling term

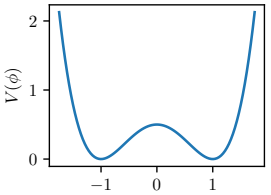
[MG, Rotskoff & Vanden-Eijnden – *PNAS 2022*]
Cf Grant's talk

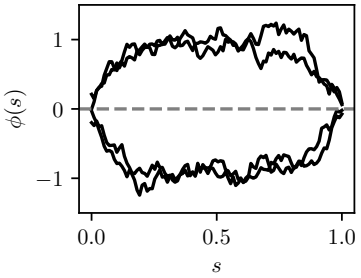
One key idea is physics informed learning: for instance system tailored base measure

▷ Φ^4 model: Retain local couplings of the action

$$U(\Phi) = \sum_{i \in \Lambda} \left[-\beta(\phi_{i+e_1}\phi_i + \phi_{i+e_2}\phi_i) + \phi_i^2 + \lambda(\phi_i^2 - 1)^2 \right]$$

coupling term *local potential* →





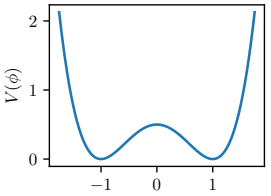
[MG, Rotskoff & Vanden-Eijnden – *PNAS 2022*]
Cf Grant's talk

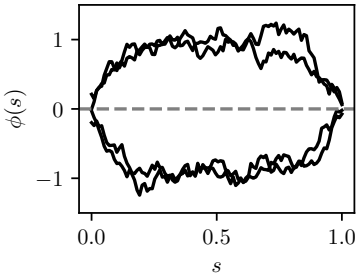
One key idea is physics informed learning: for instance system tailored base measure

▷ Φ^4 model: Retain local couplings of the action

$$U(\Phi) = \sum_{i \in \Lambda} \left[-\beta(\phi_{i+e_1}\phi_i + \phi_{i+e_2}\phi_i) + \phi_i^2 + \lambda(\phi_i^2 - 1)^2 \right]$$

coupling term *local potential* →





$$U_{\mathrm{B}}(\Phi) = \sum_{i \in \Lambda} \left[-\beta(\phi_{i+e_1}\phi_i + \phi_{i+e_2}\phi_i) + (1 - 2\lambda)\phi_i^2 \right]$$

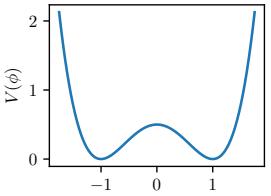
[MG, Rotskoff & Vanden-Eijnden – *PNAS 2022*]
Cf Grant's talk

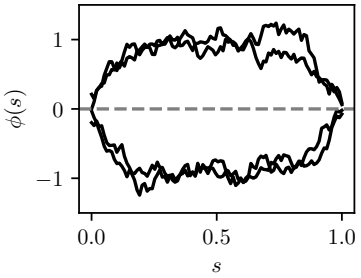
One key idea is physics informed learning: for instance system tailored base measure

▷ Φ^4 model: Retain local couplings of the action

$$U(\Phi) = \sum_{i \in \Lambda} \left[-\beta(\phi_{i+e_1}\phi_i + \phi_{i+e_2}\phi_i) + \phi_i^2 + \lambda(\phi_i^2 - 1)^2 \right]$$

coupling term *local potential* →



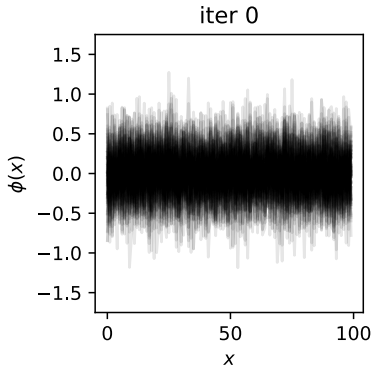


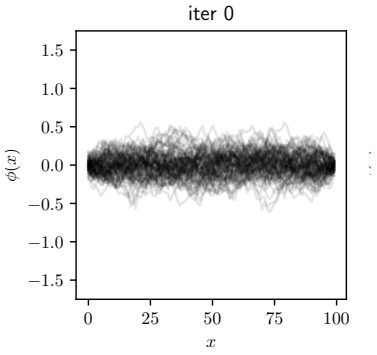
$$U_{\mathrm{B}}(\Phi) = \sum_{i \in \Lambda} \left[-\beta(\phi_{i+e_1}\phi_i + \phi_{i+e_2}\phi_i) + (1 - 2\lambda)\phi_i^2 \right]$$

