Enhancing Sampling with Learning: MCMC, Generative Models and Overlaps

ML x Science Summer School

June 16 2022

Marylou Gabrié

(CMAP, École Polytechnique,

CCM Visiting Scholar, Flatiron Institute)

High-dimensional probabilistic models

▷ Statistical mechanics / Chemistry

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

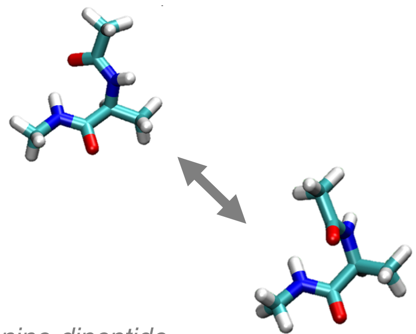
▷ Quantum mechanics (wave functions)

▷ Bayesian statistical modelling

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

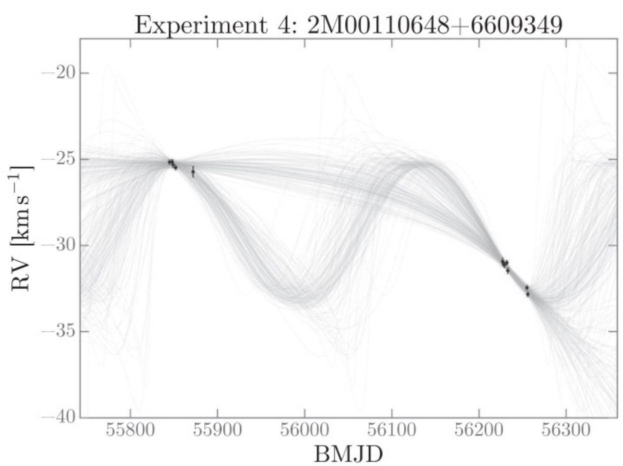
▷ Typically known up to normalization constant

ex: molecular configurations



Alanine-dipeptide
Jiang et al J. Phys. Chem. B 2019

ex: Astrophysics data modelling



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

Monte Carlo Methods

- ▷ Random variable $x \in \Omega \subset \mathbb{R}^D$, and density $\rho(x) = \frac{1}{\mathcal{Z}} e^{-U(x)}$ with unknown $\mathcal{Z}$

- ▷ Task: Compute expectations $\mathbb{E}_\rho[f(x)] = \int_\Omega f(x)\rho(x)\mathrm{d}x$

- ▷ Method: Monte Carlo approximations, generate $x_1, \ldots x_N, \ldots$

  such that $$\mathbb{E}_\rho[f(x)] = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

  In particular if $x_1, \ldots x_N, \ldots$ are i.i.d. draws from $\rho(x)$

- ▷ Monte Carlo **Markov Chains** idea to obtain samples:

  Design transition kernel $\pi(x_{t+1}|x_t)$ such that

  chain $x_0, x_1, \ldots, x_t$ = samples from $\rho(x) \propto e^{-U(x)}$ for t large enough

[e.g. Liu. *Monte Carlo Strategies in Scientific Computing,* 2004 – Brooks et al. *Handbook of MCMC,* 2011]

Outline for today

1.1 How to obtain samples? Markov Chain MC

- Idea: design transition kernel $\pi(x_{t+1}|x_t)$ such that chain $x_0, x_1, \ldots, x_t$ produces samples from ρ_* for t large enough

- Important example:

Metropolis-Hastings sampler

Initialize: x_0

Iterate:

- Propose $x_{t+1} \sim \rho_p(x_{t+1}|x_t)$

- Accept/Reject with prob.

$$\mathrm{acc}(x_{t+1}|x_t) = \min\left[1, \frac{\rho_*(x_{t+1})\rho_p(x_t|x_{t+1})}{\rho_*(x_t)\rho_p(x_{t+1}|x_t)}\right]$$

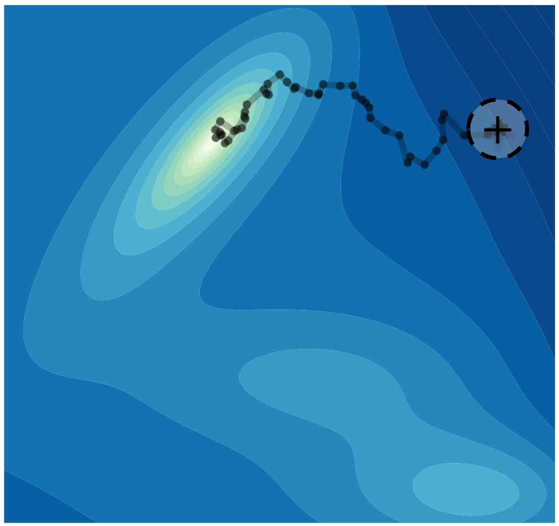
- If reject stay $x_{t+1} = x_t$

[e.g. Liu. *Monte Carlo Strategies in Scientific Computing,* 2004 – Brooks et al. *Handbook of MCMC,* 2011]

Examples of Metropolis-Hastings MCMC

- ▷ Gaussian random walk $\rho_p(x_{t+1}|x_t) = \mathcal{N}(x_t, \Sigma)$

e.g. 2d Müller-Brown potential
$\rho_*(x) = e^{-U_*(x)}/Z$



T = 100 steps

Metropolis-Hastings sampler

Initialize: x_0

Iterate:

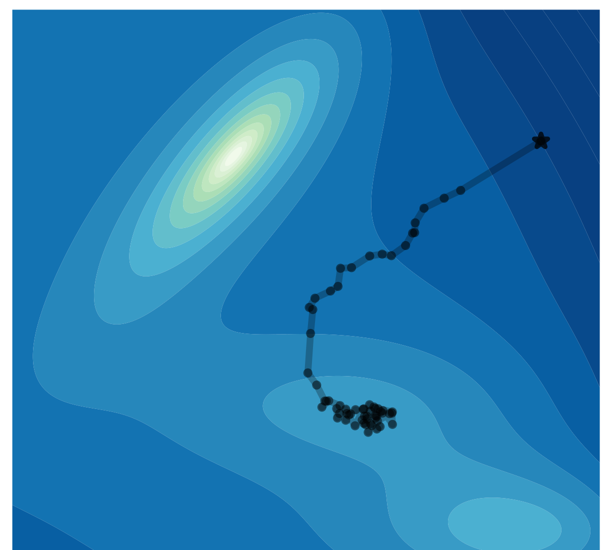
- Propose $x_{t+1} \sim \rho_p(x_{t+1}|x_t)$
- Accept/Reject with prob.

$$\mathrm{acc}(x_{t+1}|x_t) = \min\left[1, \frac{\rho_*(x_{t+1})\rho_p(x_t|x_{t+1})}{\rho_*(x_t)\rho_p(x_{t+1}|x_t)}\right]$$

- If reject stay $x_{t+1} = x_t$

- ▷ (Metropolis Adjusted) Langevin algorithm (MALA)

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



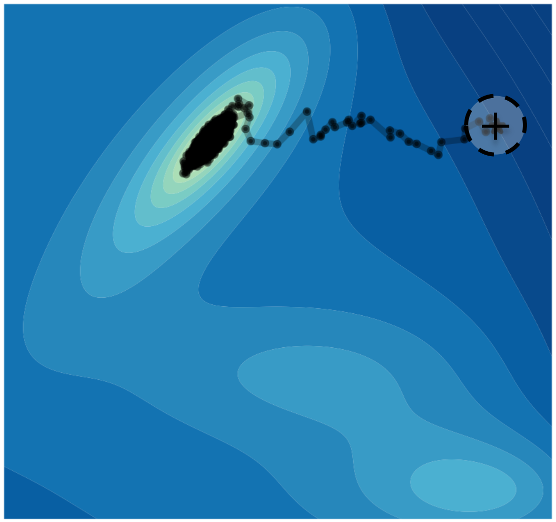
T = 100 steps

Challenge: Decorrelation and convergence

▷ Gaussian random walk $\rho_p(x_{t+1}|x_t) = \mathcal{N}(x_t, \Sigma)$

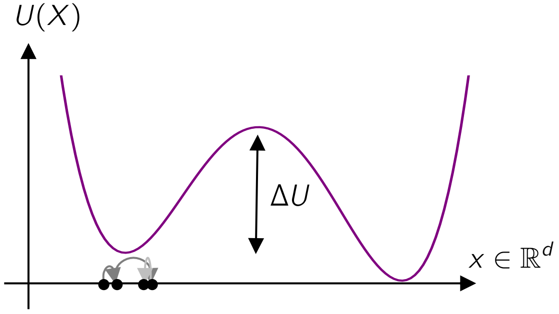
e.g. 2d Müller-Brown potential

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

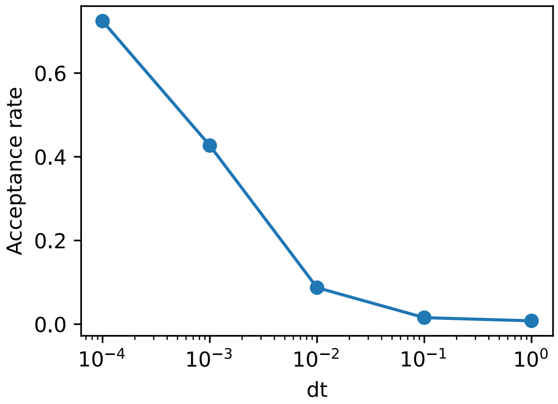


T = 10000 steps

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



▷ Trade-off size local moves / acceptance



▷ Many many proposition for faster “mixing”

