Assisting sampling (of physical states) with generative models

MCMC, Generative Models and Overlaps

ASC Summer School: Physics meet AI - September 12-16, 2022

Marylou Gabrié (CMAP, École Polytechnique, Paris)

Manipulating high-dimensional probabilistic models: motivations

- ▷ Statistical mechanics / Chemistry

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

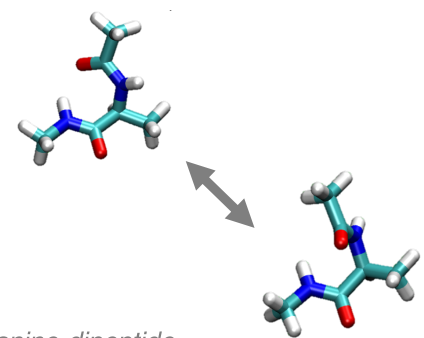
- ▷ Quantum mechanics (wave functions)

- ▷ Bayesian statistical modelling

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

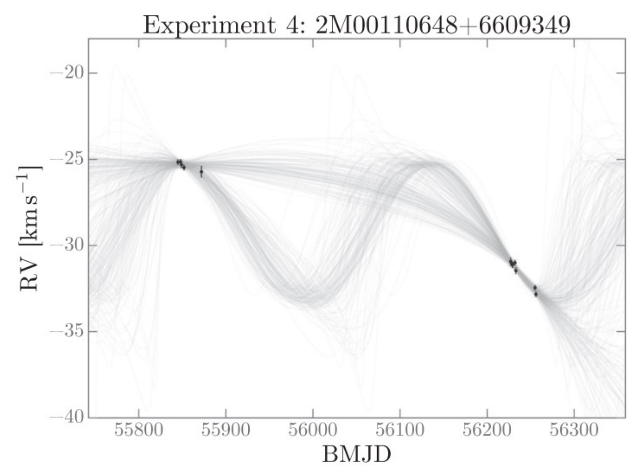
- ▷ Typically known up to normalization constant

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

Now that I know the Boltzmann distribution, what can I do?

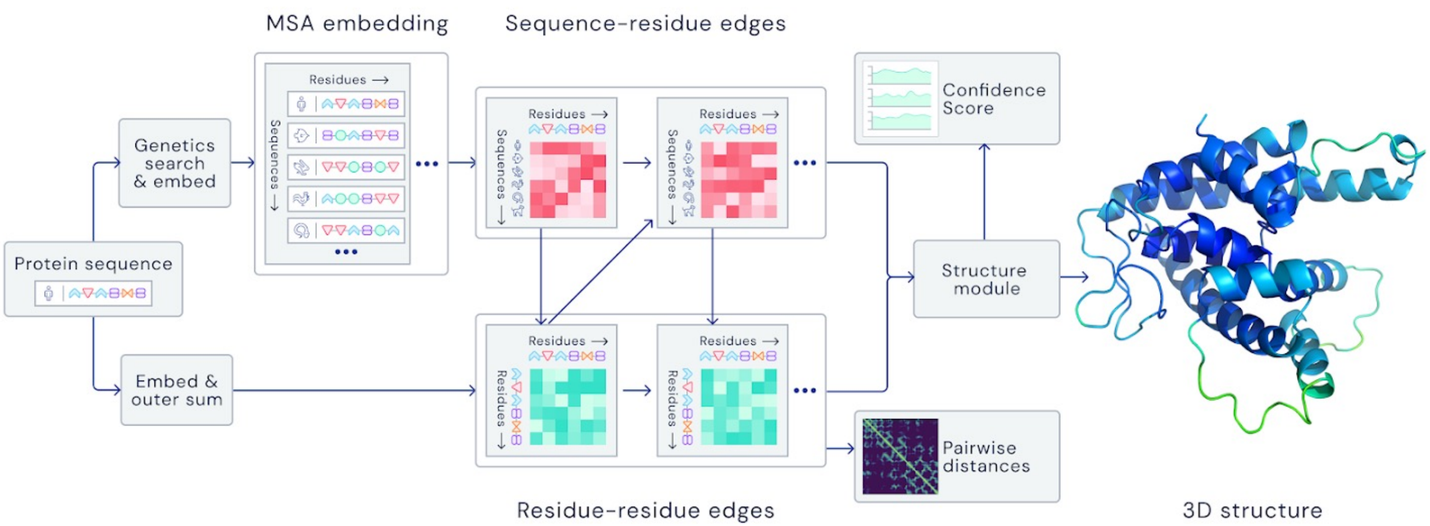
Distribution of physical state:

$$\rho(x) = \frac{1}{\mathcal{Z}_\beta} e^{-\beta U(x)} \qquad x \in \Omega \subset \mathbb{R}^D$$

e.g. particle positions, field values on a lattice etc …

▷ Next?

- Look for ground states $U_0 = U(x_0) = \min_x U(x)$



[Jumper et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583–589. (2021)]

Now that I know the Boltzmann distribution, what can I do?

Distribution of physical state:

$$\rho(x) = \frac{1}{\mathcal{Z}_\beta} e^{-\beta U(x)} \qquad x \in \Omega \subset \mathbb{R}^D$$

e.g. particle positions, field values on a lattice etc …

▷ Next?

- Look for ground states $U_0 = U(x_0) = \min_x U(x)$
- Compute equilibrium properties
  - Approximate the partition function (and derivatives!) $\mathcal{Z}_\beta = \int_\Omega e^{-\beta U(x)} \mathrm{d}x$

    in some models/limits e.g. mean field limits
  - Sample !

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

  Are we done already ??

  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. Gibbs sampling, Metropolis-Hastings

- ▷ **Importance** sampling:

  Reweight samples from an “easy” distribution $\mathbb{E}_\rho[f(x)] \approx \frac{1}{N} \sum_{i=1}^{N} w_i f(x_i)$

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

Outline

T
O
D
A
Y

1. A couple of important sampling methods
   1.1 - Importance sampling
   1.2 - Metropolis-Hasting
2. Unsupervised learning / generative models
   2.1 - Latent deep generative models
   2.2 - Normalizing flows
3. Combining traditional inference method and learning
   3.1 - Variational Inference
   3.2 - Adaptive algorithms

T
H
U
R
S
D
A
Y

4. Will it scale?
   4.1 - Local sampling in reparametrized space
   4.2 - Local-global sampling
   4.3 - Joining forces with annealing
   4.4 - Leveraging physics

1.1 Importance Sampling

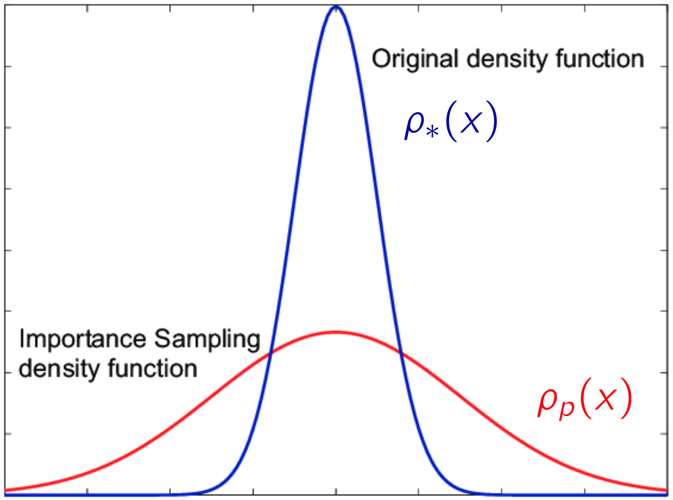
▷ Context: $\rho_*(x) = \frac{1}{\mathcal{Z}} e^{-U(x)}$ with unkown $\mathcal{Z}$

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

▷ Importance sampling

- Samples from proposal distribution $x_i \sim \rho_p(x_i)$ e.g. Gaussian, factorized, …
- Self-normalized weights $w_i = \frac{e^{-U(x_i)}/\rho_p(x_i)}{\sum_{i=1}^N e^{-U(x_i)}/\rho_p(x_i)}$
- Compute $\mathbb{E}_{\rho^*}[f(x)] \approx \frac{1}{N}\sum_{i=1}^N w_i f(x_i)$
- Asymptotically "unbiased" $N \to \infty$

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



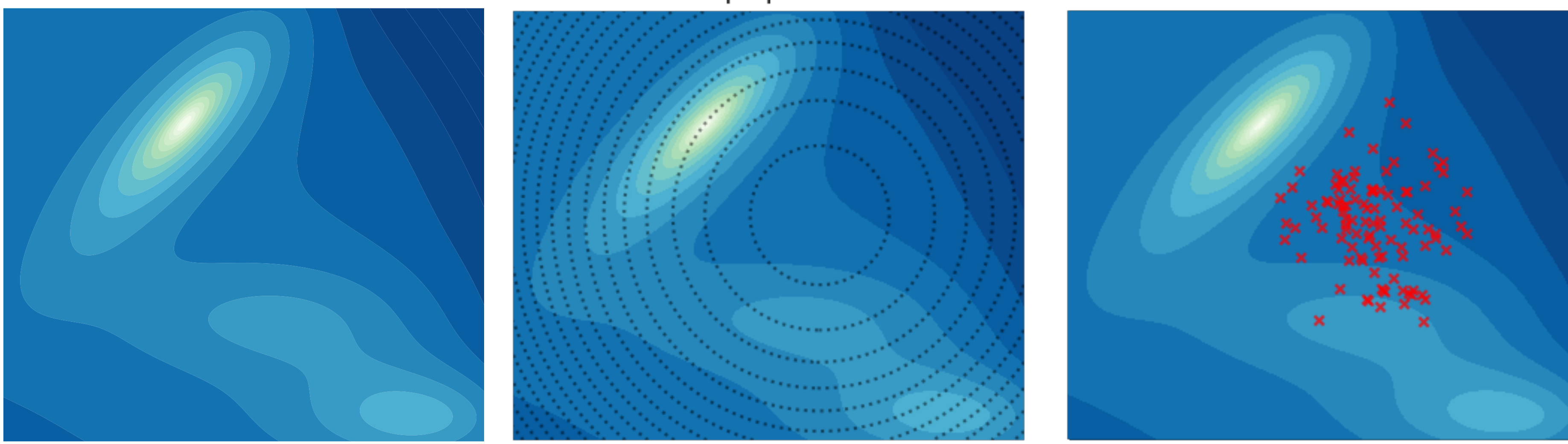
1.1 Importance Sampling: did it work?

▷ Look at a concrete example: 2d Muller Brown potential

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

Variance of proposal: 5.00e-02

ESS: 104



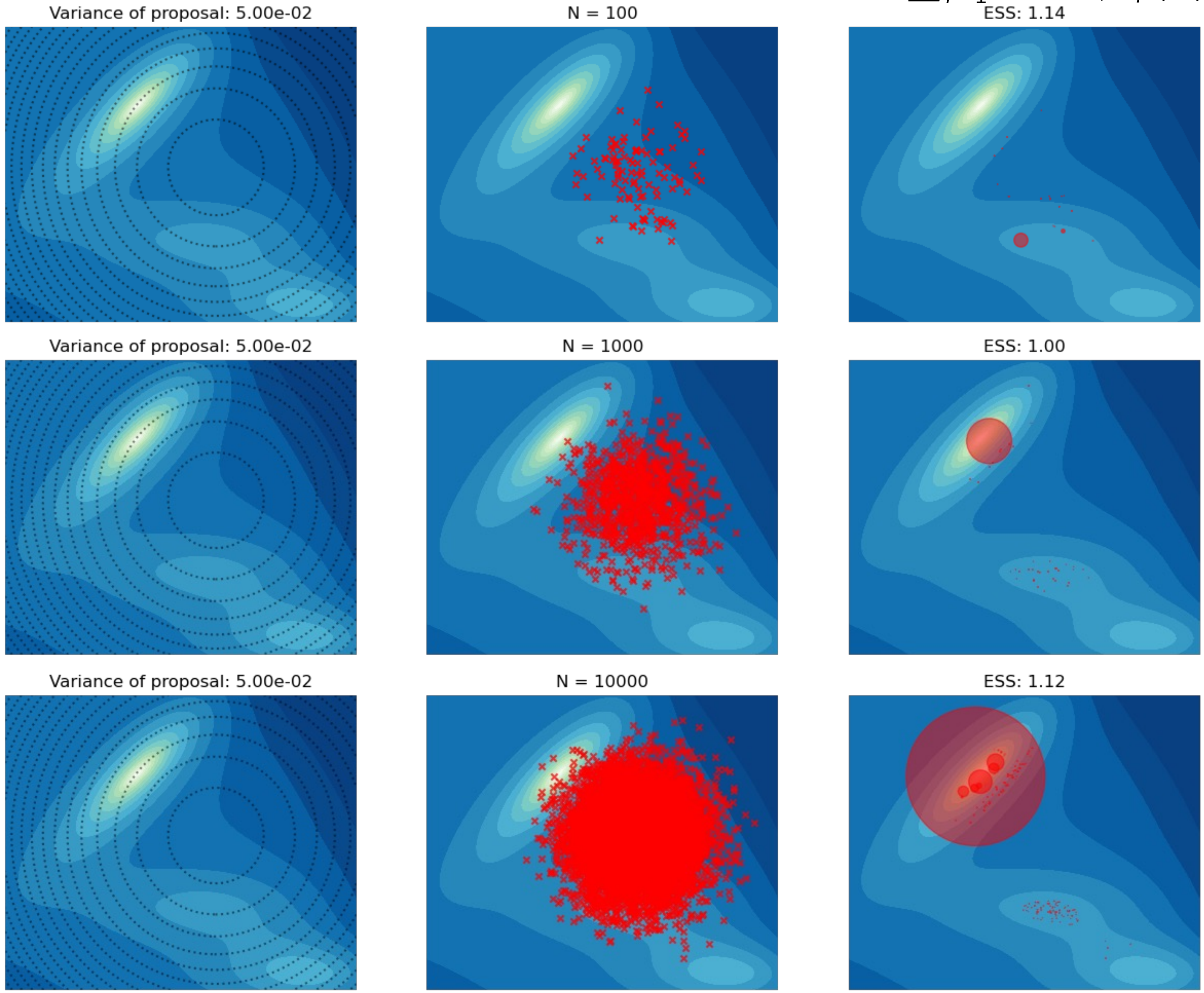
$$w_i = \frac{e^{-U(x_i)}/\rho_p(x_i)}{\sum_{i=1}^{N} e^{-U(x_i)}/\rho_p(x_i)}$$

▷ What can go wrong?