Base samples

Gaussian white noise

$$\rho_{\mathrm{B}}(\Phi) \propto e^{-\beta U_{\mathrm{B}}(\Phi)}$$





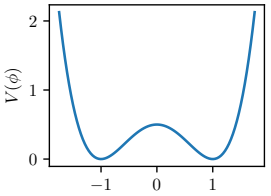
[MG, Rotskoff & Vanden-Eijnden – *PNAS 2022*]
Cf Grant's talk

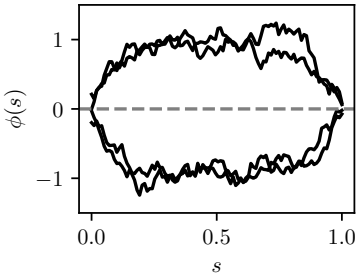
One key idea is physics informed learning: for instance system tailored base measure

▷ Φ^4 model: Retain local couplings of the action

$$U(\Phi) = \sum_{i \in \Lambda} \left[-\beta(\phi_{i+e_1}\phi_i + \phi_{i+e_2}\phi_i) + \phi_i^2 + \lambda(\phi_i^2 - 1)^2 \right]$$

coupling term *local potential* →



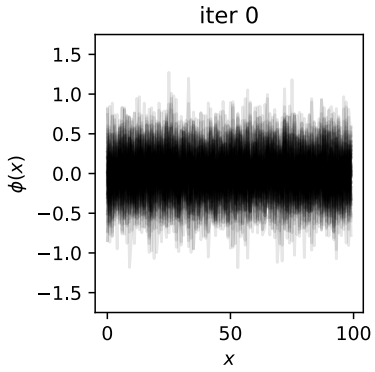


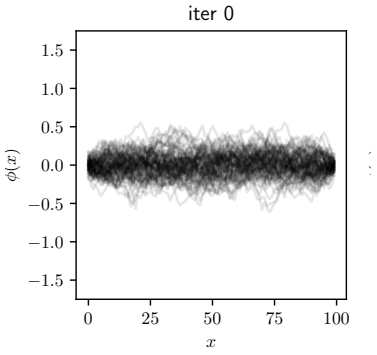
$$U_{\mathrm{B}}(\Phi) = \sum_{i \in \Lambda} \left[-\beta(\phi_{i+e_1}\phi_i + \phi_{i+e_2}\phi_i) + (1 - 2\lambda)\phi_i^2 \right]$$