- Use gradient information: Langevin dynamics, Hamiltonian MC
- Gradually approach the target: Sequential Monte Carlo, Annealed Importance Sampling

1.2. Variational Inference

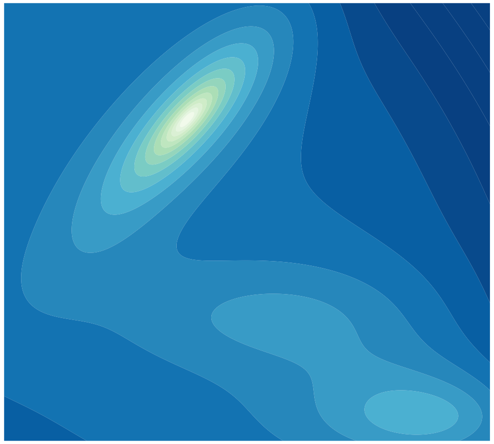
▷ Task: Compute expectations $\mathbb{E}_\rho[f(x)] = \int_\Omega f(x)\rho(x)\mathrm{d}x$

▷ Variational inference (original idea from statistical mechanics!)

- Optimize surrogate tractable distribution: minimize Kullback-Leibler divergence

$$D_{\mathrm{KL}}(\rho_\theta \| \rho_*) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_\theta(x)\mathrm{d}x \quad \Longrightarrow \quad L[\rho_\theta] = -\sum_{i=1}^{N} \log \frac{\rho_\theta(x_i)}{\rho_*(x_i)} \qquad x_i \sim \rho_\theta(x)$$

- e.g. Gaussian $\rho_\theta(x) = \mathcal{N}(x; \mu_\theta, \Sigma_\theta)$



▷ Issue: expressiveness of surrogate model? How to control the quality?

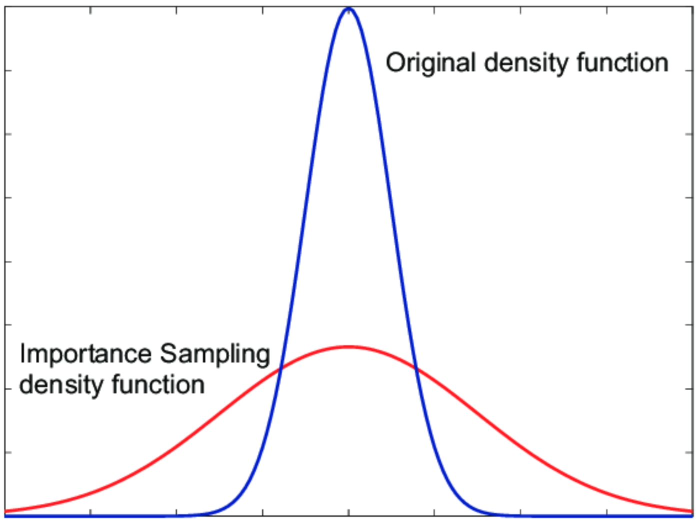
Weiss, P. (1907). L'hypothèse du champ moléculaire et la propriété ferromagnétique.
Wainwright, M. J., & Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference.

1.3 Importance Sampling

▷ Task: Compute expectations $\mathbb{E}_\rho[f(x)] = \int_\Omega f(x)\rho(x)\mathrm{d}x$

▷ Importance sampling

- Samples from proposal distribution $x_i \sim \rho_p(x_i)$
- Reweight $w_i = \frac{\rho_*(x_i)/\rho_p(x_i)}{\sum_{i=1}^N \rho_*(x_i)/\rho_p(x_i)}$
- Compute $\mathbb{E}_{\rho^*}[f(x)] \approx \frac{1}{N}\sum_{i=1}^N w_i f(x_i)$
- Issue: correspondence target/proposal?



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.

Outline for today

1. Inference and sampling: motivation and challenges

    1.1 - Metropolis-Hasting

    1.2 - Variational inference

    1.3 - Importance sampling

2. Unsupervised learning / generative models

    2.1 - Latent deep generative models

    2.2 - Normalizing flows

3. Combining traditional inference method and learning

    3.1 - Borrowing from Variational Inference & Importance sampling

    3.2 - Reparametrization

    3.2 - Adaptive algorithms

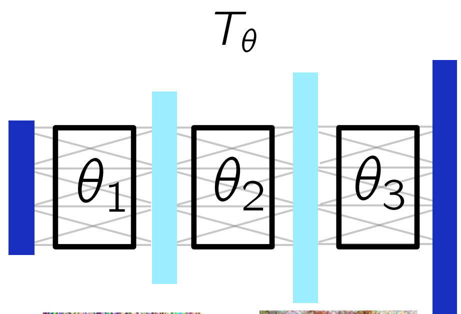
    3.3 - Incorporating more physics in models

2.1 Deep generative models

▷ Use transformation T_θ (deep neural network) from simple base distribution ρ_B :

[“GANs” Goodfellow et al. *NeurIPS* 2014,
“VAEs” Kingma & Welling *ICLR* 2014,
“Normalizing flows” Papamakarios et al. *JMLR* 2021]

$z \sim \rho_{\mathrm{B}}(z)$



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

Song et al. *ICLR* 2021

[“GANs” Goodfellow et al. *NeurIPS* 2014, “VAEs” Kingma & Welling *ICLR* 2014, “Normalizing flows” Papamakarios et al. *JMLR* 2021, “Score based diffusion models” Song et al. *ICLR 2021,* Tabak & V.-E. *Commun. Math. Sci.* 2010, Dinh et al *ICLR* 2017, Papamakarios et al *JMLR* 2021, Kingma et al *Neurips* 2018]

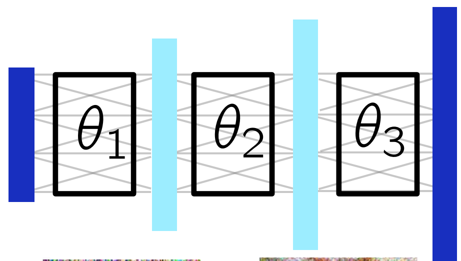
2.1 Deep generative models

▷ Use transformation T_θ (deep neural network) from simple base distribution ρ_B :

["GANs" Goodfellow et al. *NeurIPS* 2014,
"VAEs" Kingma & Welling *ICLR* 2014,
"Normalizing flows" Papamakarios et al. *JMLR* 2021]

T_θ

$z \sim \rho_{\mathrm{B}}(z)$



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

Song et al. *ICLR* 2021

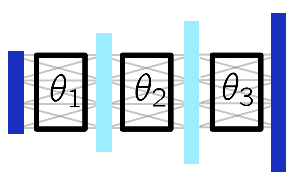
▷ Two main training methods of unsupervised learning:

- Maximum likelihood: $L[\rho_\theta] = -\sum_{i=1}^{N} \log \rho_\theta(x_i)$ with x_i data samples

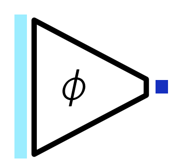
+ SGD!