1.1 Importance Sampling: did it work?

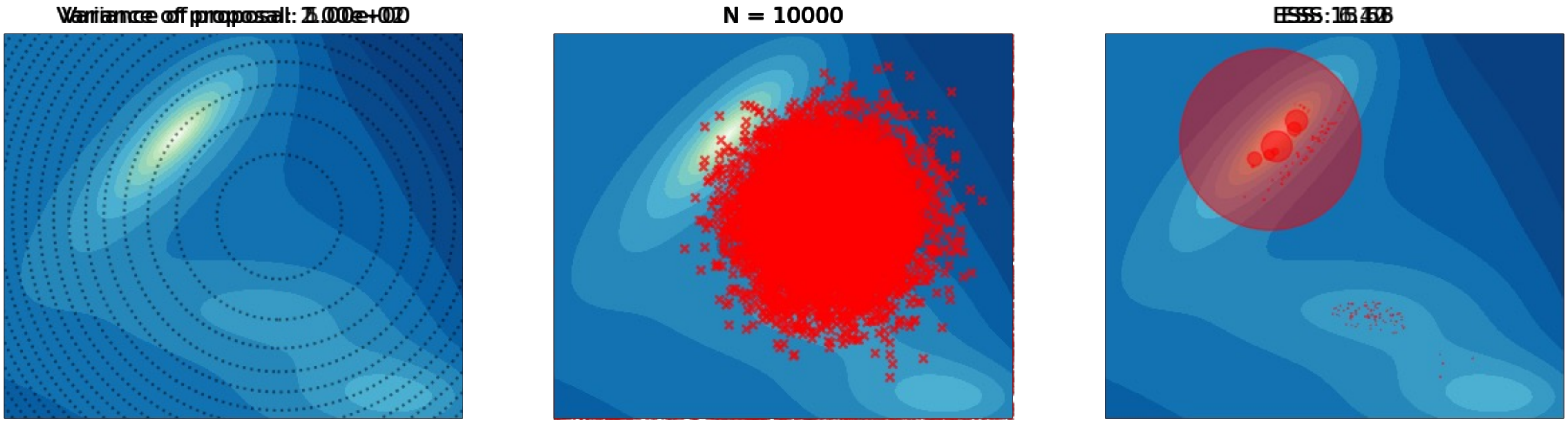
$$w_i = \frac{e^{-U(x_i)}/\rho_p(x_i)}{\sum_{i=1}^{N} e^{-U(x_i)}/\rho_p(x_i)}$$

$\rho_p(x)$ $\qquad$ $x_i \sim \rho_p(x_i)$



1.1 Importance Sampling: did it work?

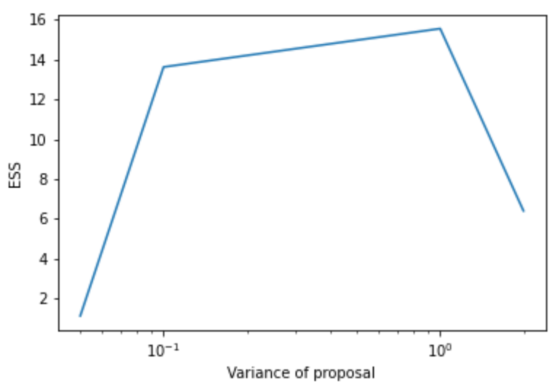
▷ Changing the variance now



▷ How will I realize this is happening?

- Importance weights $$w_i = \frac{\rho_*(x_i)/\rho_p(x_i)}{\sum_{i=1}^{N} \rho_*(x_i)/\rho_p(x_i)}$$
- Effective sample size $$ESS = \frac{\left(\sum_{i=1}^{N} w_i\right)^2}{\sum_{i=1}^{N} w_i^2}$$
- All samples "participates" $w_i = 1/N \Rightarrow \; ESS = N$
- Only one "participates" $w_1 = 1, \; \Rightarrow \; ESS = 1$

Not too small, not too big ..



Proposal and target need to be well adapted!

1.2 Markov Chain Monte Carlo

▷ 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 $x_{t+1} = x_t$

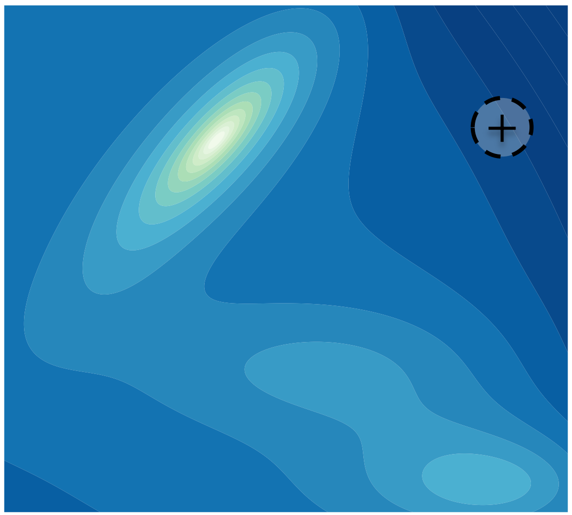
[e.g. Liu. *Monte Carlo Strategies in Scientific Computing,* 2004 – Brooks et al. *Handbook of MCMC,* 2011]

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 Adjusted) Langevin algorithm (MALA)

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

Metropolis-Hastings sampler

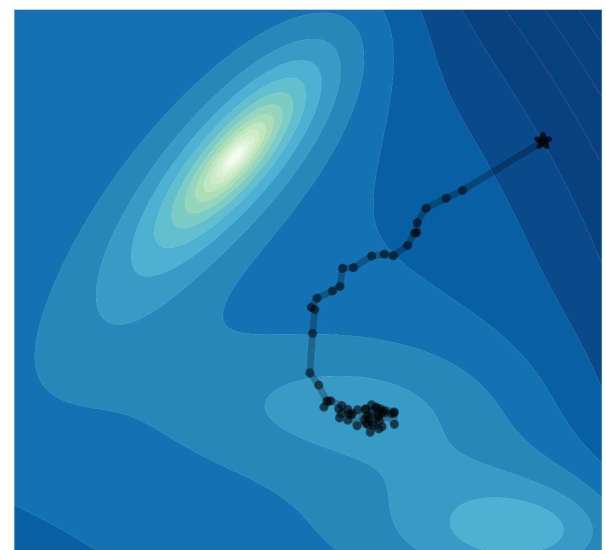
Initialize: x_0

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 $x_{t+1} = x_t$

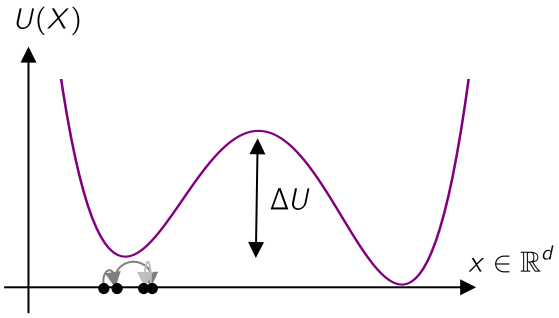


T = 100 steps

Challenge: Decorrelation and convergence

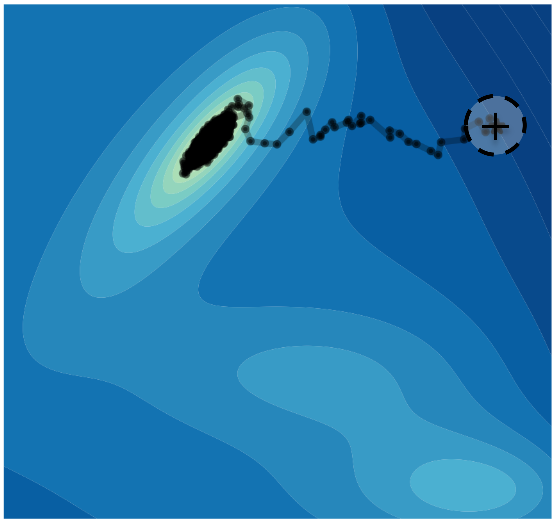
▷ Gaussian random walk $\rho_p(x_{t+1}|x_t) = \mathcal{N}(x_t, \Sigma)$

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



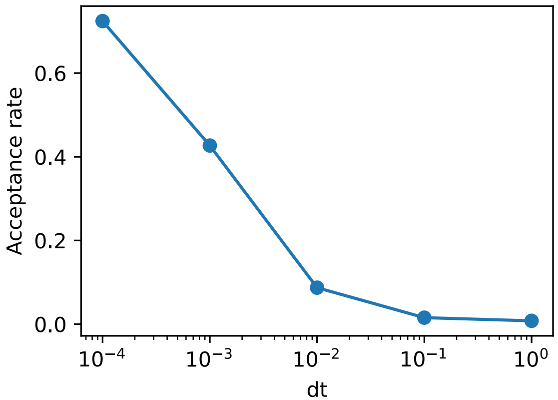
e.g. 2d Müller-Brown potential

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



T = 10000 steps

▷ Trade-off size local moves / acceptance



still difficult to switch mode!

▷ Many many proposition for faster "mixing"

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

powerful but computationally heavy!

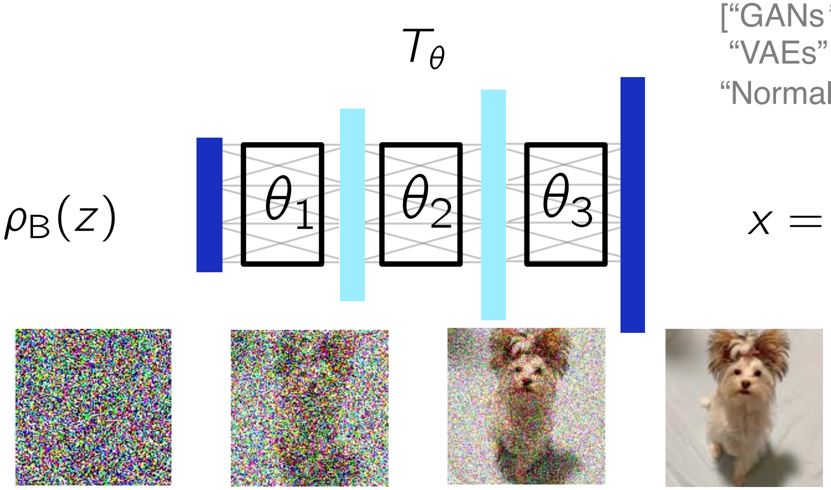
Outline

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

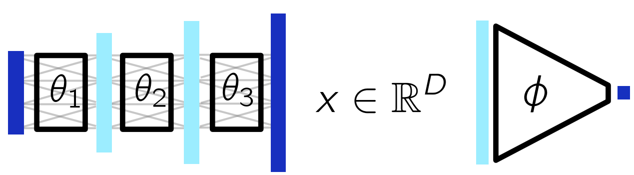
▷ Two main training methods of unsupervised learning:

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

+ SGD!

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

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



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

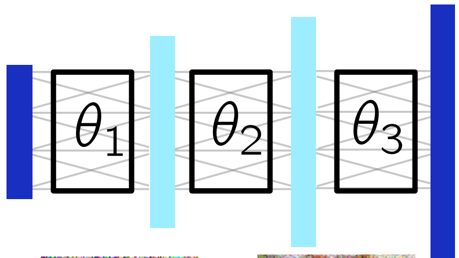
2.1 Deep generative models

▷ 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

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

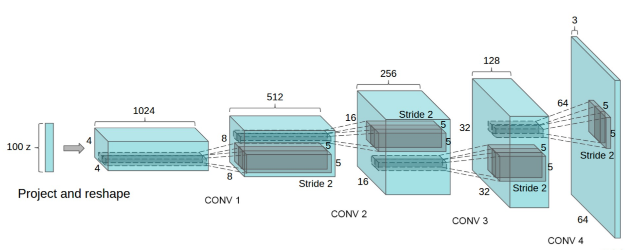


$x = T_\theta(z) \quad \sim \rho_\theta(x)$ “push-forward” distribution

Song et al. *ICLR* 2021

▷ Two main training methods of unsupervised learning:

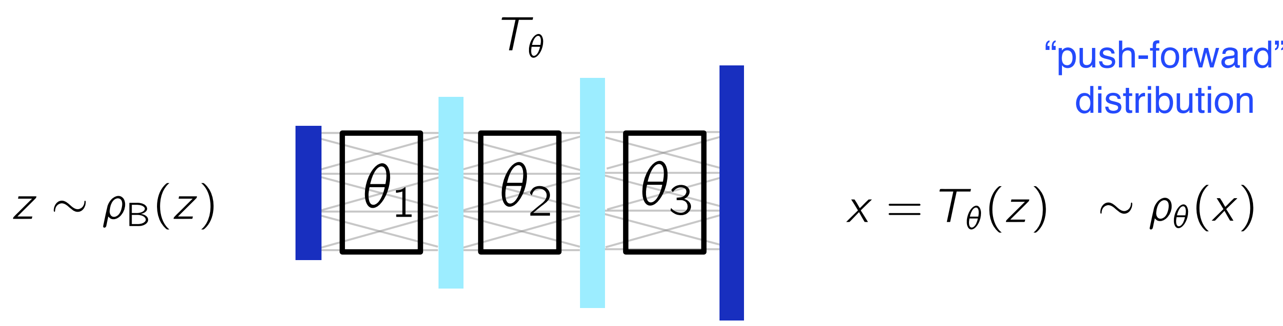
- Maximum likelihood: $L[\rho_\theta] = -\sum_{i=1}^{N} \log \rho_\theta(x_i)$ with x_i data samples
- Adversarial training: $\min_\theta \max_\phi \left[\mathbb{E}_{\rho_D} [\ln D_\phi(x)] + \mathbb{E}_{\rho_{\mathrm{B}}} [\ln(1 - D_\phi(T_\theta(z)))] \right]$ with ρ_D data distribution