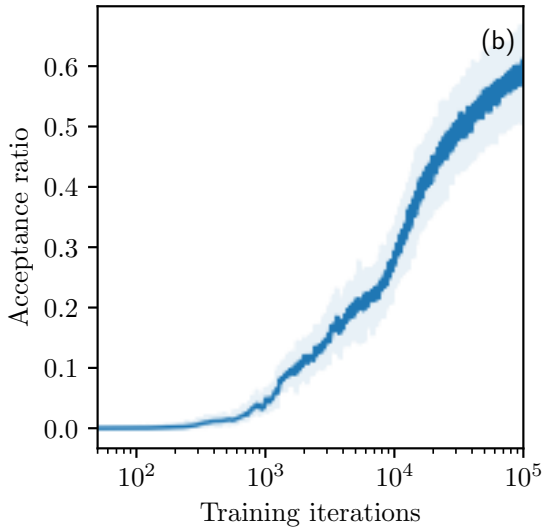
Base samples

$$\rho_{\mathrm{B}}(\Phi) \propto e^{-\beta U_{\mathrm{B}}(\Phi)}$$

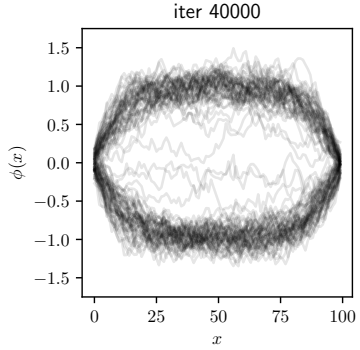
Gaussian white noise





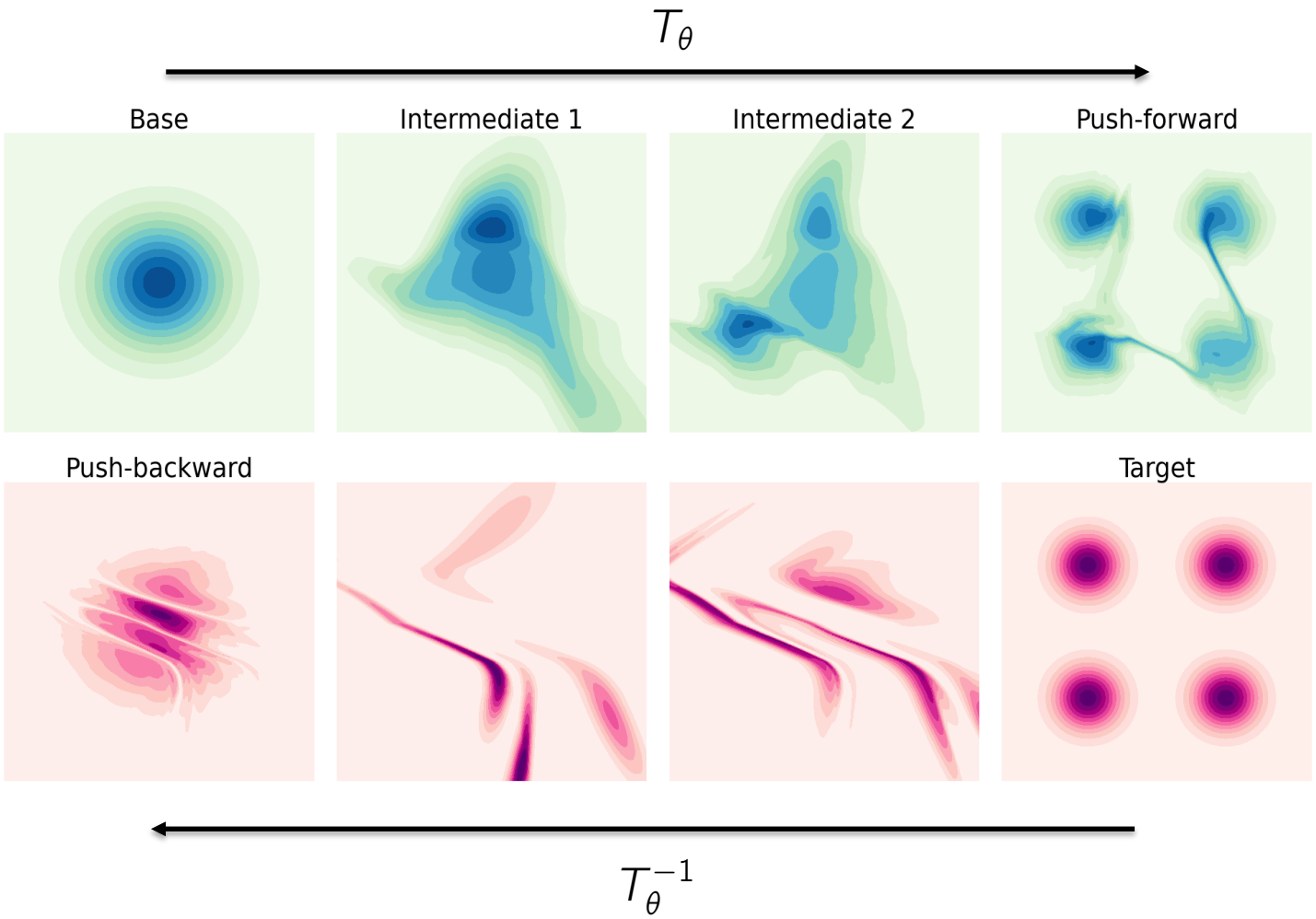


After training



[MG, Rotskoff & Vanden-Eijnden – *PNAS 2022*]
Cf Grant's talk

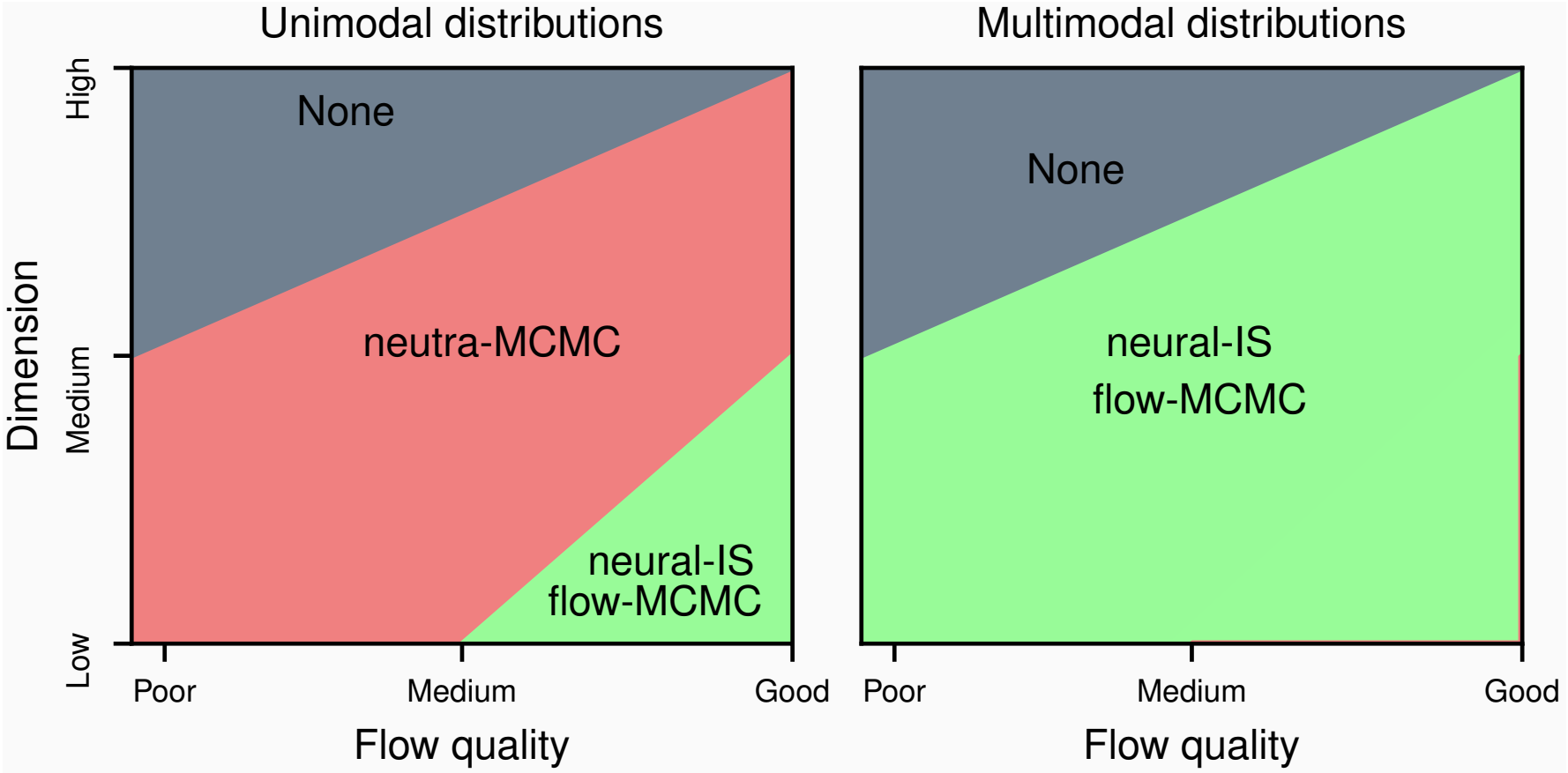
Yet learned transport does not erase mutlimodality



- Transporting a unimodal base to a multimodal target requires learning a very steep/almost flat transformation

Cornish et al. ICML 2020. Relaxing Bijectivity Constraints with Continuously Indexed Normalising Flows
Salmona et al. Neurips 2022. Can Push-forward Generative Models Fit Multimodal Distributions?