- Adversarial training: $\min_\theta \max_\phi \left[\mathbb{E}_{\rho_D} [\ln D_\phi(x)] + \mathbb{E}_{\rho_{\mathrm{B}}} [\ln(1 - D_\phi(T_\theta(z)))] \right]$ with ρ_D data distribution

$z \sim \rho_{\mathrm{B}}(z)$



$x \in \mathbb{R}^D$



$D_\phi(x) \in [0, 1]$

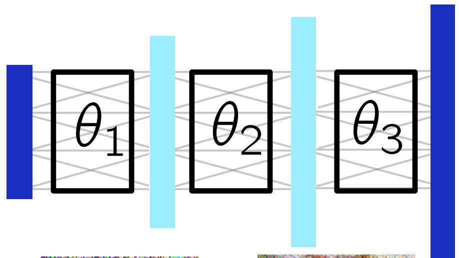
["GANs" Goodfellow et al. *NeurIPS* 2014, "VAEs" Kingma & Welling *ICLR* 2014, "Normalizing flows" Papamakarios et al. *JMLR* 2021, "Score based diffusion models" Song et al. *ICLR 2021,* Tabak & V.-E. *Commun. Math. Sci.* 2010, Dinh et al *ICLR* 2017*,* Papamakarios et al *JMLR* 2021, Kingma et al *Neurips* 2018]

2.1 Deep generative models

▷ Use transformation T_θ (deep neural network) from simple base distribution ρ_B :

T_θ

$z \sim \rho_B(z)$



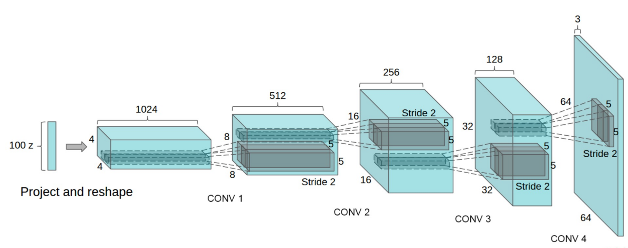
$x = T_\theta(z) \sim \rho_\theta(x)$ "push-forward" distribution

[“GANs” Goodfellow et al. *NeurIPS* 2014,
“VAEs” Kingma & Welling *ICLR* 2014,
“Normalizing flows” Papamakarios et al. *JMLR* 2021]

Song et al. *ICLR* 2021

▷ Two main training methods of unsupervised learning:

- Maximum likelihood: $L[\rho_\theta] = -\sum_{i=1}^{N} \log \rho_\theta(x_i)$ with x_i data samples
- Adversarial training: $\min_\theta \max_\phi \left[\mathbb{E}_{\rho_D} [\ln D_\phi(x)] + \mathbb{E}_{\rho_B} [\ln(1 - D_\phi(T_\theta(z)))] \right]$ with ρ_D data distribution

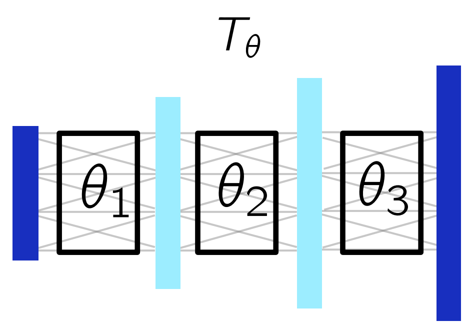


[Radford et al ICLR 2016; Karras et al CVPR 2019]

Nota Bene: Intractability of the push-forward of many latent generative models

▷ In general latent dimension much smaller than data dimension

$z \sim \rho_{\mathrm{B}}(z)$



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

"push-forward" distribution

- Push-forward computation involves marginalization …

$$\rho_\theta(x)\mathrm{d}x = \int_{\mathbb{R}^d} \mathrm{d}z \, \rho_B(z) \, \delta(T_\theta(z) - x)$$

▷ Hence difficult to do maximum likelihood:
e.g. optimize ELBLO (evidence lower bound in VAE)

"VAEs" Kingma & Welling *ICLR* 2014]

2.2 A special type of Deep Generative Models

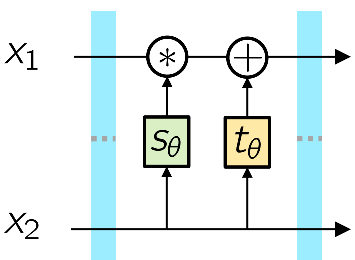
Normalizing Flows (NF): Invertible networks

- ▷ Parametrized invertible map $T_\theta : \Omega \mapsto \Omega \qquad \Omega \subset \mathbb{R}^d$
  - Base distribution $z \sim \rho_B(z)$
  - Push-forward distribution $x = T_\theta(z) \quad \sim \rho_\theta(x) = \rho_B(T_\theta^{-1}(x)) \det \left| \nabla_x T_\theta^{-1} \right|$

Most generative model are not invertible!
Intractable push-forward.

- ▷ e.g. "Coupling layers": easy-to-compute inverse and Jacobian

Affine coupling layer $T_\theta(x)$

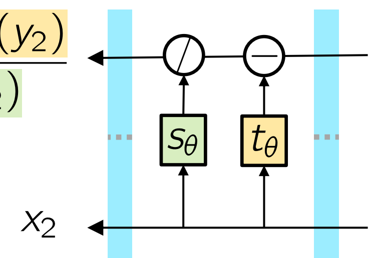


$y_1 = s_\theta(x_2) * x_1 + t_\theta(x_2)$

$y_2 = x_2$

Inverse layer $T_\theta^{-1}(y)$

$x_1 = \dfrac{y_1 - t_\theta(y_2)}{s_\theta(y_2)}$

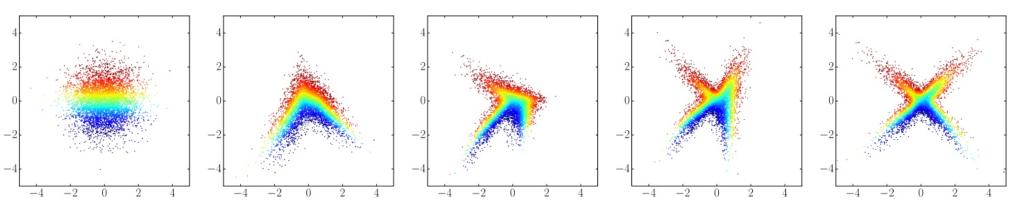


$y_2 = x_2$

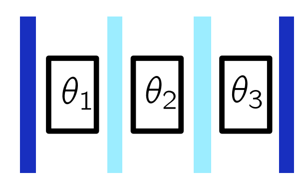
Block diagonal Jacobian: $\nabla_x T_\theta(x) = \begin{bmatrix} s_\theta(x_2) I_{d/2} & 0 \\ 0 & I_{d/2} \end{bmatrix}$

Easy to **sample** and easy to **evaluate density**

- ▷ Composition to encode for sophisticated transformations



$T_\theta = T_{\theta_3} \circ T_{\theta_2} \circ T_{\theta_1}$



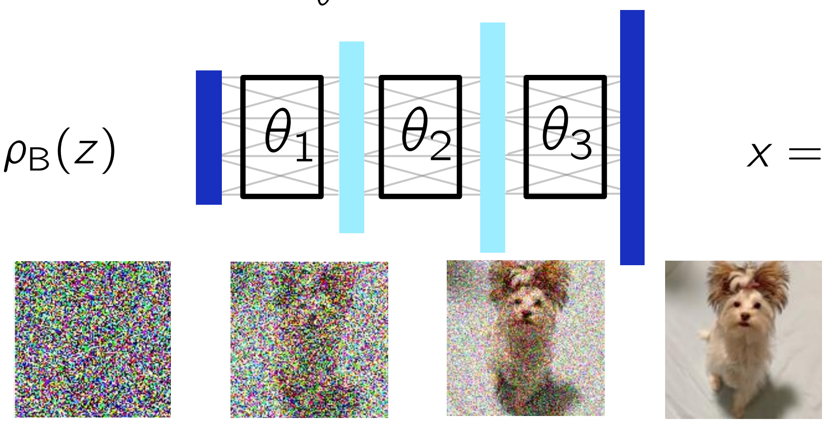
[Tabak & Vanden Eijnden *Commun. Math. Sci.* 2010, Dinh, L. et al *ICLR* 2017, Papamakarios, G et al *JMLR* 2021]

Deep generative models for sampling target $\rho_*(x)$

▷ Parametric model: Simple base random variable transformed by a deep neural network T_θ

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

$z \sim \rho_{\mathrm{B}}(z)$



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

New dog picture for each new base variable!

Song et al. *ICLR* 2021

▷ Sample complicated $\rho_*(x)$ by modelling it with deep generative model? Well …

- Need to learn T_θ for which we need data - $x_i \sim \rho_*(x)$ - do we?
- Even with data $x_i \sim \rho_*(x)$ to learn, unlikely to learn perfect model $\rho_\theta(x) = \rho_*(x)$, right?

["GANs" Goodfellow et al. *NeurIPS* 2014, "VAEs" Kingma & Welling *ICLR* 2014,
"Normalizing flows" Papamakarios et al. *JMLR* 2021,
"Score based diffusion models" Song et al. *ICLR 2021*]

Outline for today

3.1 Training NF with variational principle

▷ No need for data?

- minimize Kullback-Leibler $D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$ = variational principle with expressive $\rho_\theta(x)$ ansatz

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

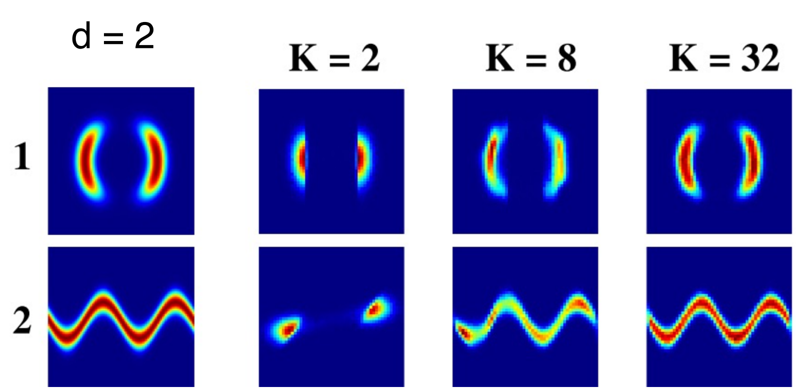
easy to obtain!