[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



- 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]

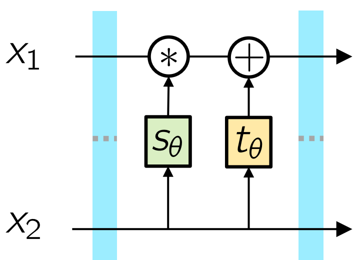
2.2 A special type of Deep Generative Models Normalizing Flows (NF): Invertible networks

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

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

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

Affine coupling layer $T_\theta(x)$

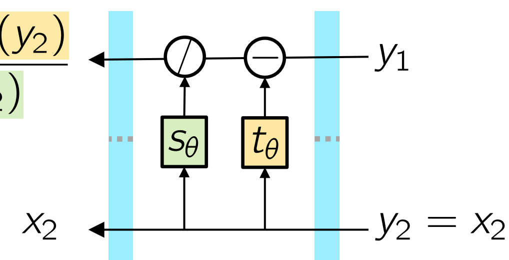


$$y_1 = s_\theta(x_2) * x_1 + t_\theta(x_2)$$

$$y_2 = x_2$$

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

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



Block diagonal Jacobian:

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

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

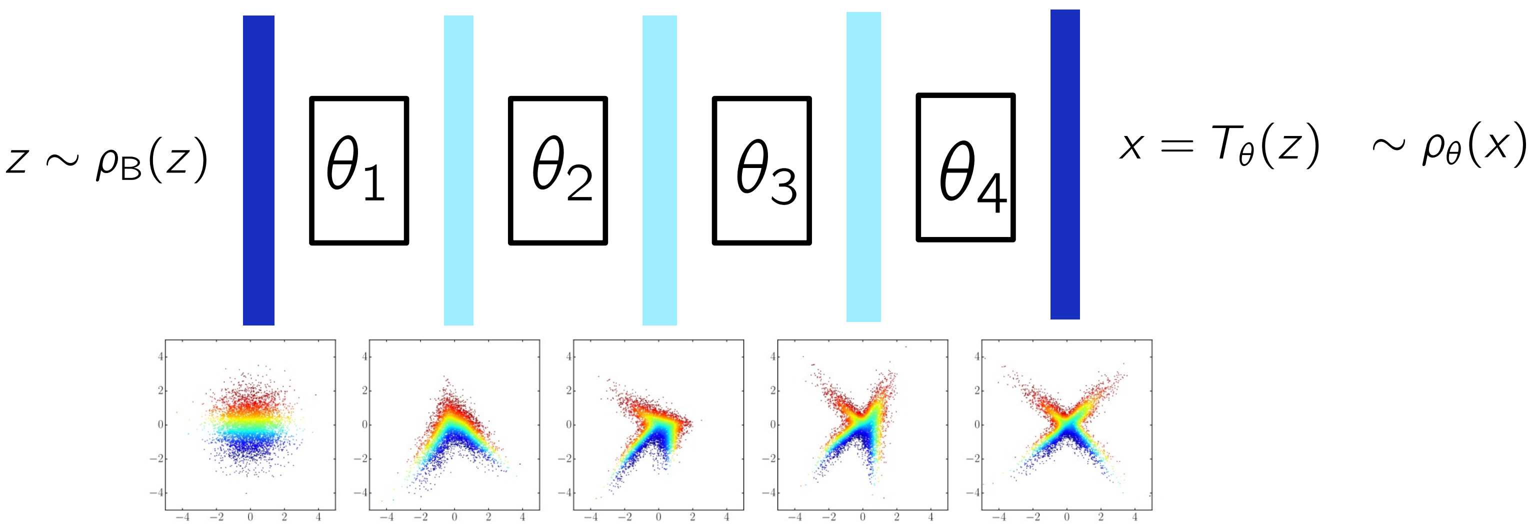
2.2 A special type of Deep Generative Models Normalizing Flows (NF): Invertible networks

▷ Parametrized invertible map $T_\theta : \Omega \mapsto \Omega \qquad \Omega \subset \mathbb{R}^d$

- Base distribution $z \sim \rho_B(z)$
- Push-forward distribution $x = T_\theta(z) \quad \sim \rho_\theta(x) = \rho_B(T_\theta^{-1}(x)) \det |\nabla_x T_\theta^{-1}|$

▷ Composition to encode for sophisticated transformations

$$T_\theta = T_{\theta_4} \circ T_{\theta_3} \circ T_{\theta_2} \circ T_{\theta_1}$$



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

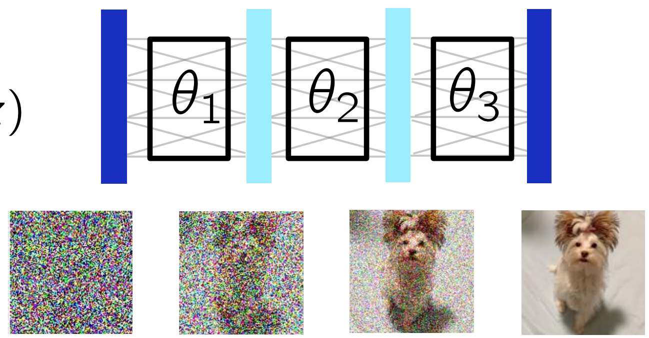
NADE networks also!

[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

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

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

▷ Opportunity alert! Sample complicated $\rho_*(x)$ by modelling it with deep generative model?

- Need to learn T_θ for which we need data - $x_i \sim \rho_*(x)$ - do we?

a.k.a. chicken-and-egg problem!

- Even if we get $\rho_\theta(x) \approx \rho_*(x)$, unlikely to learn perfect model $\rho_\theta(x) = \rho_*(x)$, right?

["Score based diffusion models" Song et al. *ICLR 2021*]

Outline

3.1. Variational Inference or do we really need data?

▷ Context: known $\rho_*(x) = \frac{1}{\mathcal{Z}} e^{-U(x)}$ up to normalizing factor $\mathcal{Z}$

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

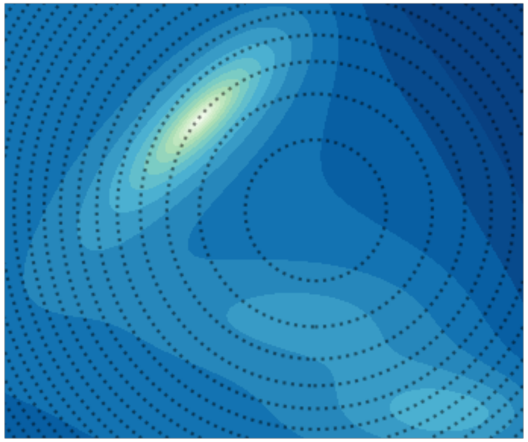
▷ Variational inference (VI):

- Optimize surrogate tractable distribution to minimize Kullback-Leibler divergence

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

- Then use the proxy for all purpose $\rho_\theta(x) \approx \rho_*(x)$

Quickly need expressive proxy!



▷ Questions:

1. Example of suitable $\rho_\theta(x)$?
2. Which problems do you anticipate?

1. Factorized/mean-field, Gaussian …
2. Quality of the approximation?

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.

Normalizing flows as powerful ansatz for VI

- ▷ Training without 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!

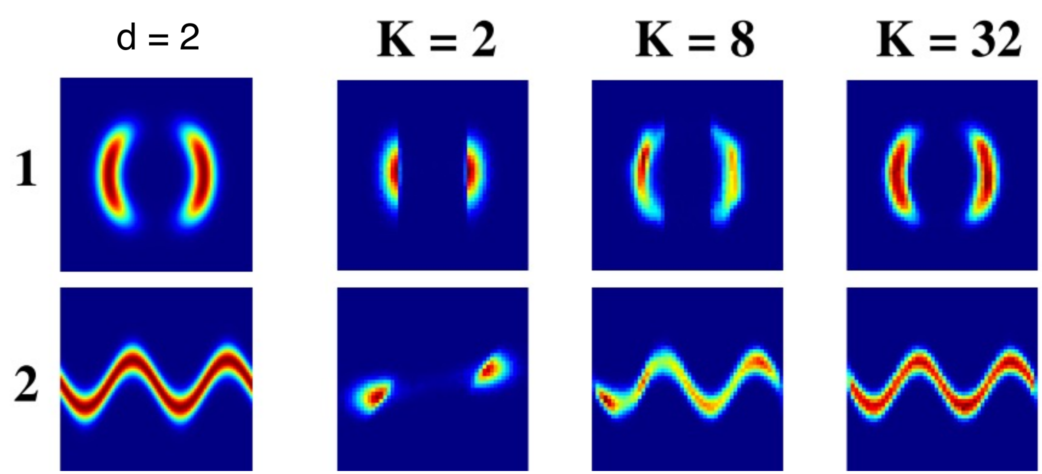
$$\rho_\theta(x) = \rho_{\mathrm{B}}(T_\theta^{-1}(x)) \det \left| \nabla_x T_\theta^{-1} \right|$$

explicit!

D. Rezende: “good entry point for ML”

- ▷ First results: quality as a function of expressivity

\# layers



Rezende & Mohamed, (2015). Variational inference with normalizing flows,
Albergo et al (2019). Flow-based generative models for Markov chain Monte Carlo in lattice field theory. *PRD 2019*.
Wu et al. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks.

Correcting the samples with a MCMC

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

Generative model parametrized density $\rho_\theta(x)$ trained by Variational inference

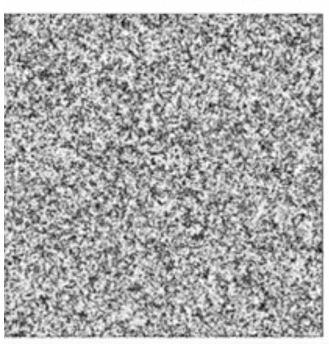
▷ Algorithm: Metropolis-Hastings with generative model proposal

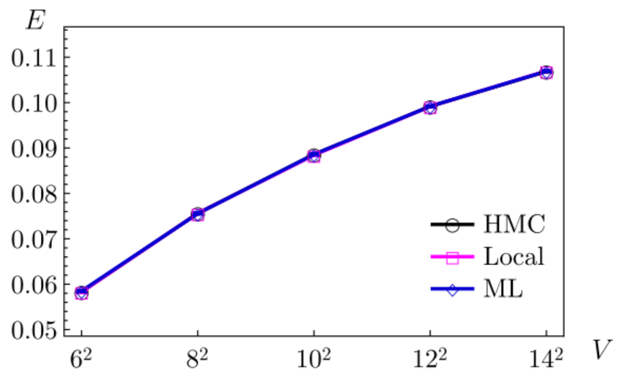
Initialize: $x_0^i \quad i = 1 \cdots N$

Loop:

- Draw from generative model $x_{t+1}^i \sim \rho_\theta(x)$ NF proposal!
- Accept-reject $\mathrm{acc}(x_{t+1}^i | x_t^i) = \min \left[1, \frac{\rho_*(x_{t+1}^i)\rho_\theta(x_t^i)}{\rho_*(x_t^i)\rho_\theta(x_{t+1}^i)} \right]$

▷ Φ^4 model at $T > T_c$





*"For simplicity in this initial work, all parameters were chosen to lie in the **symmetric phase**. In principle, the flow-based MCMC algorithm can be applied with identical methods to the broken-symmetry phase of the theory, but it **remains to be shown that models can be trained** for such choices of parameters."*

Albergo et al (2019). Flow-based generative models for Markov chain Monte Carlo in lattice field theory. *PRD 2019*.
Wu et al. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks.

Normalizing flows as powerful ansatz for VI

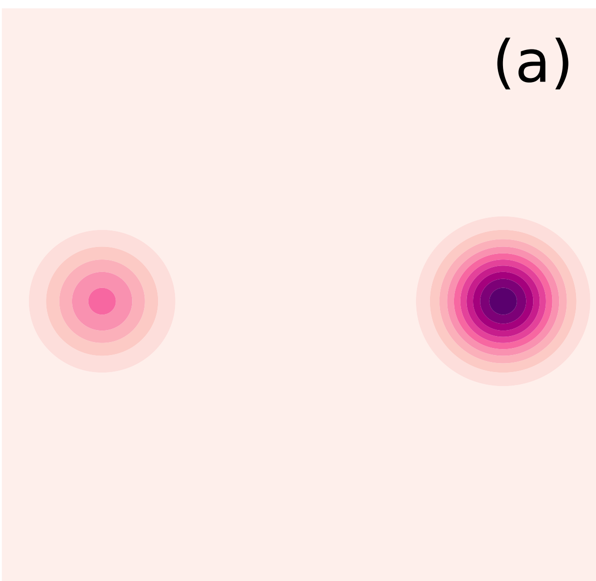
- ▷ No need for data?
  - ○ minimize Kullback-Leibler $D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$ = variational principle with expressive $\rho_\theta(x)$ ansatz

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

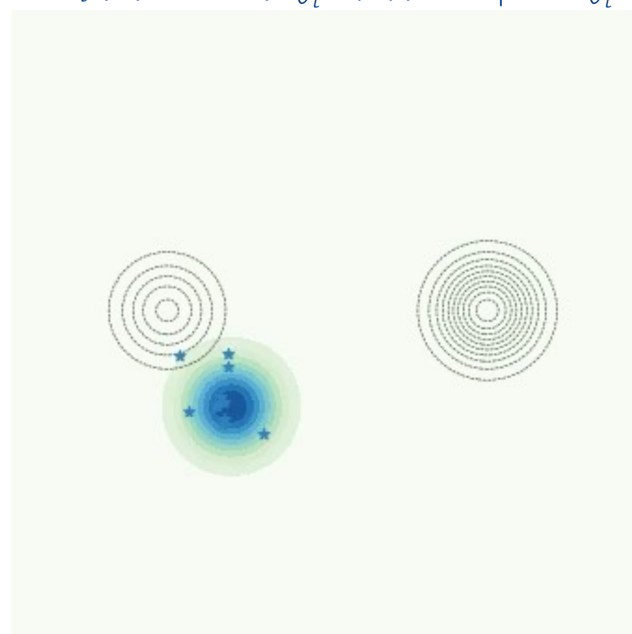
easy to obtain!