Insights from emprical study for $\rho_\theta(x) \approx \rho_*(x)$



[Grenioux, Durmus, Moulines & MG, *ICML 2023*]

Second Idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

▷ Adaptive MCMCs [Haario at al. *Bernoulli* 2001, Jasra et al Statistics and Computing, 2007, Andrieu et al. *Bernoulli* 2011, Sejdinovic et al *ICML* 2014 …]

▷ Algorithm: Metropolis-Hastings with **non-local** generative model proposal

Initialize: $x_0^i \quad i = 1 \cdots N$

Loop:

Loop over parallel chains: $i = 1 \cdots N$

Draw from generative model $x_{t+1}^i \sim \rho_\theta(x)$

Accept-reject $\mathrm{acc}(x_{t+1}^i | x_t^i) = \min\left[1, \frac{\boldsymbol{\rho}_*(x_{t+1}^i)\rho_\theta(x_t^i)}{\boldsymbol{\rho}_*(x_t^i)\rho_\theta(x_{t+1}^i)}\right]$

Local resampling $x_{t+1}^i \sim \pi_{\mathrm{local}}(x_{t+1}^i | x_t^i)$

Update NF parameters $\theta \leftarrow \theta + \eta \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log \rho_\theta(x_{t+1}^i)$

▷ Local + Mode jumping: [Tjelmeland & Hegstad *Scandinavian J. of Statistics* 2001, Sminchisescu & Welling *AISTAT* 2017, Pompe et al. *Ann. Stat* 2020, Sbailò et al. *J. Chem. Phys.* 2021, Tawn, Moores & Roberts 2021 …]

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

Second Idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

▷ Adaptive MCMCs [Haario at al. *Bernoulli* 2001, Jasra et al Statistics and Computing, 2007, Andrieu et al. *Bernoulli* 2011, Sejdinovic et al *ICML* 2014 …]

▷ Algorithm: Metropolis-Hastings with **non-local** generative model proposal

Initialize: $x_0^i \quad i = 1 \cdots N$

Loop:

Loop over parallel chains: $i = 1 \cdots N$

Metropolis-Hastings with NF

Draw from generative model $x_{t+1}^i \sim \rho_\theta(x)$

Accept-reject $\mathrm{acc}(x_{t+1}^i | x_t^i) = \min\left[1, \frac{\boldsymbol{\rho}_*(x_{t+1}^i)\rho_\theta(x_t^i)}{\boldsymbol{\rho}_*(x_t^i)\rho_\theta(x_{t+1}^i)}\right]$

Local resampling $x_{t+1}^i \sim \pi_{\mathrm{local}}(x_{t+1}^i | x_t^i)$

Update NF parameters $\theta \leftarrow \theta + \eta \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log \rho_\theta(x_{t+1}^i)$

▷ Local + Mode jumping: [Tjelmeland & Hegstad *Scandinavian J. of Statistics* 2001, Sminchisescu & Welling *AISTAT* 2017, Pompe et al. *Ann. Stat* 2020, Sbailò et al. *J. Chem. Phys.* 2021, Tawn, Moores & Roberts 2021 …]

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

Second Idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

▷ Adaptive MCMCs [Haario at al. *Bernoulli* 2001, Jasra et al Statistics and Computing, 2007, Andrieu et al. *Bernoulli* 2011, Sejdinovic et al *ICML* 2014 …]

▷ Algorithm: Metropolis-Hastings with **non-local** generative model proposal

Initialize: $x_0^i \quad i = 1 \cdots N$

Loop:

Loop over parallel chains: $i = 1 \cdots N$

Metropolis-Hastings with NF

Draw from generative model $x_{t+1}^i \sim \rho_\theta(x)$

Accept-reject $\mathrm{acc}(x_{t+1}^i | x_t^i) = \min\left[1, \frac{\boldsymbol{\rho}_*(x_{t+1}^i)\rho_\theta(x_t^i)}{\boldsymbol{\rho}_*(x_t^i)\rho_\theta(x_{t+1}^i)}\right]$

Local resampling $x_{t+1}^i \sim \pi_{\mathrm{local}}(x_{t+1}^i | x_t^i)$

Update NF parameters $\theta \leftarrow \theta + \eta \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log \rho_\theta(x_{t+1}^i)$