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

explicit!

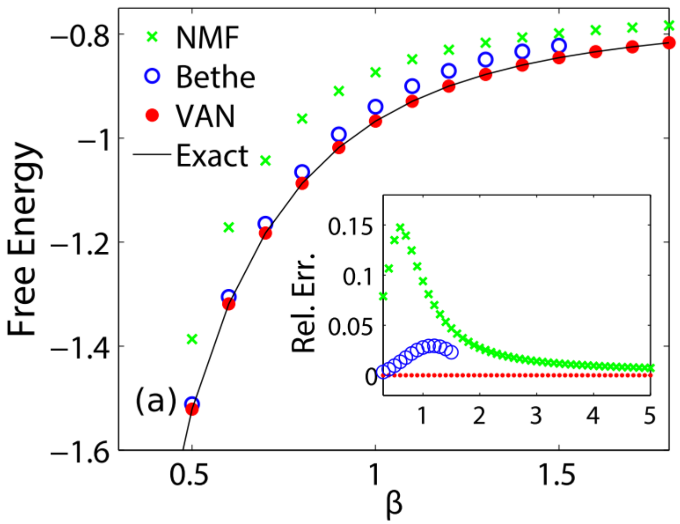
▷ First results: quality as a function of expressivity

layers



How to control the quality of surrogate model?

Spin system with random couplings d = 20



Requires annealing of target distribution!

Rezende & Mohamed, (2015). Variational inference with normalizing flows
Wu et al. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks.

3.1 Training NF with variational principle

▷ No need for data?

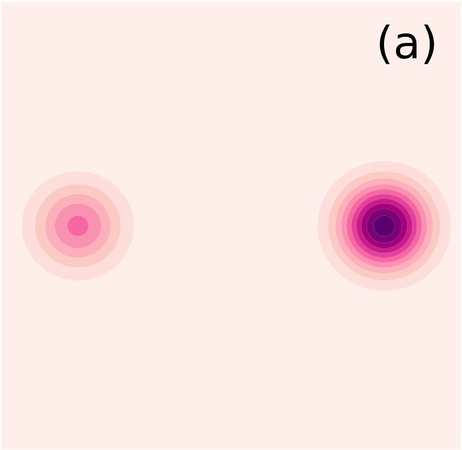
- minimize Kullback-Leibler $D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$ = variational principle with expressive $\rho_\theta(x)$ ansatz

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

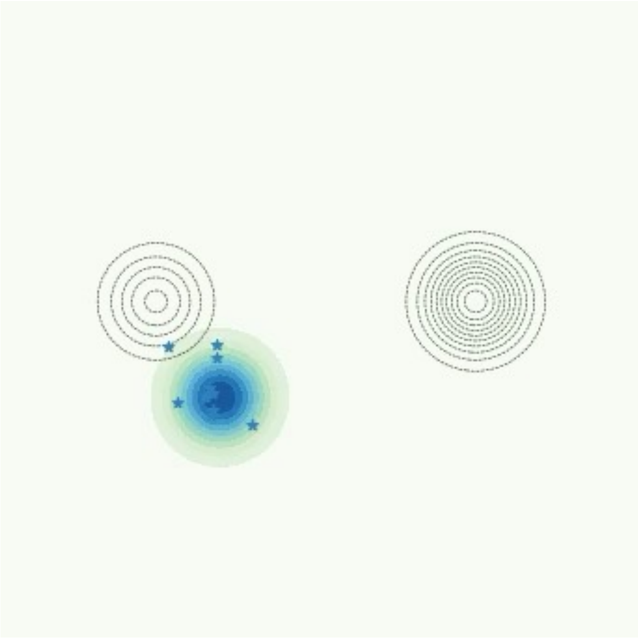
$$\rho_\theta(x) = \rho_{\mathrm{B}}(T_\theta^{-1}(x)) \det \left| \nabla_x T_\theta^{-1} \right|$$ explicit!

▷ Annealing of target? Why?

example:
$\rho_*(x)$ *mixture of 2 Gaussians (2d)*



$$\rho_{\theta_t}(x) = \rho_{\mathrm{B}}(T_{\theta_t}^{-1}(x)) \det \left| \nabla_x T_{\theta_t}^{-1} \right|$$



prone to mode collapse !

Rezende & Mohamed, (2015). Variational inference with normalizing flows
Wu et al. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks.

3.2 VI + max likelihood + importance sampling

"Boltzmann generator" Noé et al. (Science 2019)

▷ Training scheme

- Parametrized push-forward

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

- Minimize combined loss $L_{\mathrm{VI}}[\rho_\theta] + L_{\mathrm{data}}[\rho_\theta]$

$$L_{\mathrm{VI}}[\rho_\theta] = -\sum_{i=1}^{N} \log \frac{\rho_\theta(x_i)}{\rho_*(x_i)} \qquad x_i \sim \rho_\theta(x)$$

$$L_{\mathrm{data}}[\rho_\theta] = -\sum_{i=1}^{N} \log \rho_\theta(x_{d,i}) \qquad x_{d,i} \text{ small data set (from MD)}$$

▷ Importance sampling

- Sample from flow

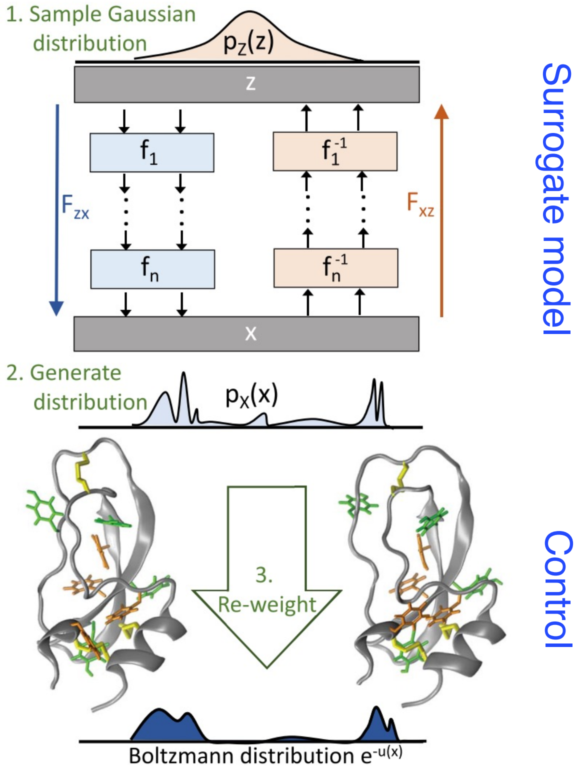
$$x_i \sim \rho_\theta(x)$$

- Compute importance weights

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

- Estimate

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



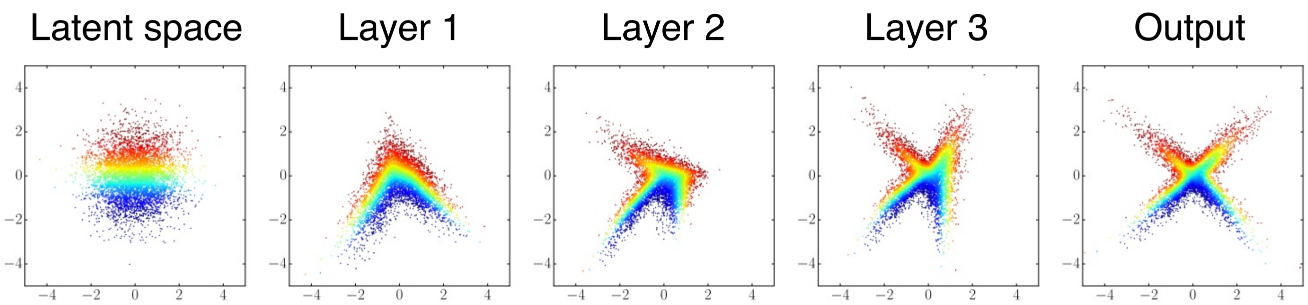
e.g. BPTI protein (58 amino acids)