- ▷ What can do wrong?

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.

3.1 Combinig VI with a little dataset

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

▷ Minimize combined loss $L[\rho_\theta] = L_{\mathrm{VI}}[\rho_\theta] + L_{\mathrm{ML}}[\rho_\theta]$

- Training "by energy" (= Variational Inference)

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

- Training by data (= maximum likehood (ML))

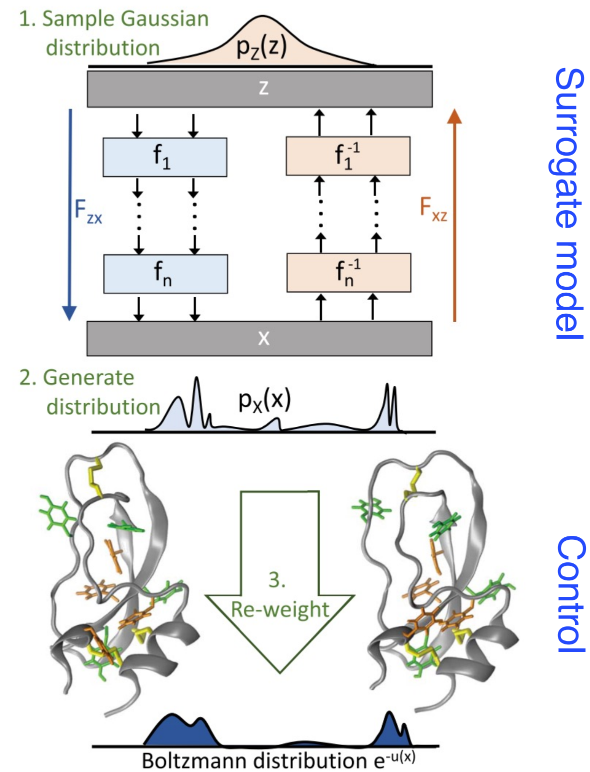
$$L_{\mathrm{ML}}[\rho_\theta] = -\sum_{i=1}^{N} \log \rho_\theta(x_{d,i})$$

$x_{d,i}$ small data set (from MD)

▷ Correct samples $x_i \sim \rho_\theta(x)$ with importance weights

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

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



e.g. BPTI protein (58 amino acids)

How much data is enough data?

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

3.2 Adaptive MCMC with normalizing flow

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

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

Another name could be: active learning!

▷ 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

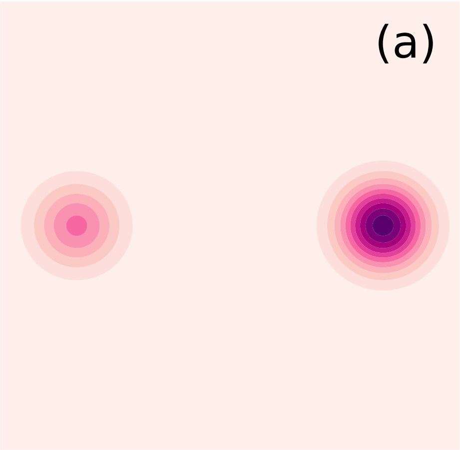
- Update NF paramters $\theta \leftarrow \theta + \eta \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log \rho_\theta(x_{t+1}^i)$

Maximum likelihood GD

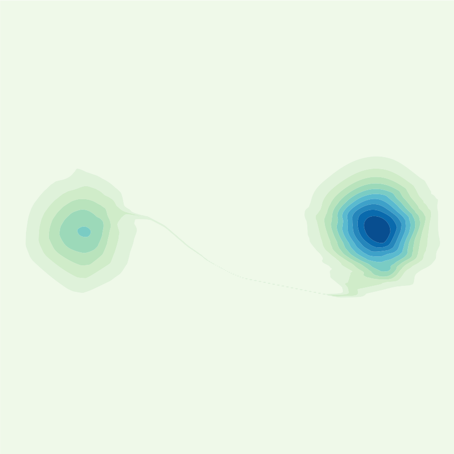
[Parno & Marzouk 2018, Gabrié, Rotskoff, Vanden-Eijden, *PNAS* 2022]

3.4 Adaptive MCMC – 2d Mixture of two Gaussians

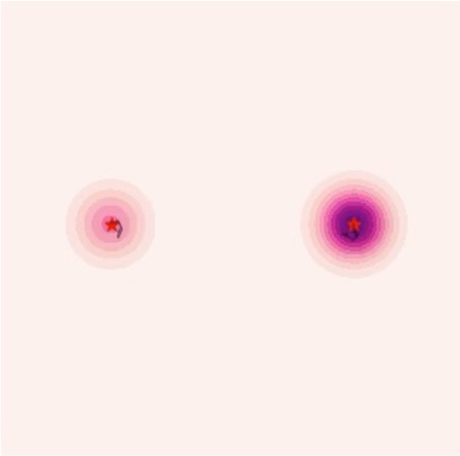
Target density:



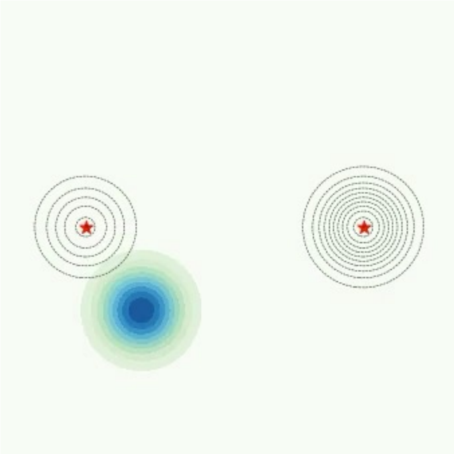
Final learned density:



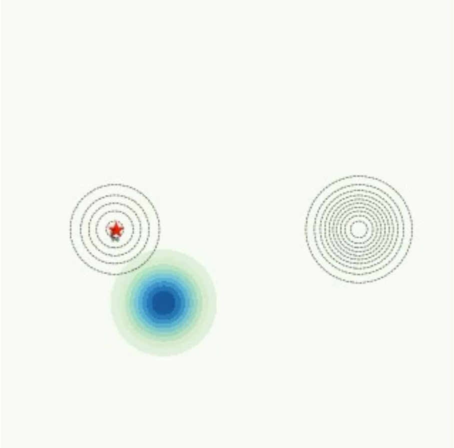
Local method only:



Concurrent:
careful intialization

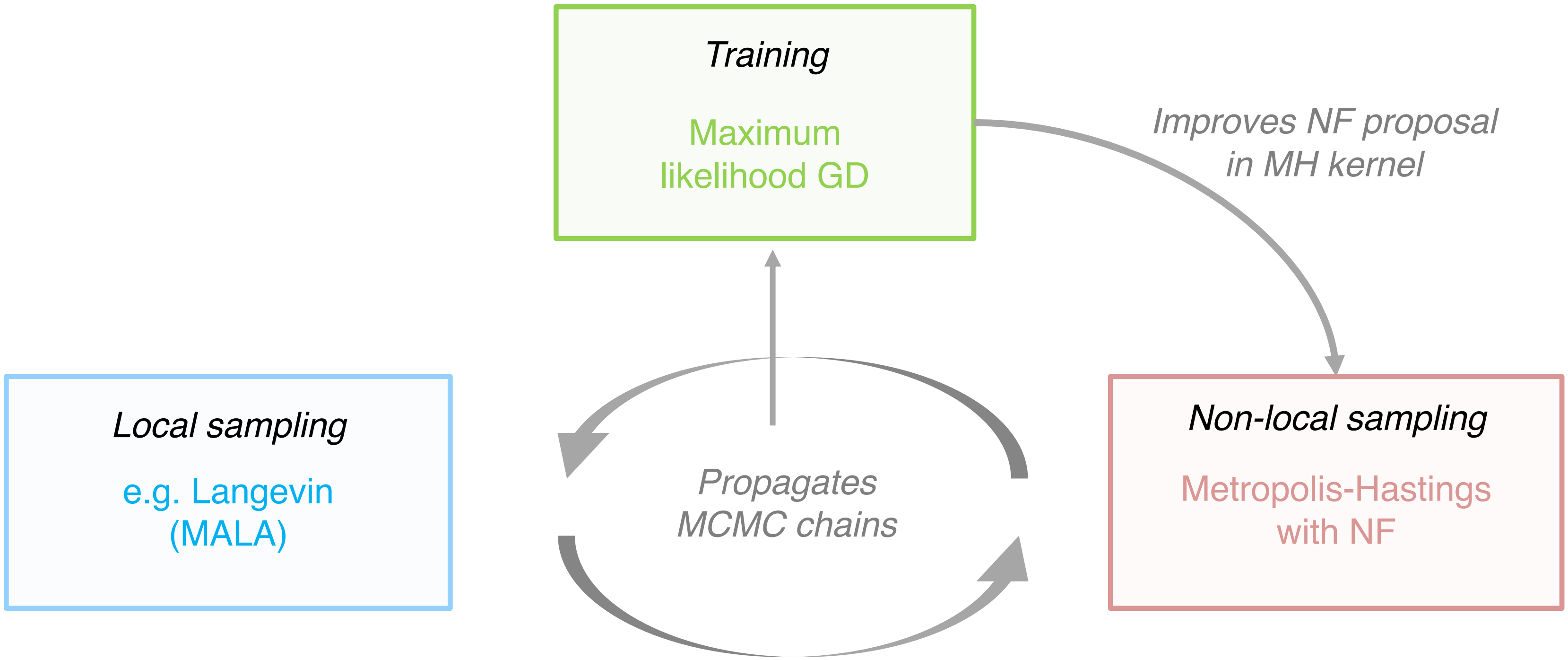


Concurrent:
starting with one walker



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, …]
- Softwares:
  - pytorch: marylou-gabrie / **flonaco** Public
  - jax: kazewong / **NFSampler** Public — with Kaze Wong (Flatiron Institute)

Outline

Y E S T E R D A Y

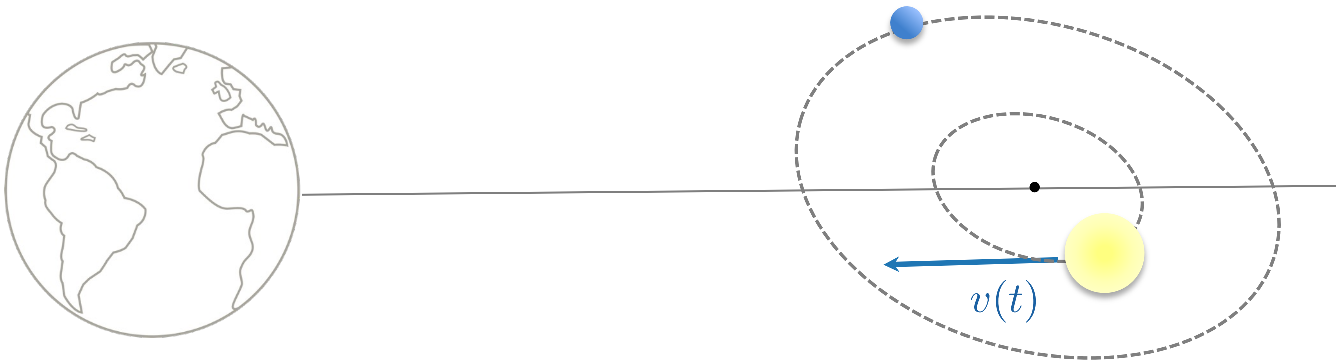
"Looks like, we can find training procedures for flows in this context and speed up sampling"