Maximum likelihood GD

▷ Local + Mode jumping: [Tjelmeland & Hegstad *Scandinavian J. of Statistics* 2001, Sminchisescu & Welling *AISTAT* 2017, Pompe et al. *Ann. Stat* 2020, Sbailò et al. *J. Chem. Phys.* 2021, Tawn, Moores & Roberts 2021 …]

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

Second Idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

▷ Adaptive MCMCs [Haario at al. *Bernoulli* 2001, Jasra et al Statistics and Computing, 2007, Andrieu et al. *Bernoulli* 2011, Sejdinovic et al *ICML* 2014 …]

▷ Algorithm: Metropolis-Hastings with **non-local** generative model proposal

Initialize: $x_0^i \quad i = 1 \cdots N$

Loop:

Loop over parallel chains: $i = 1 \cdots N$

Metropolis-Hastings with NF

Draw from generative model $x_{t+1}^i \sim \rho_\theta(x)$

Accept-reject $\mathrm{acc}(x_{t+1}^i | x_t^i) = \min\left[1, \frac{\boldsymbol{\rho}_*(x_{t+1}^i)\rho_\theta(x_t^i)}{\boldsymbol{\rho}_*(x_t^i)\rho_\theta(x_{t+1}^i)}\right]$

Local resampling $x_{t+1}^i \sim \pi_{\mathrm{local}}(x_{t+1}^i | x_t^i)$

MCMC steps converging to the target

Update NF parameters $\theta \leftarrow \theta + \eta \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log \rho_\theta(x_{t+1}^i)$

Maximum likelihood GD

▷ Local + Mode jumping: [Tjelmeland & Hegstad *Scandinavian J. of Statistics* 2001, Sminchisescu & Welling *AISTAT* 2017, Pompe et al. *Ann. Stat* 2020, Sbailò et al. *J. Chem. Phys.* 2021, Tawn, Moores & Roberts 2021 …]

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

Second Idea: Adaptive Markov Chain Monte Carlo: simultaneous convergence of training & sampling

▷ Adaptive MCMCs [Haario at al. *Bernoulli* 2001, Jasra et al Statistics and Computing, 2007, Andrieu et al. *Bernoulli* 2011, Sejdinovic et al *ICML* 2014 …]

▷ Algorithm: Metropolis-Hastings with **non-local** generative model proposal

Initialize: $x_0^i \quad i = 1 \cdots N$

Loop:

Loop over parallel chains: $i = 1 \cdots N$

Metropolis-Hastings with NF

Draw from generative model $x_{t+1}^i \sim \rho_\theta(x)$

Accept-reject $\mathrm{acc}(x_{t+1}^i | x_t^i) = \min\left[1, \frac{\boldsymbol{\rho}_*(x_{t+1}^i)\rho_\theta(x_t^i)}{\boldsymbol{\rho}_*(x_t^i)\rho_\theta(x_{t+1}^i)}\right]$

Local resampling $x_{t+1}^i \sim \pi_{\text{local}}(x_{t+1}^i | x_t^i)$

MCMC steps converging to the target

Update NF parameters $\theta \leftarrow \theta + \eta \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log \rho_\theta(x_{t+1}^i)$

Maximum likelihood GD

Improve surrogate as you go

▷ Local + Mode jumping: [Tjelmeland & Hegstad *Scandinavian J. of Statistics* 2001, Sminchisescu & Welling *AISTAT* 2017, Pompe et al. *Ann. Stat* 2020, Sbailò et al. *J. Chem. Phys.* 2021, Tawn, Moores & Roberts 2021 …]

[Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020; MG, Rotskoff, Vanden-Eijnden PNAS 2022]

Ergodicity theory of adaptive MCMC

▷ Theory for independent Metopolis-Hastings sampler:

- Independent proposal: $\pi_{\text{prop}}(x^{n+1}|x^n) = \rho_\theta(x^n)$
- Metropolis-Hastings Markov kernel:

$$\pi_{\theta^n}(y|x) = \text{acc}(y|x)\rho_{\theta^n}(y) + \left(1 - \int_\Omega \text{d}y' \text{acc}(y'|x)\rho_{\theta^n}(y')\right)\delta(y-x)$$