data/model – chicken and egg problem

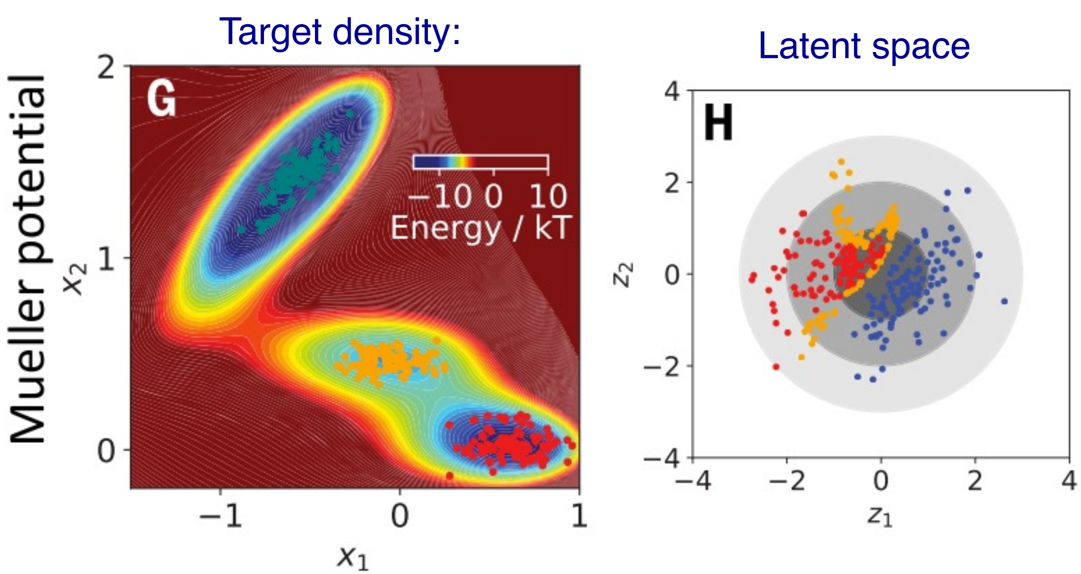
Noé et al (2019). Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. Science,

3.3 Reparametrization: reverse NF for MCMC

- ▷ Reverse transformation is normalizing = “Gaussianizing”



- ▷ Idea: train normalizing flow and use latent space to run traditional MCMC



Noé, F et al (2019). Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. Science,
Hoffman et al. (2019). NeuTra-lizing Bad Geometry in Hamiltonian Monte Carlo Using Neural Transport.

3.3 Reparametrization: reverse NF for MCMC

▷ NeuTra-lizing Bad Geometry in Hamiltonian Monte Carlo Using Neural Transport. (Hoffman et al 2019)

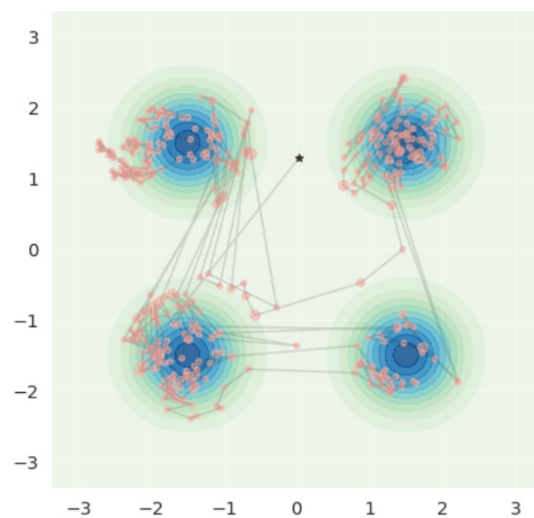
Target density: (a)

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

MALA (purely local):

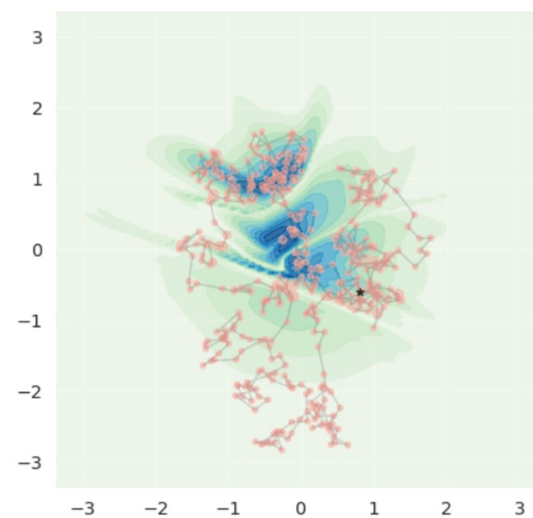
Target density:

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



Push backward:

$\rho_*(T_\theta^{-1}(z)) \det|\nabla_z T_\theta| \approx \rho_B(z)$



Louis Grenioux

Noé, F et al (2019). Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. Science,
Hoffman et al. (2019). NeuTra-lizing Bad Geometry in Hamiltonian Monte Carlo Using Neural Transport.

3.4 Adaptive MCMC with normalizing flow

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

Generative model parametrized density: $\rho_\theta(x)$

▷ Algorithm: Metropolis-Hastings with 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 $\mathrm{acc}(x_{t+1}^i | x_t^i) = \min\left[1, \frac{\rho_*(x_{t+1}^i)\rho_\theta(x_t^i)}{\rho_*(x_t^i)\rho_\theta(x_{t+1}^i)}\right]$
- Local resampling $x_{t+1}^i \sim \pi_{\mathrm{local}}(x_{t+1}^i | x_t^i)$

Metropolis-Hastings with NF

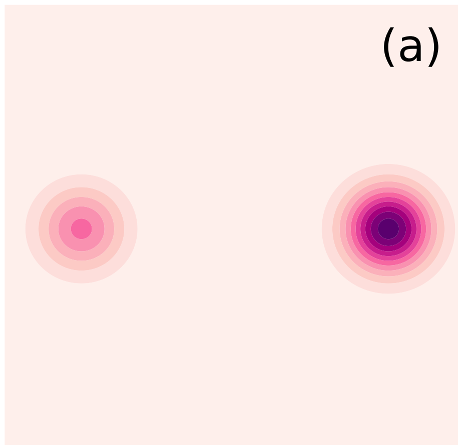
- Update NF paramters $\theta \leftarrow \theta + \eta \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log \rho_\theta(x_{t+1}^i)$

Maximum likelihood GD

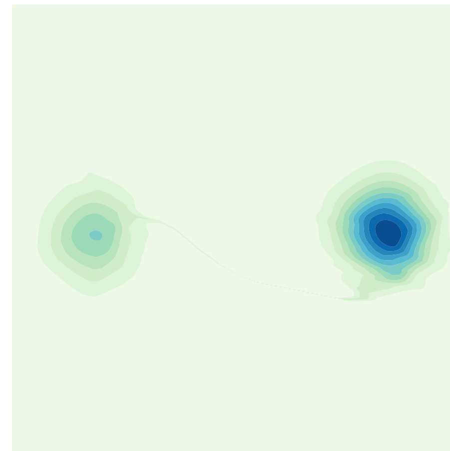
[Parno & Marzouk 2018, Gabrié, Rotskoff, Vanden-Eijden, *PNAS* 2022]

3.4 Adaptive MCMC – 2d Mixture of two Gaussians

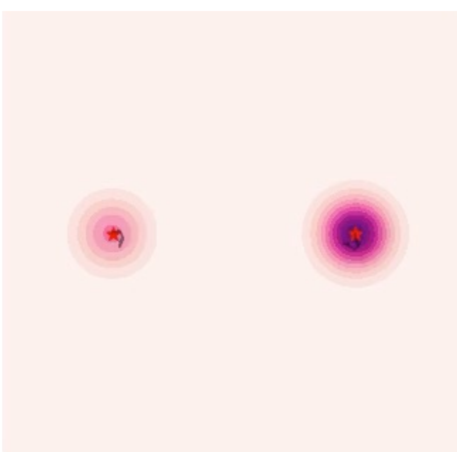
Target density:



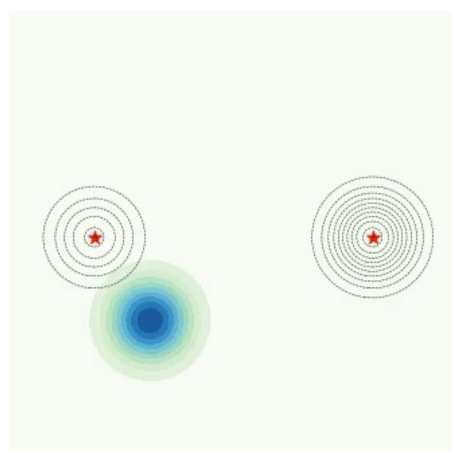
Final learned density:



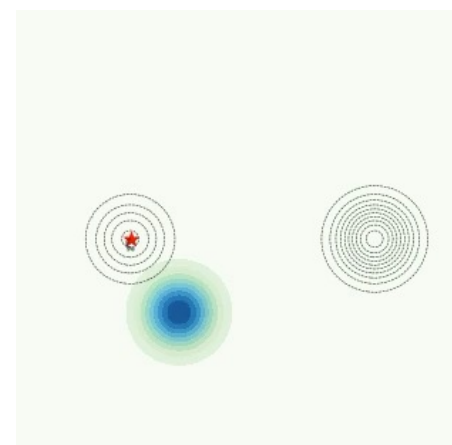
Local method only:



