DeepMind

Normalizing flows, Diffusion and Annealed Importance Sampling

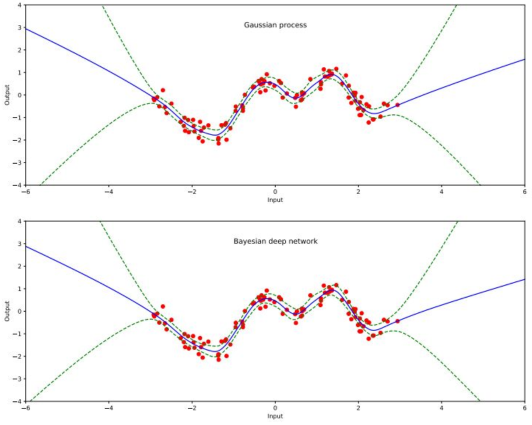
Alex Matthews
Hammer and Nails Conference.

3/11/2023

About me

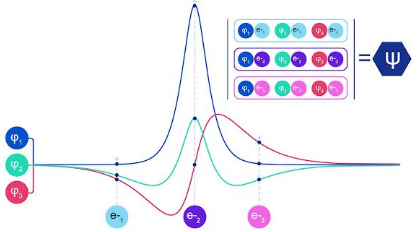
4 + 4 + 2 years

5 years

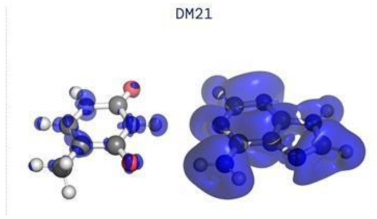


Matthews et al. 2018

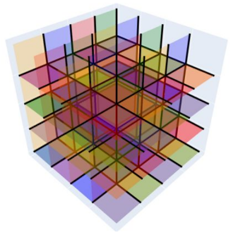
FermiNet



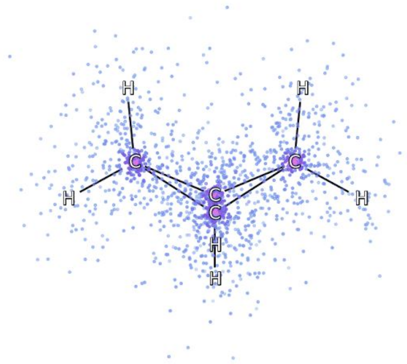
Density Functionals



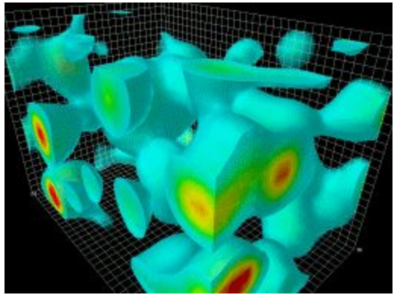
Lattice QCD



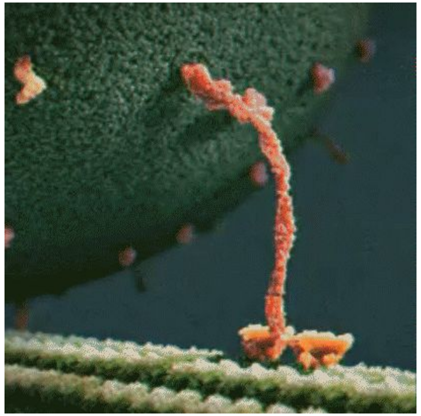
Monte Carlo is ubiquitous in the natural sciences