[Andrieu & Moulines 2006, Roberts & Rosenthal 2007, Brofos, Gabrié, Lederman & Brubacker 2022]

Ergodicity theory of adaptive MCMC

▷ Theory for independent Metopolis-Hastings sampler:

- Independent proposal: $\pi_{\text{prop}}(x^{n+1}|x^n) = \rho_\theta(x^n)$
- Metropolis-Hastings Markov kernel:

$$\pi_{\theta^n}(y|x) = \text{acc}(y|x)\rho_{\theta^n}(y) + \left(1 - \int_\Omega \text{d}y' \text{acc}(y'|x)\rho_{\theta^n}(y')\right)\delta(y-x)$$

Assumptions:

▷ The sequence of Markov kernels exhibits **diminishing adaptation** if

$$\lim_{n\to+\infty} \|\pi_{\theta^n}(\cdot) - \pi_{\theta^{n+1}}(\cdot)\|_{\text{TV}} = 0 \text{ in probability.}$$

- e.g.: probability to adapt goes to 0, or converging sequence of parameters

[Andrieu & Moulines 2006, Roberts & Rosenthal 2007, Brofos, Gabrié, Lederman & Brubacker 2022]

Ergodicity theory of adaptive MCMC

▷ Theory for independent Metopolis-Hastings sampler:

- Independent proposal: $\pi_{\text{prop}}(x^{n+1}|x^n) = \rho_\theta(x^n)$
- Metropolis-Hastings Markov kernel:

$$\pi_{\theta^n}(y|x) = \text{acc}(y|x)\rho_{\theta^n}(y) + \left(1 - \int_\Omega \text{d}y' \text{acc}(y'|x)\rho_{\theta^n}(y')\right)\delta(y-x)$$

Assumptions:

▷ The sequence of Markov kernels exhibits **diminishing adaptation** if

$$\lim_{n\to+\infty} \|\pi_{\theta^n}(\cdot) - \pi_{\theta^{n+1}}(\cdot)\|_{\text{TV}} = 0 \text{ in probability.}$$

- e.g.: probability to adapt goes to 0, or converging sequence of parameters

▷ The sequence of Markov kernels exhibits **containement** if:

For any δ , there exists $M(\delta) > 0$ such that

$$\Pr\left(\frac{\rho_*}{\rho_{\theta^n}} \leq M(\delta)\,,\ \forall x \in \mathcal{X}\right) \geq 1-\delta \quad \forall n \in \mathbb{N}$$

[Andrieu & Moulines 2006, Roberts & Rosenthal 2007, Brofos, Gabrié, Lederman & Brubacker 2022]

Ergodicity theory of adaptive MCMC

▷ Theory for independent Metopolis-Hastings sampler:

- Independent proposal: $\pi_{\text{prop}}(x^{n+1}|x^n) = \rho_\theta(x^n)$
- Metropolis-Hastings Markov kernel:

$$\pi_{\theta^n}(y|x) = \text{acc}(y|x)\rho_{\theta^n}(y) + \left(1 - \int_\Omega \text{d}y' \text{acc}(y'|x)\rho_{\theta^n}(y')\right)\delta(y-x)$$

Assumptions:

▷ The sequence of Markov kernels exhibits **diminishing adaptation** if

$$\lim_{n\to+\infty} \|\pi_{\theta^n}(\cdot) - \pi_{\theta^{n+1}}(\cdot)\|_{\text{TV}} = 0 \text{ in probability.}$$

- e.g.: probability to adapt goes to 0, or converging sequence of parameters

▷ The sequence of Markov kernels exhibits **containement** if:

For any δ , there exists $M(\delta) > 0$ such that

$$\Pr\left(\frac{\rho_*}{\rho_{\theta^n}} \leq M(\delta)\,,\ \forall x \in \mathcal{X}\right) \geq 1-\delta \quad \forall n \in \mathbb{N}$$

▷ Theorems: (Andrieu & Moulines 2006, Roberts & Rosenthal 2007):

If the sequence of Markov kernels exhibits diminishing **adaptation** and **containment**, the chain is ergodic for the distribution ρ^*.

[Andrieu & Moulines 2006, Roberts & Rosenthal 2007, Brofos, Gabrié, Lederman & Brubacker 2022]