Concurrent:
careful intialization

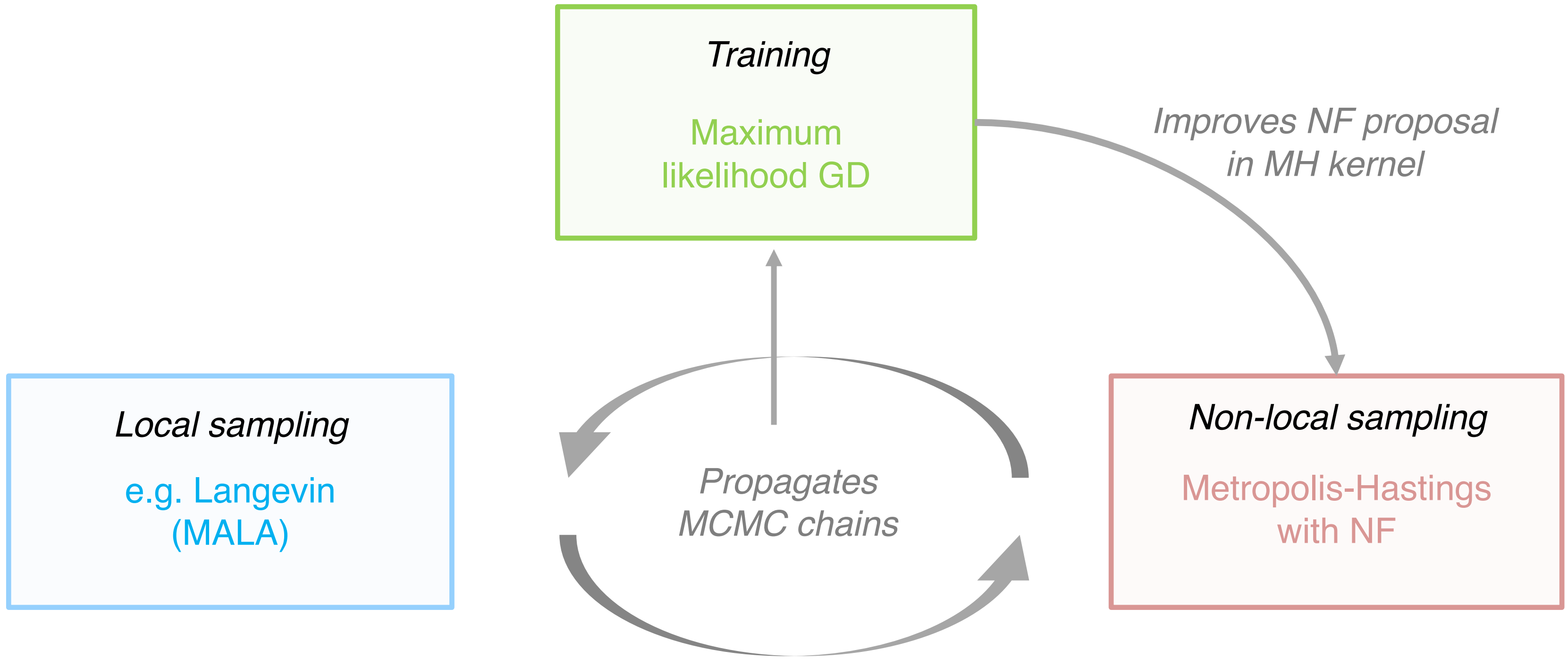


Concurrent:
starting with one walker



No mode discovery!

3.4 Adaptive MCMC with normalizing flow



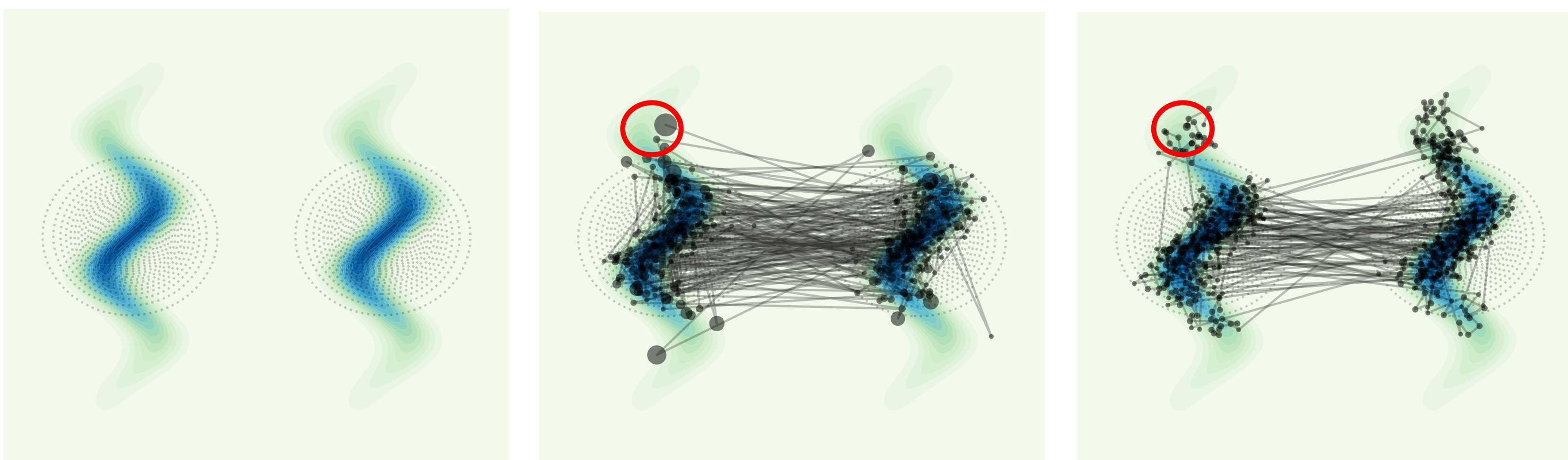
- Adaptive / "non-linear" Monte Carlo [Haario at al Bernoulli 2001, Jasra et al Statistics and Computing, 2007, Andrieu et al Bernoulli 2011, Sejdinovic et al ICML 2014, Parno & Marzouk 2018, Naesseth et al. Neurips 2020, Gabrié et al. PNAS 2022, …]

- Local + Mode jumping methods [Sminchisescu & Welling AISTAT 2017, Pompe et al. Ann. Stat 2020, Sbailò et al. J. Chem. Phys. 2021, ,…]

Why keep a local kernel on top of adaptive MCMC?

- ▷ In general tails of the distribution will be learned poorly

Global only　　　　Local + global



- Exploration – Explotation compromise
- Compensate for mismatch proposal/target

1) We cannot learn it all
2) Traditional local kernels still of great help!

[Samsonov et al. (ariXv2206.????)]