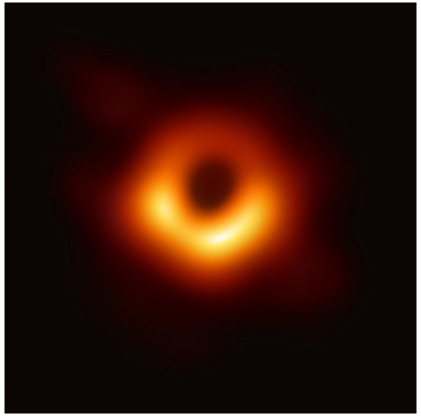
Quantum Monte Carlo



Lattice QCD



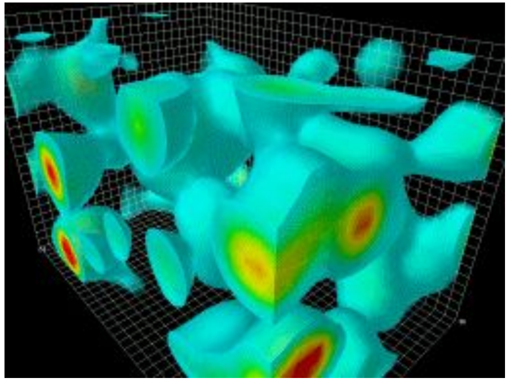
Protein physics



Black hole astronomy

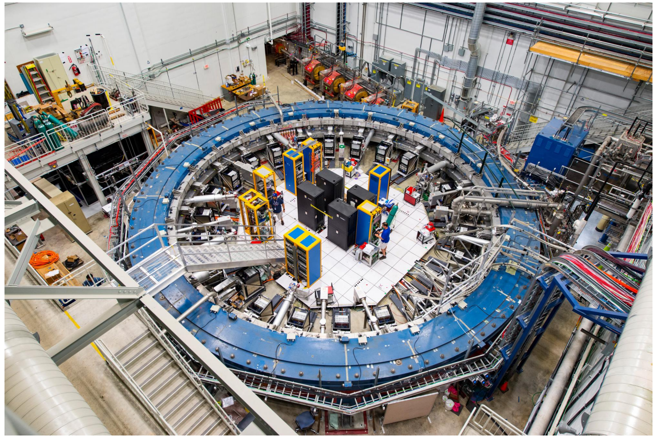
Black hole image: Event Horizon Telescope

Long term motivating example: the Muon g-2 experiment



→ ? ←

New physics



Lattice QCD is a dominant error term in standard model background.

Massive amounts of supercomputer time.

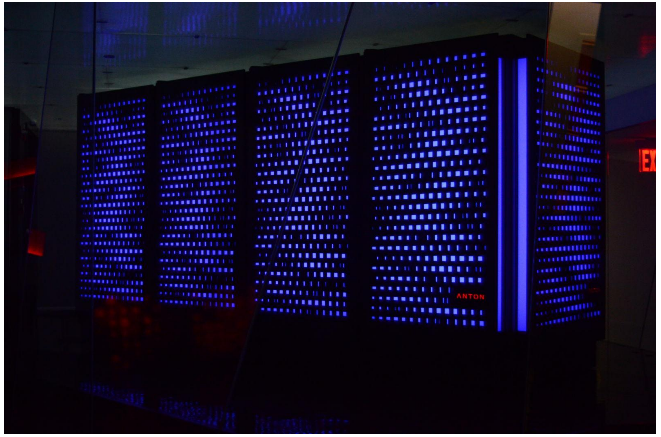
Calculate the muon magnetic moment to astonishing precision.

Animation credit: Derek B. Leinweber

Protein physics

Real protein folding is a complex dynamical process.

Many proteins do not have a single structure.



Anton supercomputer

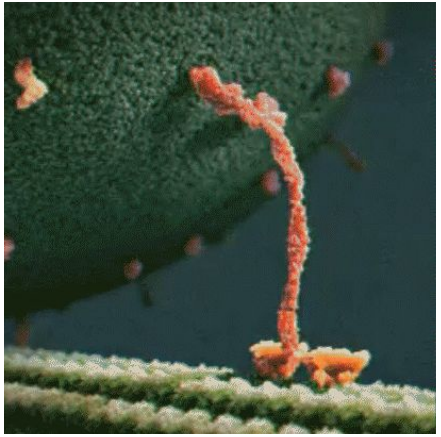


Image credits: D. E. Shaw Research

Two fundamental challenges

1) Computing normalizing constants of unnormalized distributions $\pi(\mathrm{x}) = \frac{\gamma(\mathrm{x})}{\mathrm{Z}}$.

$$Z = \int \gamma(\mathrm{x}) d\,\mathrm{x}$$

2) Computing expectations under unnormalized distributions.

$$\mathbb{E}_{\pi}[\mathrm{f}] = \frac{\int f(\mathrm{x})\,\gamma(\mathrm{x}) d\,\mathrm{x}}{\int \gamma(u) du}$$

Mapping to Boltzmann Distribution

Inverse temperature

$$\gamma_\beta(\mathrm{x}) = \exp\{-\beta E(\mathrm{x})\}$$

Energy

Partition function Z_β

Mapping to Bayesian inference

$$\overset{\text{Posterior}}{p(\mathrm{x}|y)} = \frac{\overset{\text{Likelihood}}{p(y|\mathrm{x})}\,\overset{\text{Prior}}{p(\mathrm{x})}}{p(y) \xleftarrow{} \text{Model evidence/ Marginal likelihood}}$$