T O D A Y

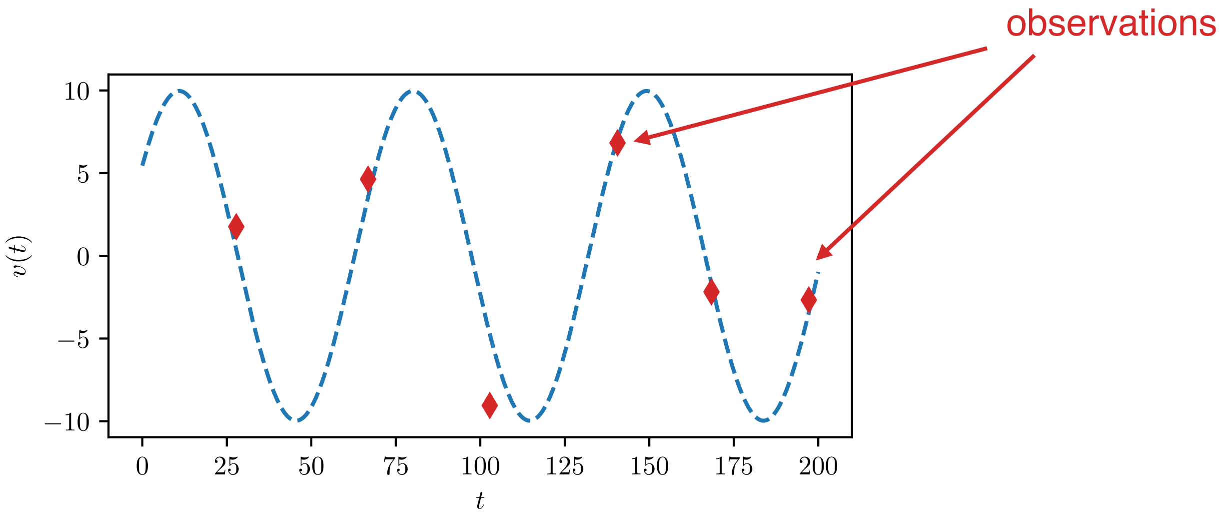
EXAMPLES

Bayesian inference:
An example of model selection from astrophysics

- ▷ Star-exoplanet system orbiting center of mass



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



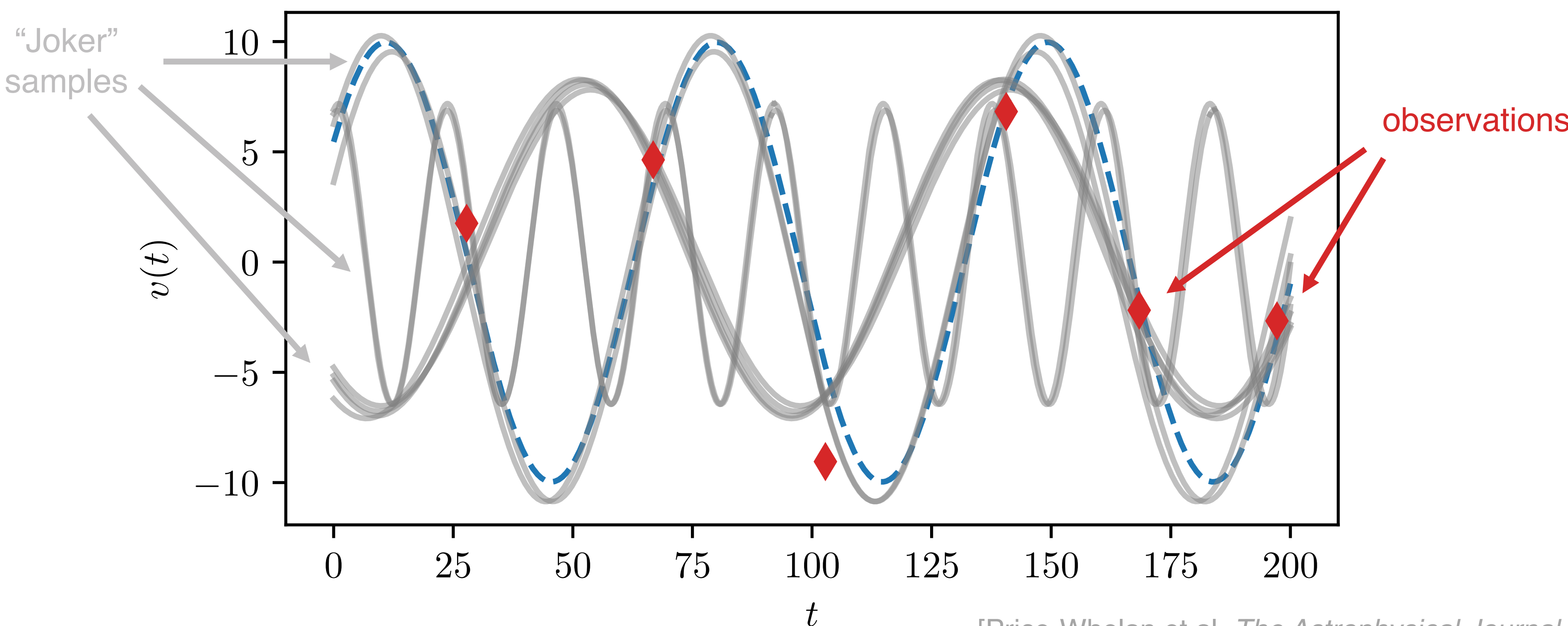
Bayesian model for velocity parameters

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

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

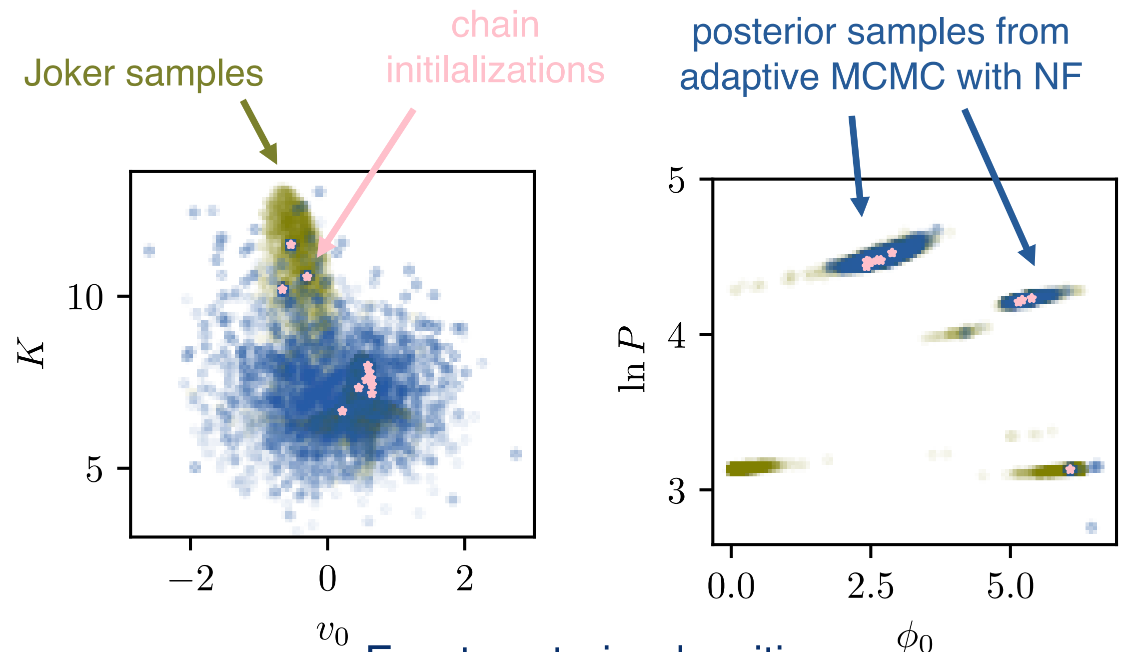
▷ Priors
$$\begin{aligned} \ln P &\sim \mathcal{U}(\ln P_{\min}, \ln P_{\max}), \\ \phi_0 &\sim \mathcal{U}(0, 2\pi), \\ K &\sim \mathcal{N}(\mu_K, \sigma_K^2), \\ v_0 &\sim \mathcal{N}(0, \sigma_{v_0}^2). \end{aligned}$$

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

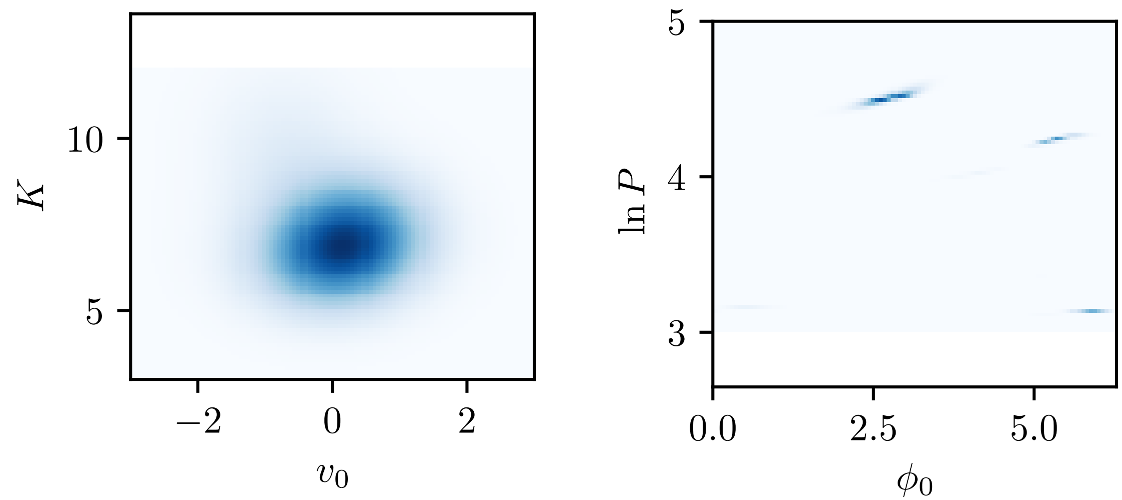


[Price-Whelan et al. *The Astrophysical Journal* 2017]

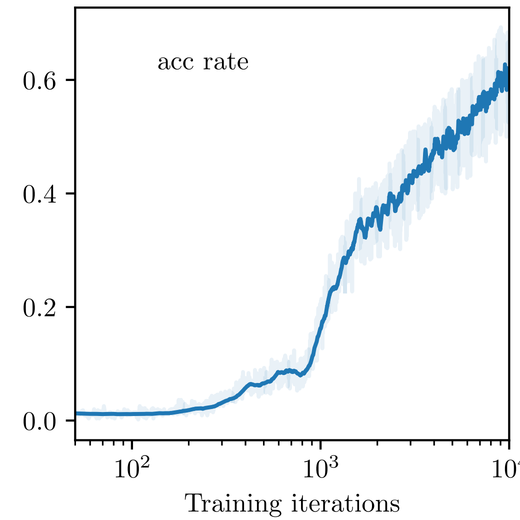
Sampling from the posterior



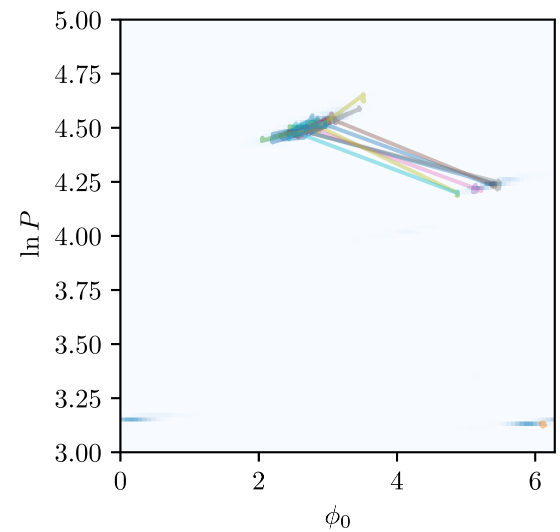
Exact posterior densities



Acceptance along training



Fast mixing chains

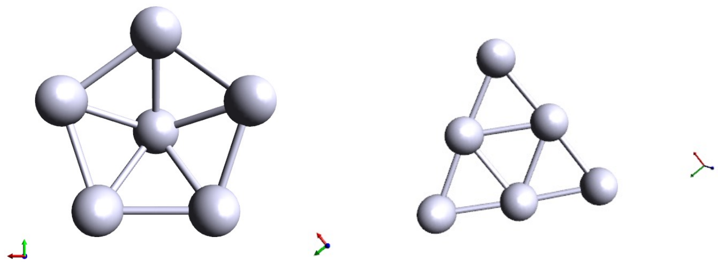


[Gabrié, Rotskoff & Vanden-Eijnden - *ICML workshop on INNF+* - arXiv 2107.08001]

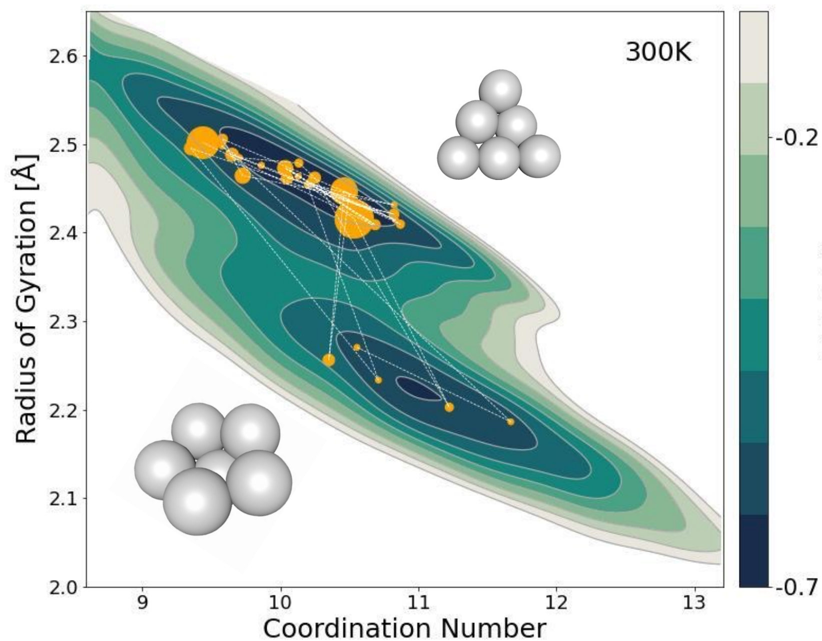
Sampling metastable silver clusters

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

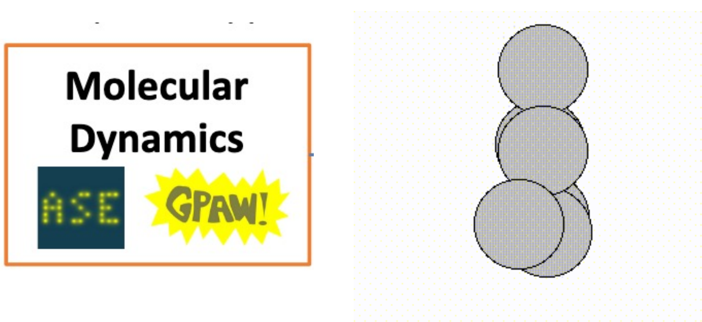
- ▷ Target density: Ground truth = Density Fonctional Theory: 2 metastable isomers