Outline for today

1. Inference and sampling: motivation and challenges

   1.1 - Metropolis-Hasting

   1.2 - Variational inference

   1.3 - Importance sampling

2. Unsupervised learning / generative models

   2.1 - Latent deep generative models

   2.2 - Normalizing flows

3. Combining traditional inference method and learning

   3.1 - Borrowing from Variational Inference & Importance sampling

   3.2 - Reparametrization

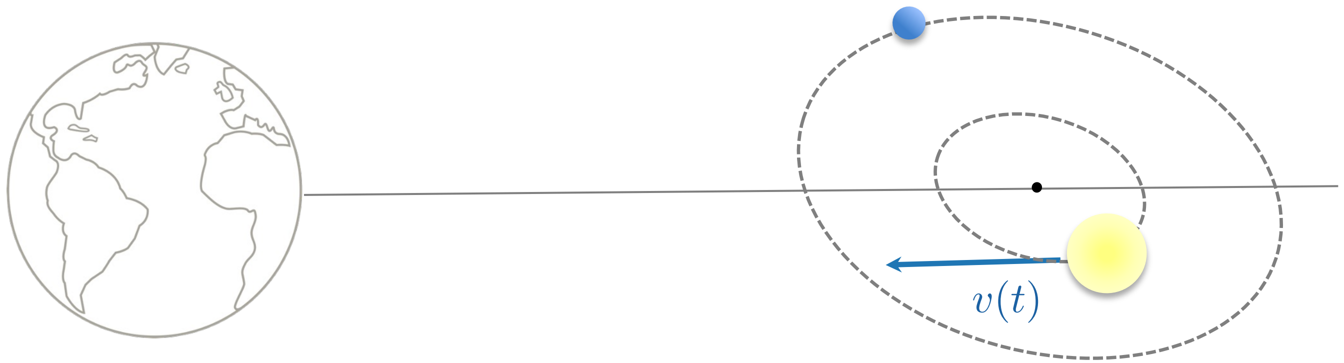
   3.2 - Adaptive algorithms

   3.3 - Incorporating more physics in models

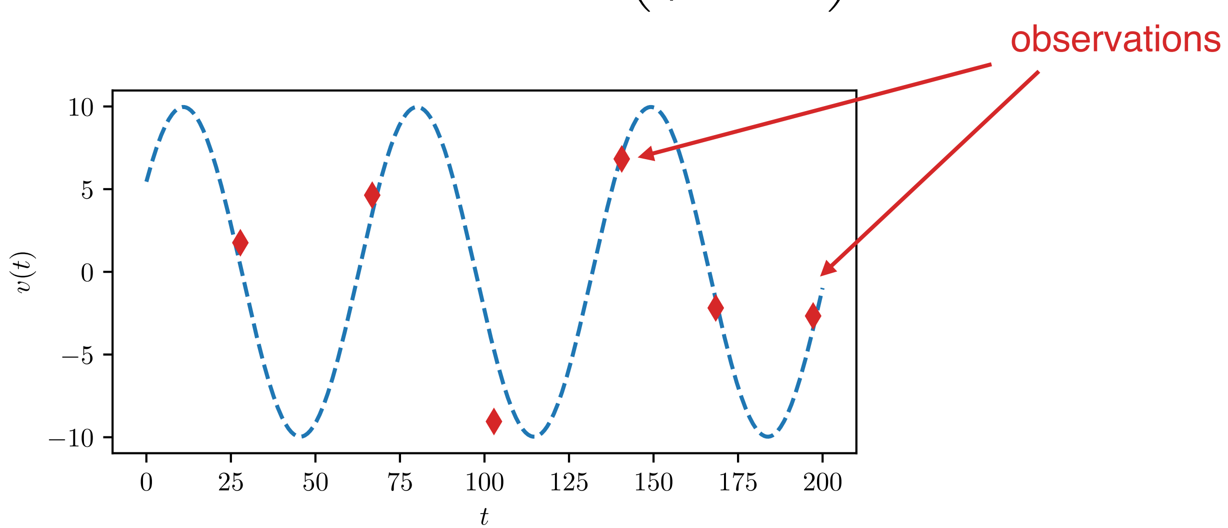
EXAMPLES

Bayesian inference:
An example of model selection from astrophysics

▷ Star-exoplanet system orbiting center of mass



▷ Radial velocity along the orbit $v(t;x) = v_0 + K \cos\left(\frac{2\pi}{P} t + \phi_0\right)$



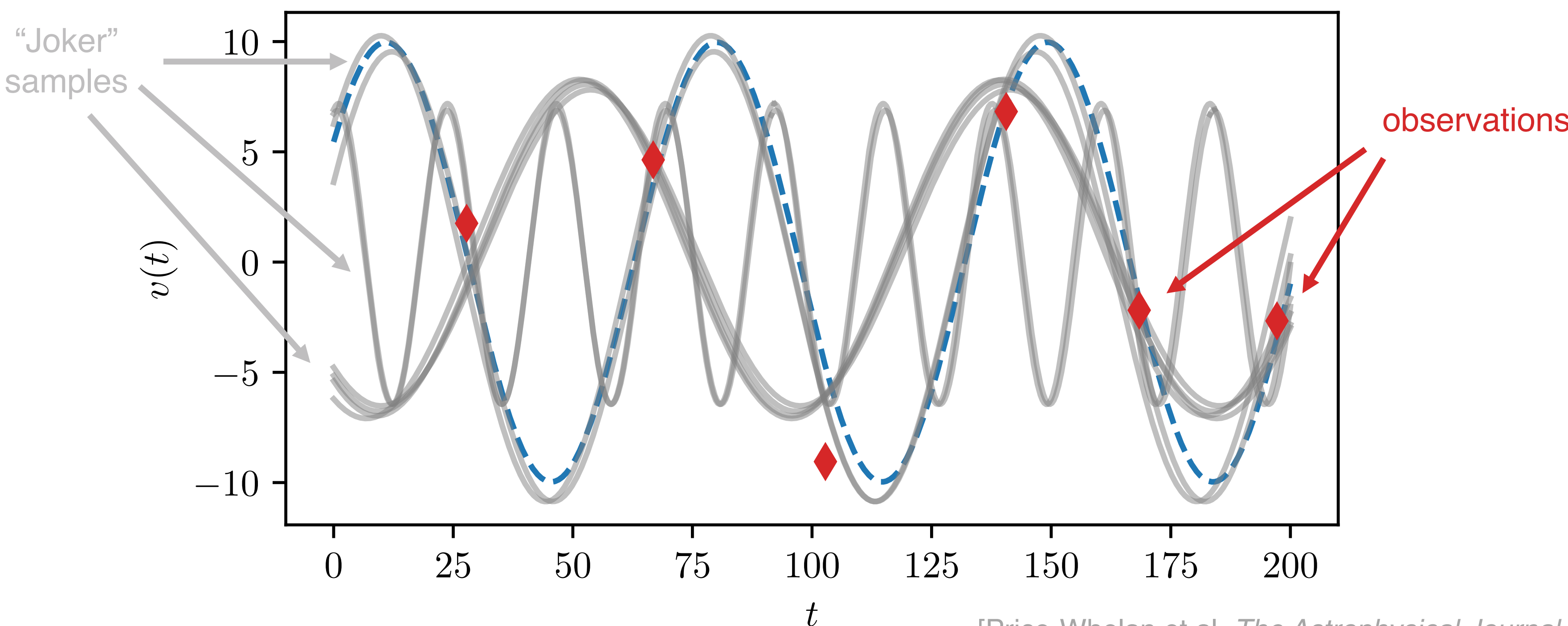
Bayesian model for velocity parameters

▷ Radial velocity $v(t;x) = v_0 + K\cos\left(\frac{2\pi}{P}t + \phi_0\right)$

▷ Parameters $x = (v_0, K, \phi_0, \ln P) \in \Omega \subset \mathrm{R}^4$

▷ Priors
$$\ln P \sim \mathcal{U}(\ln P_{\min}, \ln P_{\max}),$$
$$\phi_0 \sim \mathcal{U}(0, 2\pi),$$
$$K \sim \mathcal{N}(\mu_K, \sigma_K^2),$$
$$v_0 \sim \mathcal{N}(0, \sigma_{v_0}^2).$$

▷ Likelihood from observations $L(x) = \mathcal{N}(v_k; v(t_k; x), \sigma_{\mathrm{obs}}^2)$



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

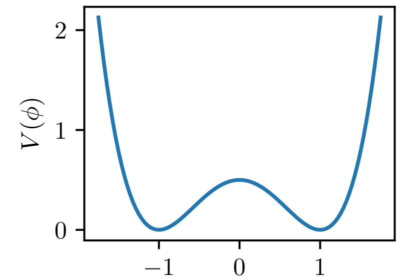
High-dimensional field models

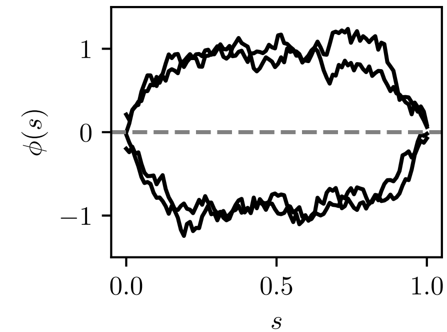
▷ Examples: Φ^4 model

- Random field $\phi\colon [0,1] \mapsto \mathbb{R} \in C([0,1];\mathbb{R})$
- Energy functional $U_*(\phi) = \int_{[0,1]} \left(\frac{a}{2}|\nabla_s \phi|^2 + V(\phi)\right) \mathrm{d}s$
- Local potential $V(\phi) = \frac{1}{2}(\phi^2 - 1)^2$
- Dirichlet boundary conditions $\phi(0) = 0, \phi(1) = 0$
- Target distribution $\rho(\phi) = \frac{1}{\mathcal{Z}_\beta} e^{-\beta U(\phi)}$