Set $\gamma(\mathrm{x}) = p(y|\mathrm{x})p(\mathrm{x})$ and $Z = p(y)$

Plan for the rest of the talk

$$\begin{tikzcd}
 & \text{1) Annealed Importance Sampling} \arrow[dl] \arrow[dr] & \\
\text{2) Adding normalizing flows} & & \text{3) Connections to diffusion models}
\end{tikzcd}$$

4) Outlook for applications

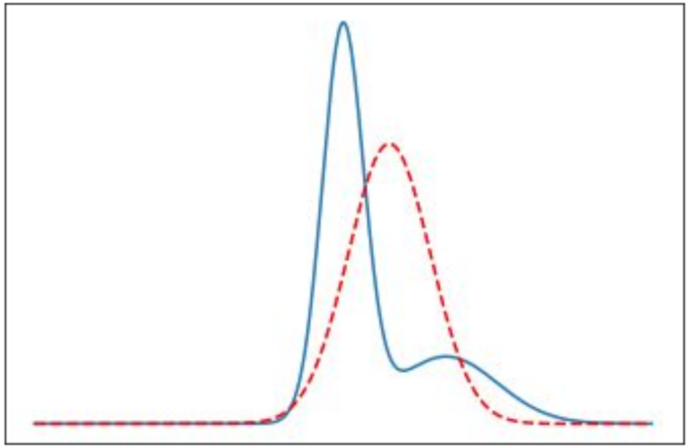
1) Annealed Importance Sampling

Naive importance sampling

Assume we can sample from q and evaluate its density.

$$Z = \int \mathrm{q}(\mathrm{x}) \frac{\gamma(\mathrm{x})}{\mathrm{q}(\mathrm{x})} d\,\mathrm{x}$$

$$\mathbb{E}_{\pi}[\mathrm{f}] = \frac{\int f(\mathrm{x})\, \mathrm{q}(\mathrm{x}) \frac{\gamma(\mathrm{x})}{\mathrm{q}(\mathrm{x})} d\,\mathrm{x}}{\int \mathrm{q}(\mathrm{x}) \frac{\gamma(\mathrm{x})}{\mathrm{q}(\mathrm{x})} d\,\mathrm{x}}$$



Naive importance sampling (2)

$$Z = \int q(x) \frac{\gamma(x)}{q(x)} dx$$

$$\mathbb{E}_{\pi}[f] = \frac{\int f(x)\, q(x) \frac{\gamma(x)}{q(x)} dx}{\int q(x) \frac{\gamma(x)}{q(x)} dx}$$

So use estimators

$$\hat{Z} = \frac{1}{N} \sum_{i=1}^{N} w_i$$

and

$$\frac{\frac{1}{N} \sum_{i=1}^{N} f(x_i)\, w_i}{\frac{1}{N} \sum_{j=1}^{N} w_j} = \sum_{i=1}^{N} W_i f(x_i)$$

where: $w_i = \frac{\gamma(x_i)}{q(x_i)}$

Naive importance sampling (3)

$$Z = \int q(x) \frac{\gamma(x)}{q(x)} dx$$

$$\mathbb{E}_\pi[f] = \frac{\int f(x)\, q(x) \frac{\gamma(x)}{q(x)} dx}{\int q(x) \frac{\gamma(x)}{q(x)} dx}$$

So use estimators

Biased but consistent

$$\hat{Z} = \frac{1}{N} \sum_{i=1}^{N} w_i \quad \text{and} \quad \frac{\frac{1}{N} \sum_{i=1}^{N} f(x_i)\, w_i}{\frac{1}{N} \sum_{j=1}^{N} w_j} = \sum_{i=1}^{N} W_i f(x_i)$$