2 dimension projection of the free energy surface



- ▷ Local sampler: Molecular Dynamics



- ▷ Adaptive MCMC jumping between isomers

[Taborda et al. in preparation]

Outline

Y E S T E R D A Y

1. A couple of important sampling methods

   1.1 - Importance sampling

   1.2 - Metropolis-Hasting

2. Unsupervised learning / generative models

   2.1 - Latent deep generative models

   2.2 - Normalizing flows

3. Combining traditional inference method and learning

   3.1 - Variational Inference

   3.2 - Adaptive algorithms

T O D A Y

4. Will it scale?

   4.1 - Local sampling in reparametrized space

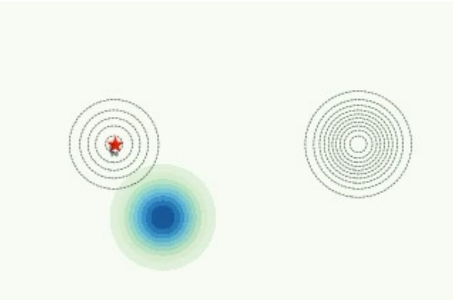
   4.2 - Local-global sampling

   4.3 - Joining forces with annealing

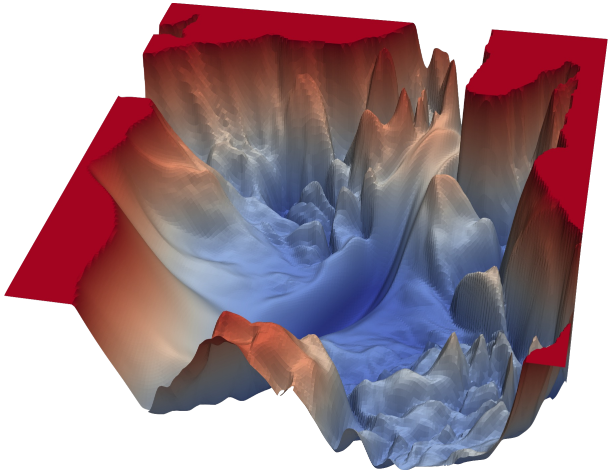
   4.4 - Leveraging physics

Will it scale? A few hard problems

- ▷ Mode finding



- ▷ Disordered/Glassy landscapes



https://www.cs.umd.edu/~tomg/projects/landscapes/

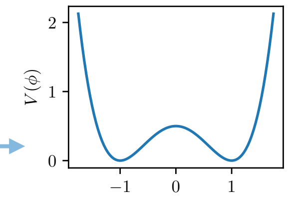
- ▷ Probing bigger and bigger systems

Can the method scale to big systems?

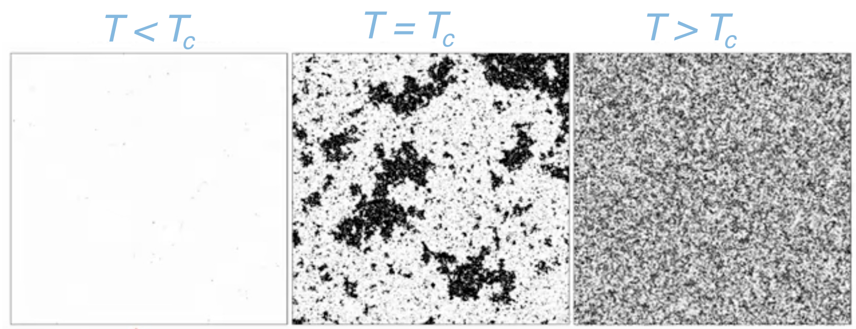
▷ A systematic study on the 2d - Φ^4 model (Del Debbio, *PRD* 2021)

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

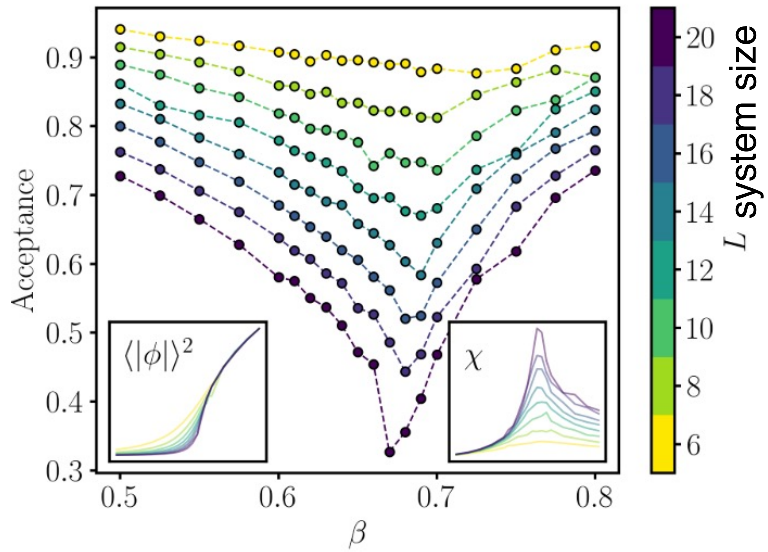
  coupling term — local potential →



- Typical configurations



- Mean acceptance probability after training

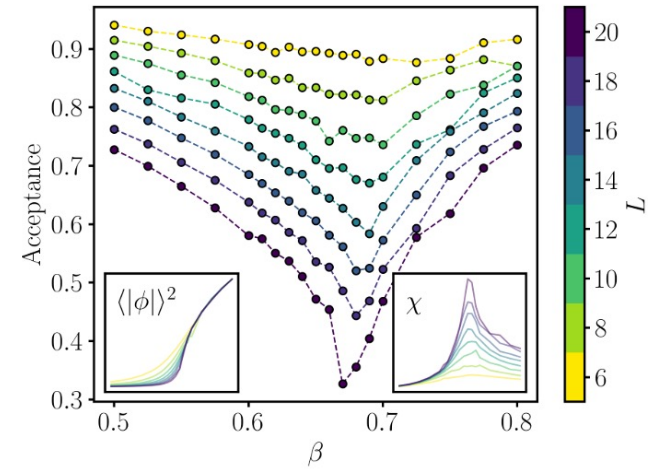


Efficient Modelling of Trivializing Maps for Lattice φ4 Theory Using Normalizing Flows: A First Look at Scalability
Del Debbio et al *PRD* 2021

Scaling to larger and larger systems

2d - Φ^4 model

Del Debbio et al *PRD* 2021



▷ Can surrogate probabilistic models scale?

▷ Metropolis acceptance scaling with dimension

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

$$\Delta U_* = U_*(x_{t+1}) - U_*(x_t) \approx O(D)$$ i.e. energy is typically extensive

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

- Same story if importance sampling

▷ A first idea: be less ambitious and retain some locality in sampling

Loop over:

- Propose $x^i_{t+1} \sim \rho_\theta(x)$ Independ flow proposal ← $x_{t+1} \sim \pi_\theta(x_{t+1}|x_t)$
- Accept/reject $\mathrm{acc}(x^i_{t+1}|x^i_t) = \min\left[1, \frac{\rho_*(x^i_{t+1})\rho_\theta(x^i_t)}{\rho_*(x^i_t)\rho_\theta(x^i_{t+1})}\right]$

Conditional proposal less local than tradiotinal kernel?

Outline

1. A couple of important sampling methods

    1.1 - Importance sampling

    1.2 - Metropolis-Hasting

2. Unsupervised learning / generative models

    2.1 - Latent deep generative models

    2.2 - Normalizing flows

3. Combining traditional inference method and learning

    3.1 - Variational Inference

    3.2 - Adaptive algorithms

4. Will it scale?

    4.1 - Local sampling in reparametrized space

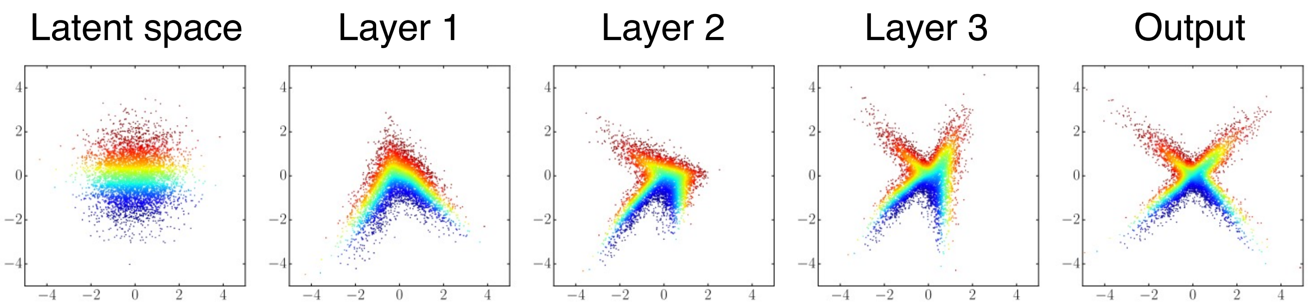
    4.2 - Local-global sampling

    4.3 - Joining forces with annealing

    4.4 - Leveraging physics

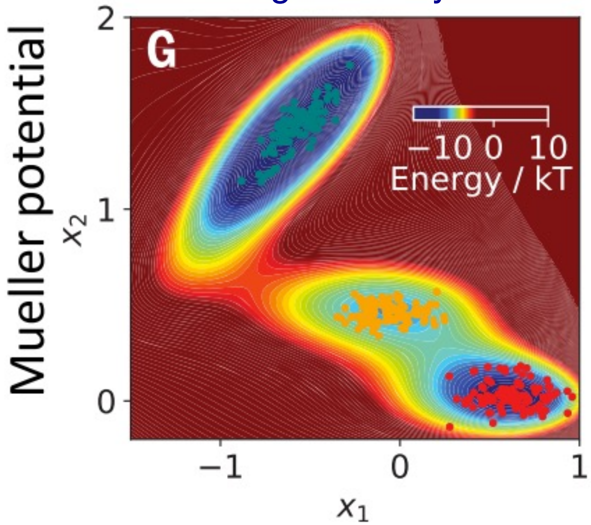
4.1 Reparametrization: reverse NF for MCMC

▷ Reverse transformation is normalizing = “Gaussianizing”

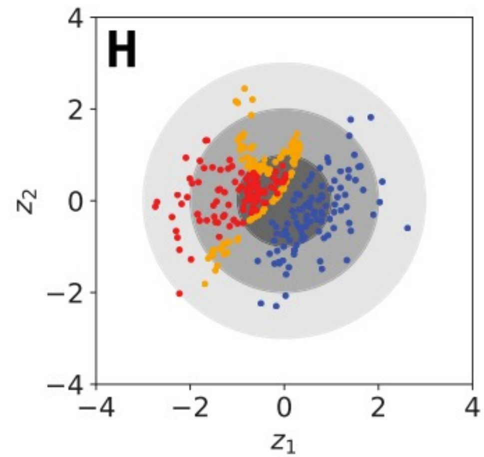


▷ Idea: train normalizing flow and use latent space to run traditional MCMC

Target density:



Latent space



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.

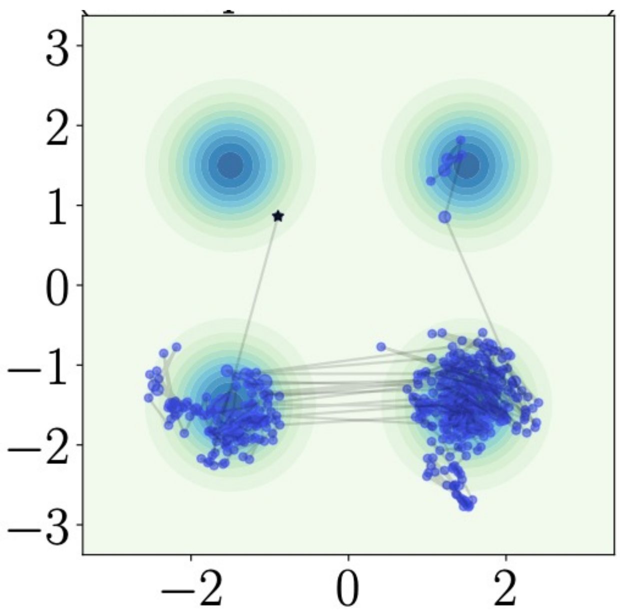
4.1 Reparametrization: reverse NF for MCMC

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

- ▷ Test case:
  1. Train a flow on a mixture of Gaussian
  2. Run MALA in the "latent space" on the push-backward

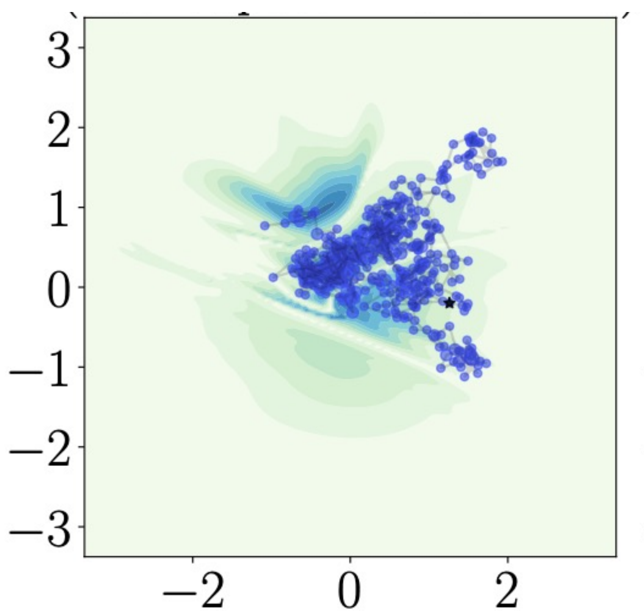
Target density:

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



Push backward:

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



(Experiments by 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.