local potential

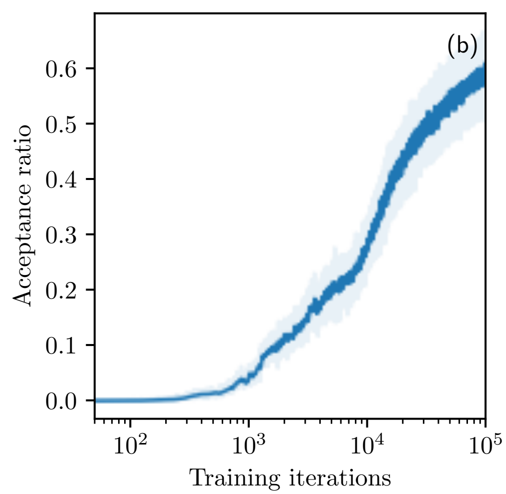
coupling term



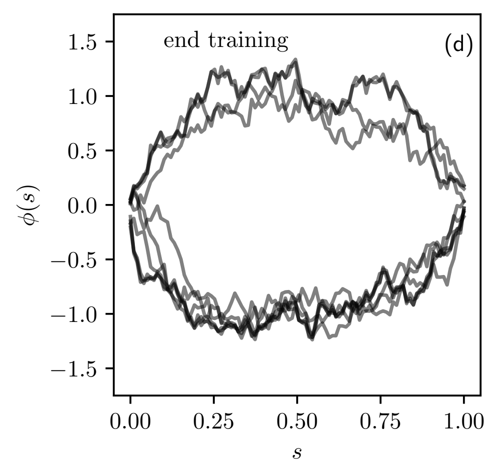


▷ Discretized: N=100

Acceptance ~ 60%



Fast mixing



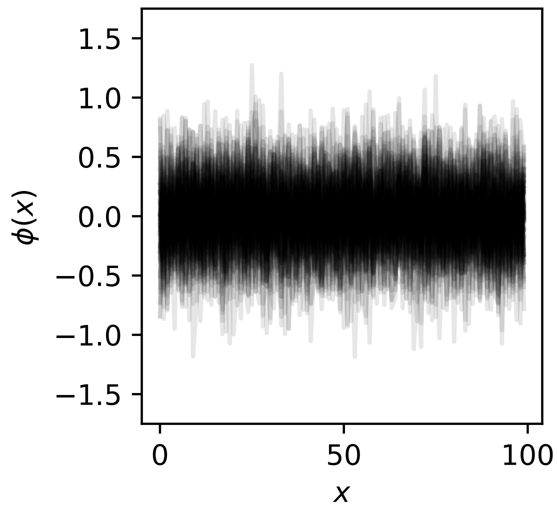
[Gabrié, Rotskoff & Vanden-Eijnden – *PNAS 2022*]

Uncoupled vs coupled base distributions

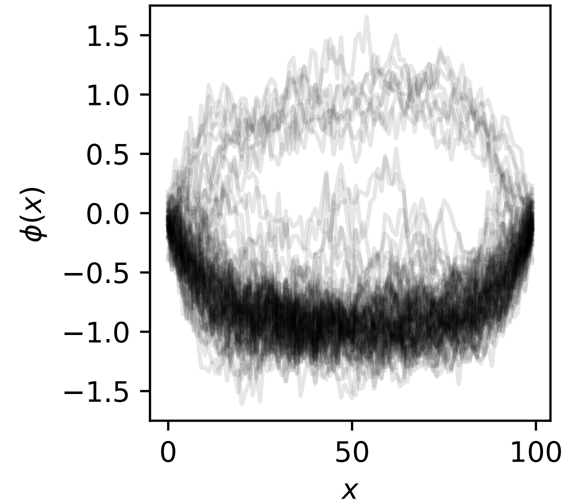
Gaussian uninformed (uncoupled)

$$U_{\mathrm{B}}(\phi) = \int \frac{1}{2\sigma^2}\phi^2 \mathrm{d}x$$

Base samples



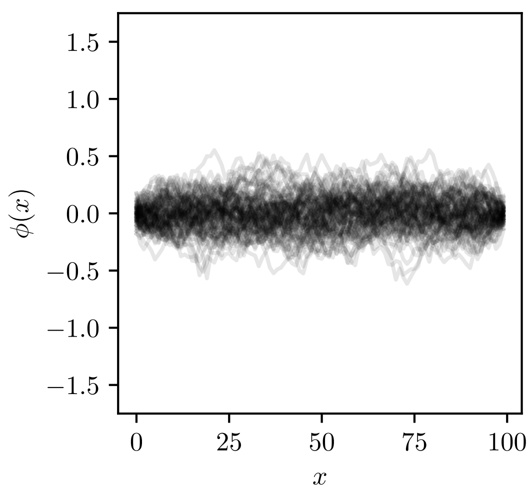
After training



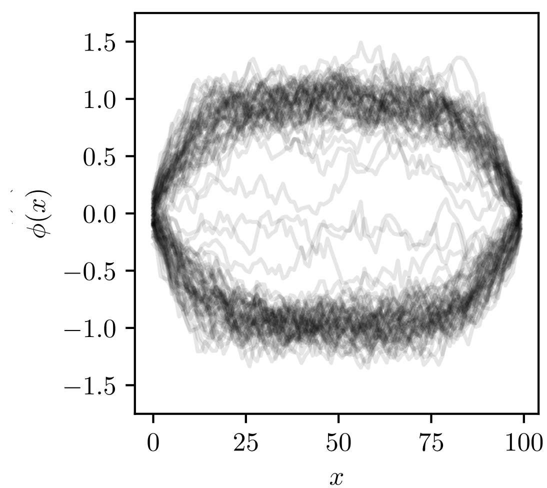
Gaussian informed (coupled)

$$U_{\mathrm{B}}(\phi) = \int \left(\frac{a}{2}|\nabla_x \phi|^2 + \frac{1}{2\sigma^2}\phi^2\right) \mathrm{d}x$$

Base samples



After training



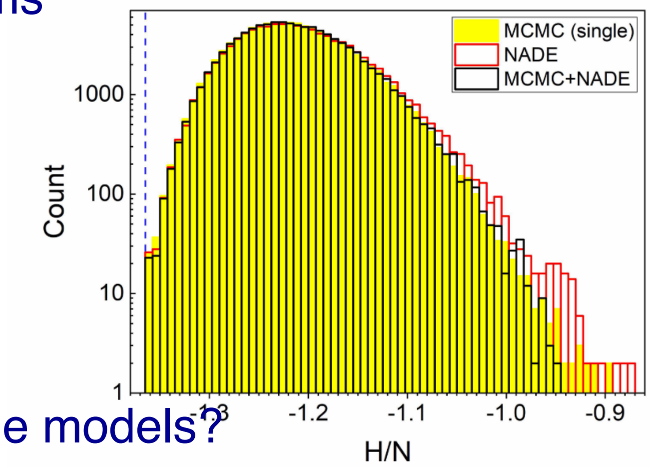
[Gabrié, Rotskoff & Vanden-Eijnden – *PNAS 2022*]

Pushing towards more complicated models

- Disordered systems = highly multimodal systems
  - Some promising results using annealing / Sequential Monte Carlo

2d – *Edwards Anderson*

S. Pilati *PRE* 2020



- Can surrogate probabilistic models scale to large models?
  - Metropolis acceptance

$$\mathrm{acc}(x_{t+1}|x_t) = \min\left[1, e^{-(\Delta U_* - \Delta U_\theta)}\right]$$

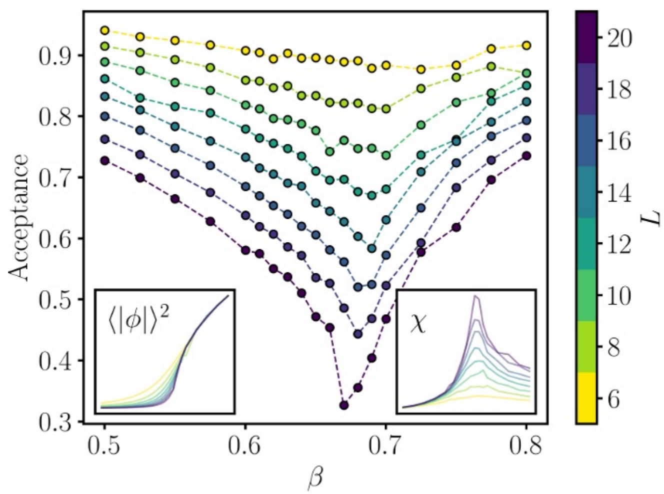
$$\Delta U_* = U_*(x_{t+1}) - U_*(x_t)$$

$$\Delta U_\theta = -\log \rho_\theta(x_{t+1}) + \log \rho_\theta(x_t)$$

  - Also in importance weights

2d - *Φ*[4] model

Del Debbio et al *PRD* 2021



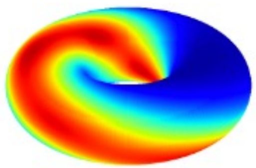
Efficient Modelling of Trivializing Maps for Lattice ϕ4 Theory Using Normalizing Flows- Del Debbio et al *PRD* 2021
Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks- B. McNaughton .., S. Pilati *PRE* 2020

Take aways

▷ Opportunities

- VI, IS and MCMC can be powered by normalizing flows
- Substantial speed up gains for

▷ Challenges for scaling things up



- Blending domain knowledge and learning is key!
  e.g. Rezende et al (2020). Normalizing flows on tori and spheres.
- Research direction: physically conditioned models
- Research direction: dimensionality reduction? separation of scales?

▷ Softwares

- Pytorch marylou-gabrie / **flonaco** Public
- Jax kazewong / **NFSampler** Public