Unbiased

where: $w_i = \frac{\gamma(x_i)}{q(x_i)}$

Importance sampling on augmented space

Easy to sample from

Proposal distribution

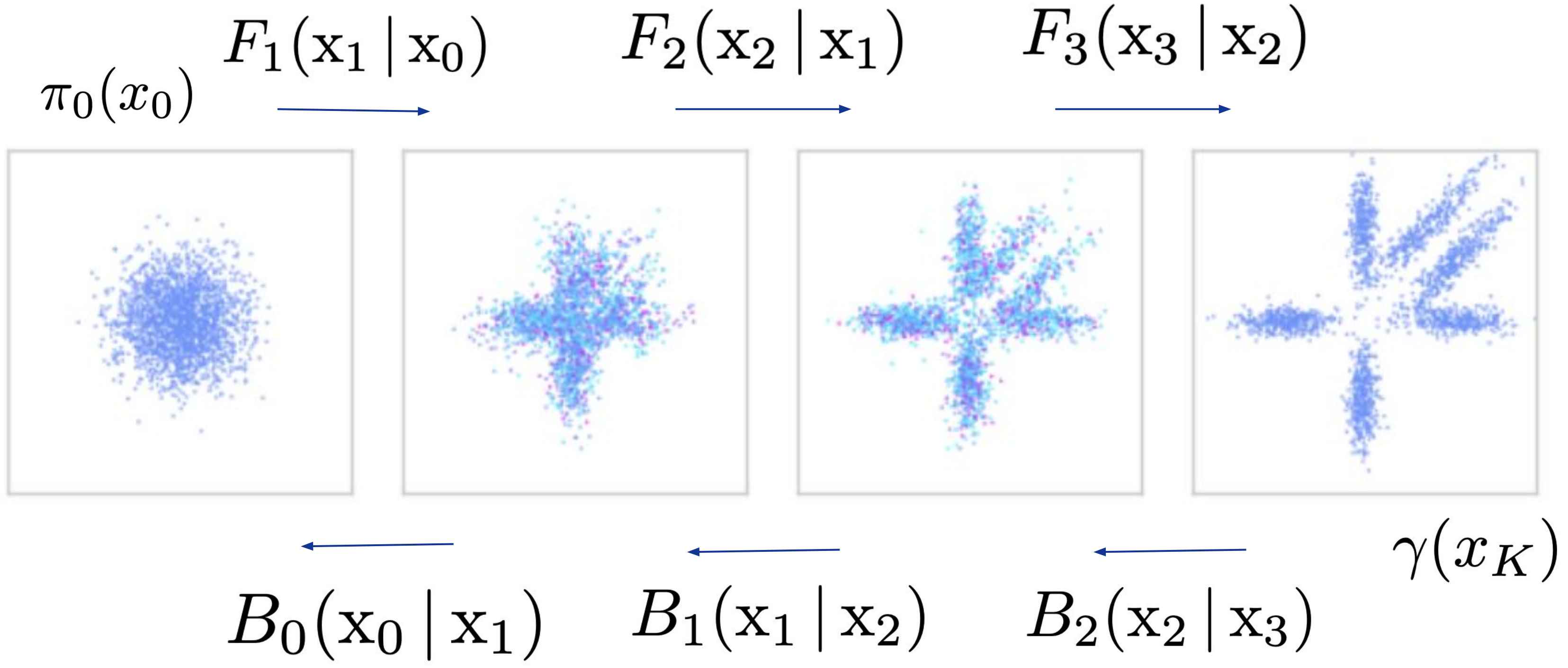
$$Q(x_{0:K}) = \pi_0(x_0) \prod_{k=1}^{K} F_k(x_k | x_{k-1}).$$

Extended target distribution

$$P(x_{0:K}) = \frac{\Gamma(x_{0:K})}{Z}, \qquad \Gamma(x_{0:K}) = \gamma(x_K) \prod_{k=0}^{K-1} B_k(x_k | x_{k+1})$$

The thing we want

Annealing


$\pi_0(x_0)$
$F_1(\mathrm{x}_1 \mid \mathrm{x}_0)$
$F_2(\mathrm{x}_2 \mid \mathrm{x}_1)$
$F_3(\mathrm{x}_3 \mid \mathrm{x}_2)$
$\gamma(x_K)$
$B_0(\mathrm{x}_0 \mid \mathrm{x}_1)$
$B_1(\mathrm{x}_1 \mid \mathrm{x}_2)$
$B_2(\mathrm{x}_2 \mid \mathrm{x}_3)$

Importance sampling on augmented space

Easy to sample from

Choose to get us closer to want we want.

Proposal distribution

$$Q(x_{0:K}) = \pi_0(x_0) \prod_{k=1}^{K} F_k(x_k|x_{k-1}).$$

Extended target distribution

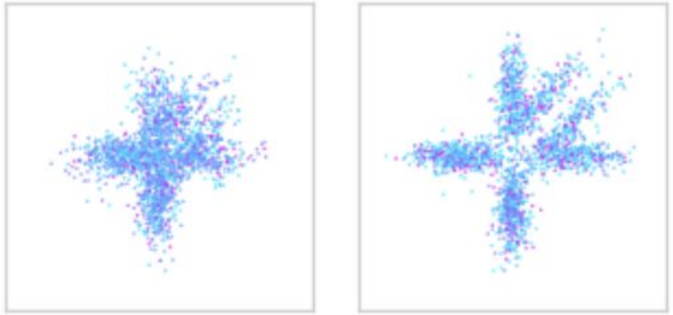
How do we choose this?