4.2 Global-Local samplers: the best of both worlds

With Eric Moulines, Sergey Samsonov and collaborators.

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

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

▷ Algorithm: “Explore-Exploit MCMC”

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

Loop over parallel chains:

- 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

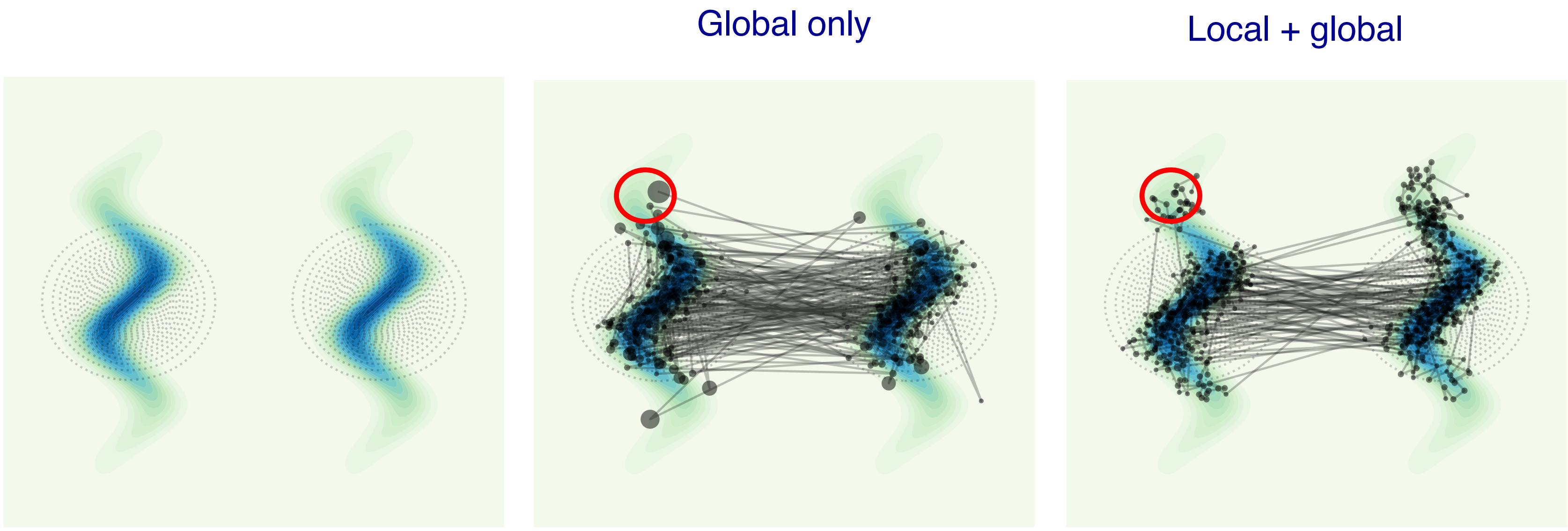
Local kernel

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

[Global-Local samplers: the best of both worlds - Samsonov et al. (ariXv2209.????)]

4.2 Global-Local samplers: the best of both worlds

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

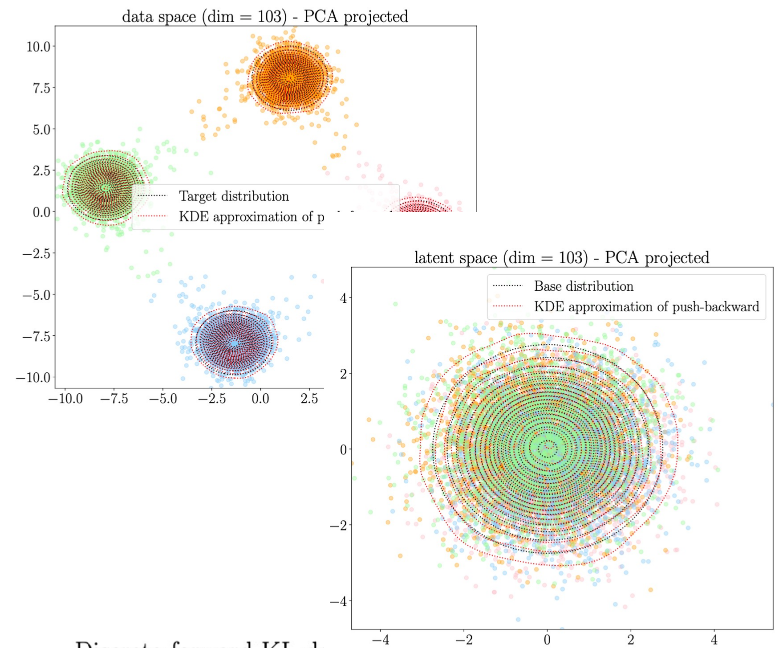


- ▷ Intuition: Local kernels allow to
  - ○ Explore regions that are not (yet) properly learned
  - ○ Drive learning there (if runing an adaptive MCMC, a.k.a. adaptive learning)
- ▷ Possible to derive theoretical guarantees of the improvement they bring.

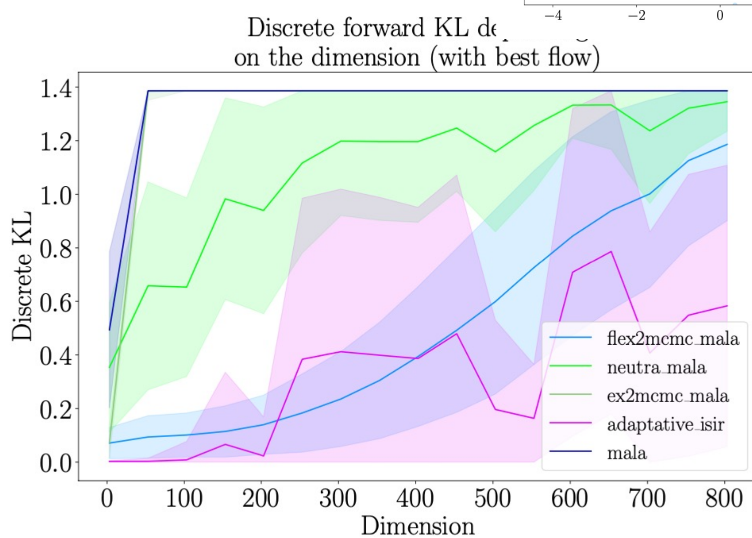
[Global-Local samplers: the best of both worlds - Samsonov et al. (ariXv2209.????)]

4.2 Global-Local samplers: the best of both worlds

▷ Does Global-local samplers scale better than NeuTraMCMC (transported MCMC)?



- Multimodal Gaussian mixture
- Train a flow to reproduce the mixture in higher dimension
- Measure of how many modes are visited by the differemt alforithms



Flows are better exploited with Explore-Exploit than the NeuTraMCMC !

[Grenioux, Moulines, Gabrié (in preparation)]

Outline

4.3 Joining forces with annealing

- ▷ Going back to Variational Inference:
  - Minimize Kullback-Leibler $D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$ = variational principle with expressive $\rho_\theta(x)$ ansatz

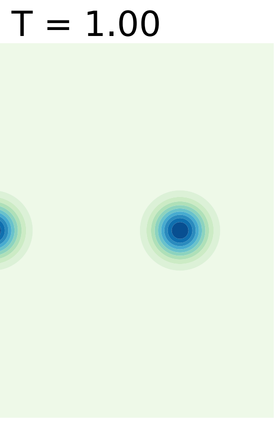
$$D_{\mathrm{KL}}(\rho_\theta \| \rho_*) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_\theta(x) \mathrm{d}x \approx \sum_{i=1}^{N} \log \frac{\rho_\theta(x_i)}{\rho_*(x_i)} \qquad x_i \sim \rho_\theta(x) \qquad \text{+ SGD!}$$

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

  - Recall, issue is mode collapse.

- ▷ Idea: start at high temperature and progressively decrease to desired model

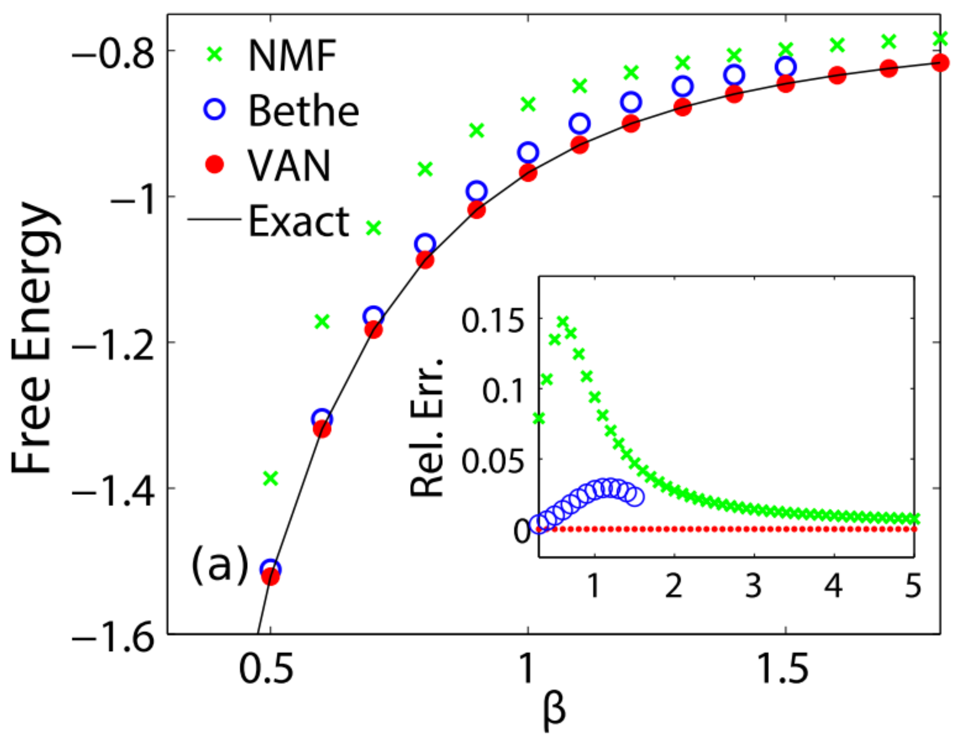
$$\rho_{\beta *}(x) = e^{-\beta U_*(x)}/Z$$



4.3 Joining forces with annealing

- ▷ "Solving Statistical Mechanics Using Variational Autoregressive Networks" Wu, Wang and Zhang (*PRL* 2019)
- ▷ Variational inference with annealed target + MCMC correction

Sherrington-Kirkpatrick model – N = 20 spins



Wu et al. (2019). Solving Statistical Mechanics Using Variational Autoregressive Networks.

4.3 Joining forces with annealing: Pushing towards more complicated models

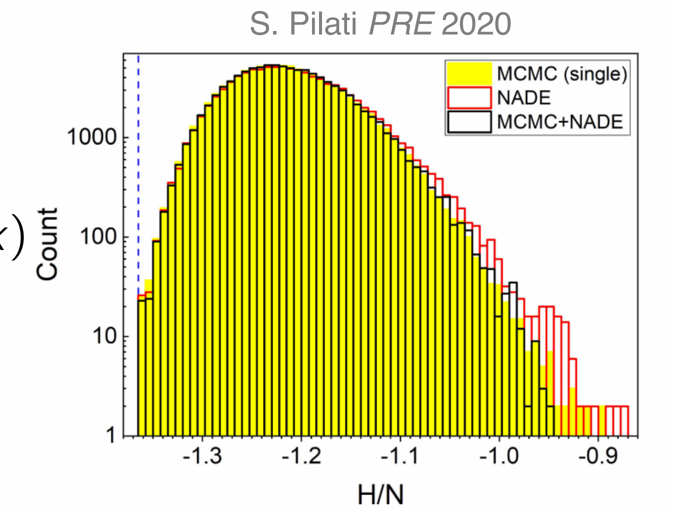
▷ Annealing to create progressively dataset of training

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

From high-temperature repeat:

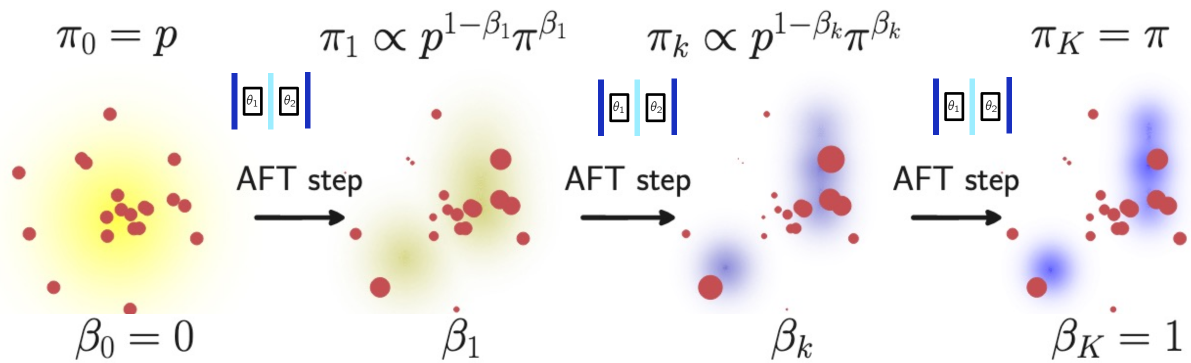
- Use $\rho_{\theta_{k-1}}^{\beta_{k-1}}(x) = \rho_p(x)$ in MCMC to sample $\rho_*^{\beta_k}(x)$
- Use $x_i \sim \rho_*^{\beta_k}(x_i)$ as data to train $\rho_{\theta_{k-1}}^{\beta_{k-1}}(x)$

2d – *Edwards Anderson*



▷ (Continuously Repeated) Annealed Flow Transport

- Add flow transport maps within steps of sequencial Monte Carlo (SMC)



Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks- B. McNaughton .., S. Pilati *PRE* 2020
(Continuously Repeated) Annealed Flow Transport Monte Carlo. Arbel, Matthews et al (2021 & 2022)

4.4 – Leveraging physics

▷ Using symetries and invariance

- cf Danilo's talk
- cf cluster updates by Wu, Rossi & Carleo, PRR (2021)

▷ Using informed base measures

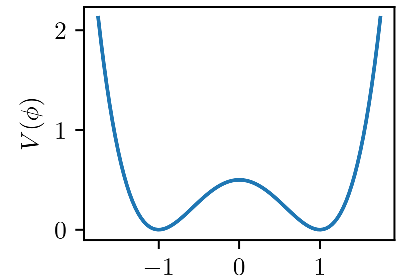
Wu, D., Rossi, R., & Carleo, G. (2021). Unbiased Monte Carlo cluster updates with autoregressive neural networks. *Physical Review Research*, *3*(4).

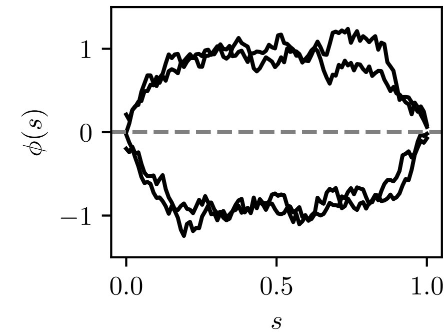
4.4 – Leveraging physics: informed base measures

- ▷ Example: 1d - 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

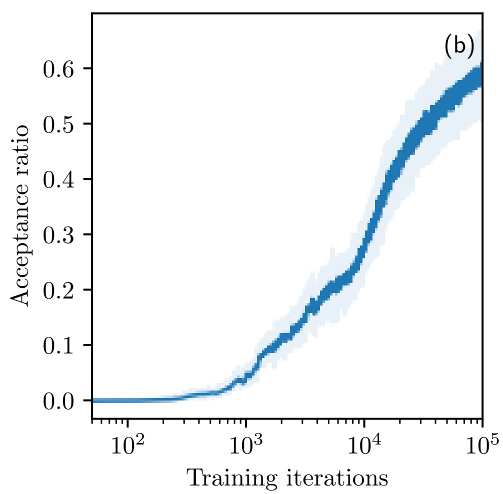
coupling term



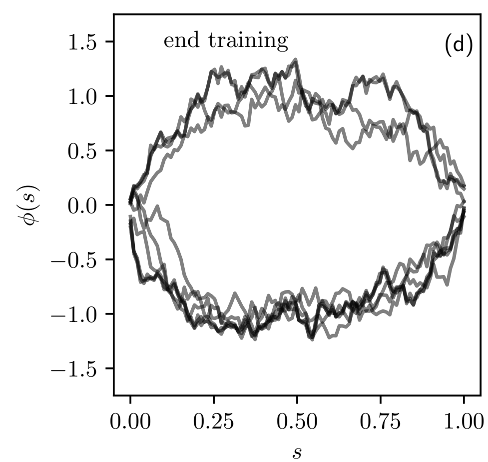


- ▷ Discretized: N=100

Acceptance ~ 60%



Fast mixing



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

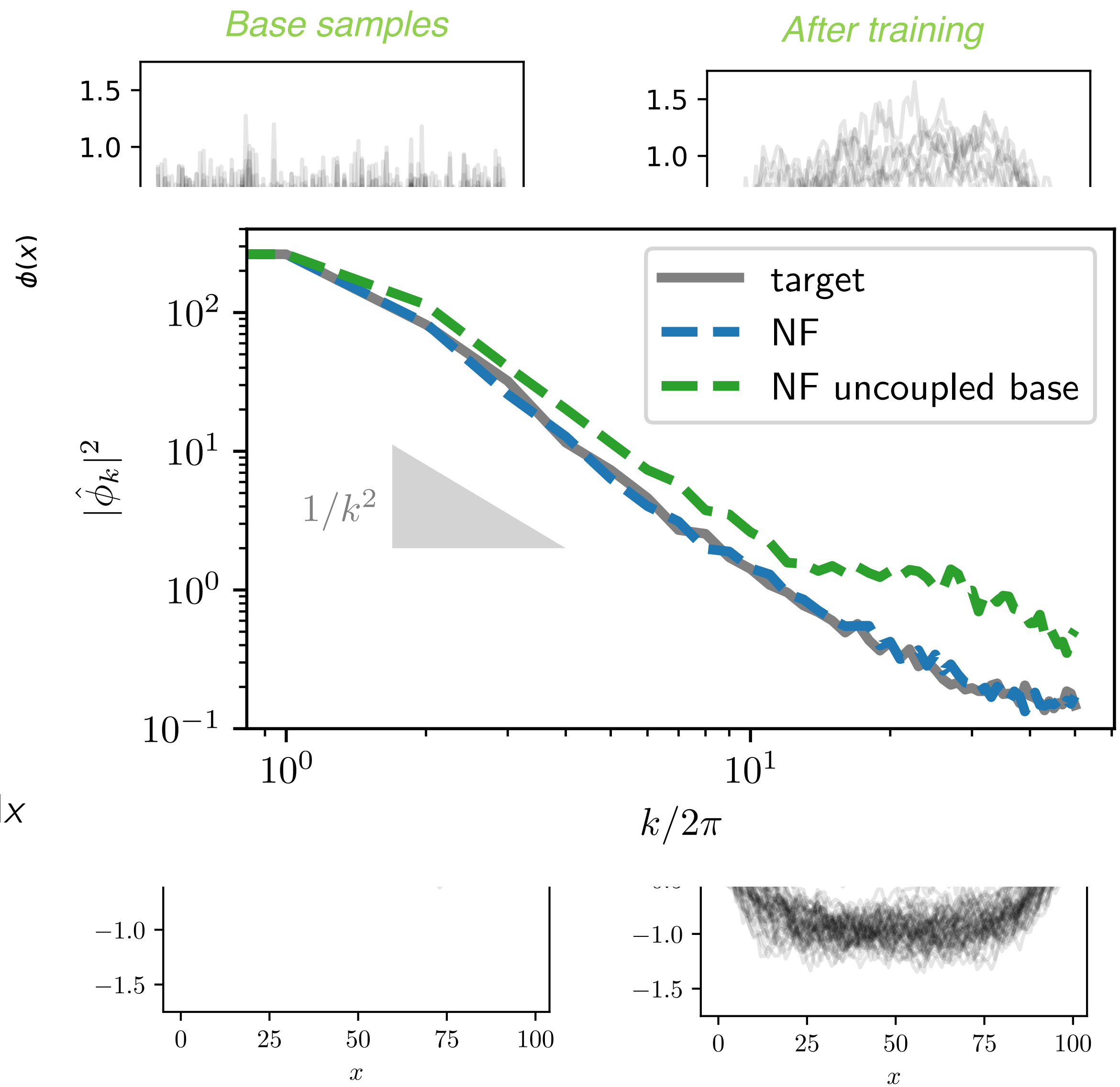
Uncoupled vs coupled base distributions

Gaussian uninformed (uncoupled)

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

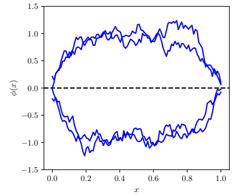
Gaussian informed (coupled)

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



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

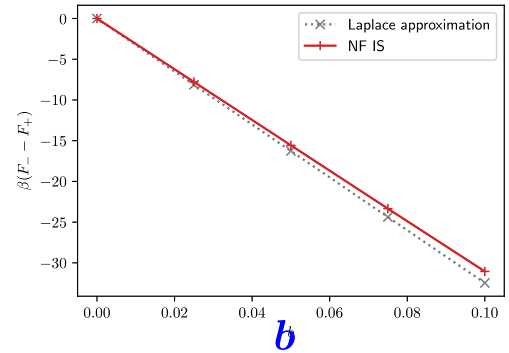
More numerical checks



- ▷ Tilt distribution towards -1 configuration with local field

$$U_{*,\boldsymbol{b}}(\phi) = \int \left(\frac{a}{2} |\nabla_x \phi|^2 + V(\phi) + \boldsymbol{b}\, \phi \right) \mathrm{d}x$$

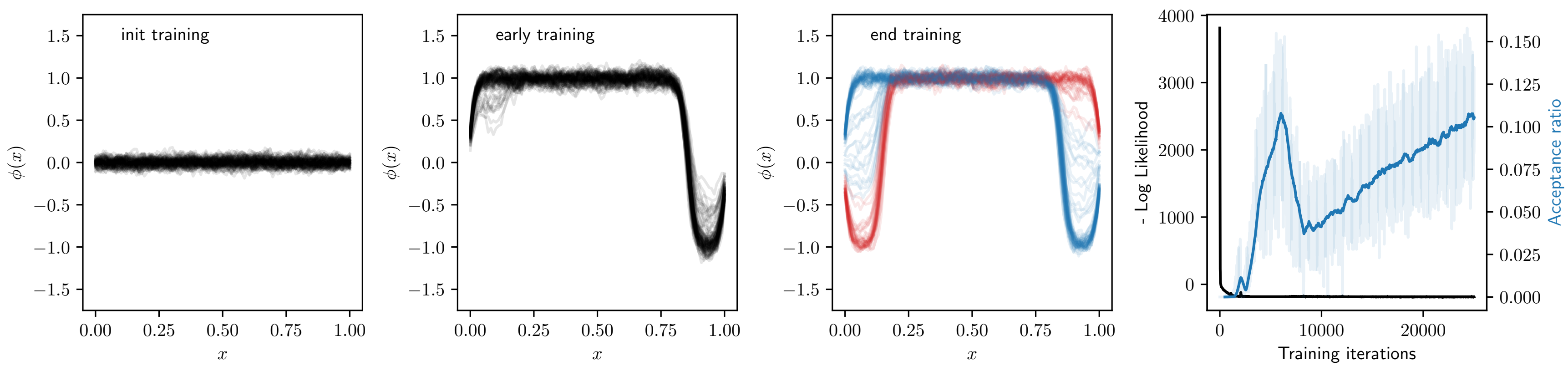
free energy difference



- ▷ Tilt distribution towards configuration with average value away from +1 or -1

$$U_{*,\boldsymbol{\lambda},\bar{\boldsymbol{\phi}}}(\phi) = \int \left(\frac{a}{2} |\nabla_x \phi|^2 + V(\phi) + \boldsymbol{\lambda} \left(\int \phi \,\mathrm{d}x' - \bar{\boldsymbol{\phi}} \right)^2 \right) \mathrm{d}x$$

  - ○ Train with $\bar{\phi} = 0.7$



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

4.4 – Leveraging physics

▷ Using symetries and invariance

- cf Danilo's talk
- cf cluster updates by Wu, Rossi & Carleo, PRR (2021)

▷ Using informed base measures

Wu, D., Rossi, R., & Carleo, G. (2021). Unbiased Monte Carlo cluster updates with autoregressive neural networks. *Physical Review Research*, *3*(4).

Outline

1. A couple of important sampling methods

   1.1 - Importance sampling

   1.2 - Metropolis-Hasting

2. Unsupervised learning / generative models

   2.1 - Latent deep generative models

   2.2 - Normalizing flows

3. Combining traditional inference method and learning

   3.1 - Variational Inference

   3.2 - Adaptive algorithms

4. Will it scale?

   4.1 - Local sampling in reparametrized space

   4.2 - Local-global sampling

   4.3 - Joining forces with annealing

   4.4 - Leveraging physics

Take aways

- ▷ Opportunities
  - VI, IS and MCMC can be powered by normalizing flows/NADE
  - IS or MCMC allows to de-bias the training model
- ▷ Challenges for scaling things up and morals
  - ML models should be joining forces with « traditional » sampling methods
  - ML models should leverage the physical knowledge about systems
- ▷ Softwares

PyTorch

Adaptive MCMCs

marylou-gabrie / **flonaco** Public

kazewong / **flowMC** Public

Sequential MCMCs

minaskar / **pocomc** Public

deepmind / **annealed_flow_transport** Public

Workshop : Machine Learning-Assisted Sampling for Scientific Computing – Applications in Physics

Invited Speakers

- David Aristoff, Colorado State University
- Peter Bolhuis, University of Amsterdam
- Freddy Bouchet, Ecole Normale Supérieure, Lyon
- Maria K. Cameron, University of Maryland
- Arnaud Doucet, University of Oxford
- Alain Durmus (ENS Paris Saclay & École Polytechnique)
- Stéphane Mallat, Ecole Normale Supérieure, Paris
- Pierre Monmarché, Pierre and Marie Curie University (UPMC)
- Jutta Rogal, New York University
- Phiala Shanahan, Massachusetts Institute of Technology
- Gabriel Stoltz, Ecole Nationale des Ponts et Chaussées, Paris
- Jonathan Weare, New York University
- Martin Weigt, Pierre and Marie Curie University (UPMC)
- Wei Zhang, Zuse Institute Berlin

Dates of conference open to all :

- 3-4 October 2022
- Registration deadline : 16 September 2022

Conference registration

Join us in Paris
or online!

Organizers

- Valentin de Bortoli, Ecole Normale Supérieure, Paris
- Marylou Gabrié, Ecole Polytechnique, Paris
- Tony Lelièvre, Ecole Nationale des Ponts et Chaussées, Paris

Open opportunities for phds or postdocs!

École Polytechnique, (almost Paris)