$$P(x_{0:K}) = \frac{\Gamma(x_{0:K})}{Z}, \qquad \Gamma(x_{0:K}) = \gamma(x_K) \prod_{k=0}^{K-1} B_k(x_k|x_{k+1})$$

The thing we want

Perfect reversal: Del Moral et al (2006)

$$F_{k+1}(x_{k+1}|x_k)$$

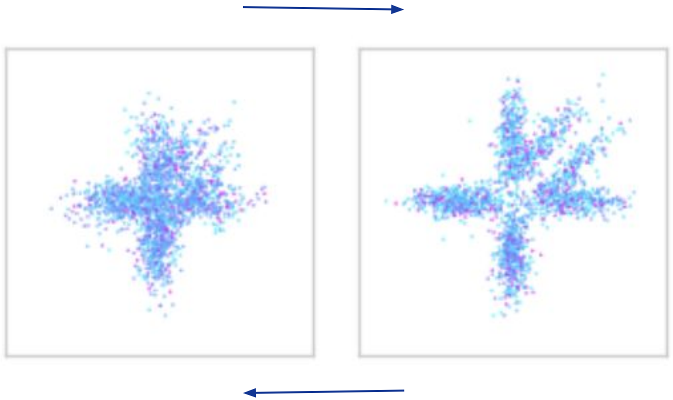


$$B_k^{\text{opt}}(x_k|x_{k+1})$$

$$B_k^{\text{opt}}(x_k|x_{k+1}) = \frac{q_k(x_k)F_{k+1}(x_{k+1}|x_k)}{q_{k+1}(x_{k+1})}$$

Perfect reversal: Del Moral et al. (2006)

$$F_{k+1}(x_{k+1}|x_k)$$



$$B_k^{\text{opt}}(x_k|x_{k+1})$$

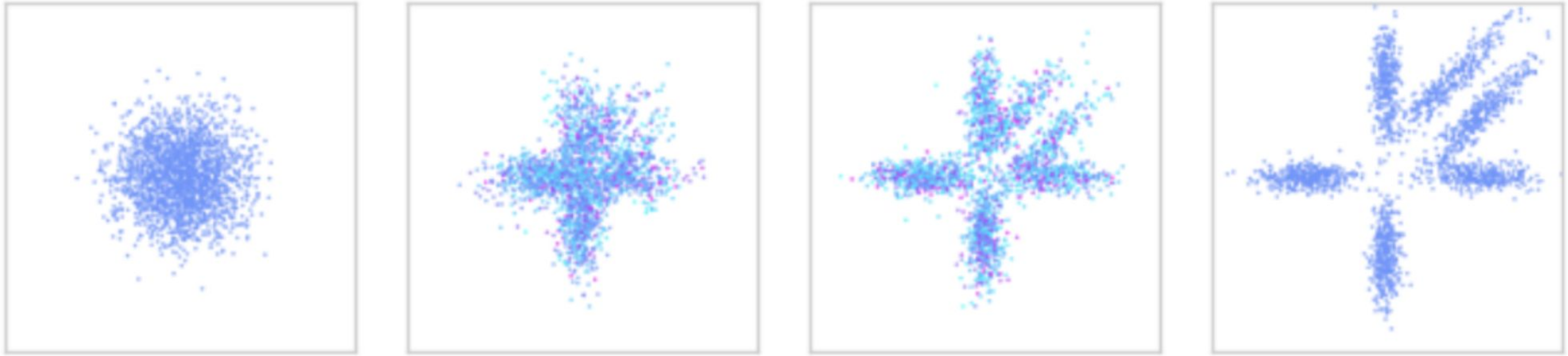
Intractable marginal distribution of proposal

$$B_k^{\text{opt}}(x_k|x_{k+1}) = \frac{q_k(x_k)F_{k+1}(x_{k+1}|x_k)}{q_{k+1}(x_{k+1})}$$

Anchor the annealing with intermediate targets

$$\pi_k(x) = \frac{\gamma_k(x)}{Z_k} = \frac{\exp(-V_k(x))}{Z_k},$$

where $Z_0 = 1$ so $\pi_0(x) = \gamma_0(x)$ and $V_k(x) = (1 - \beta_k)V_0 + \beta_k V_K$ for $0 = \beta_0 < \beta_1 < \cdots < \beta_K = 1$.



Annealed Importance Sampling: Neal (1998) / Jarzynski method (1997)

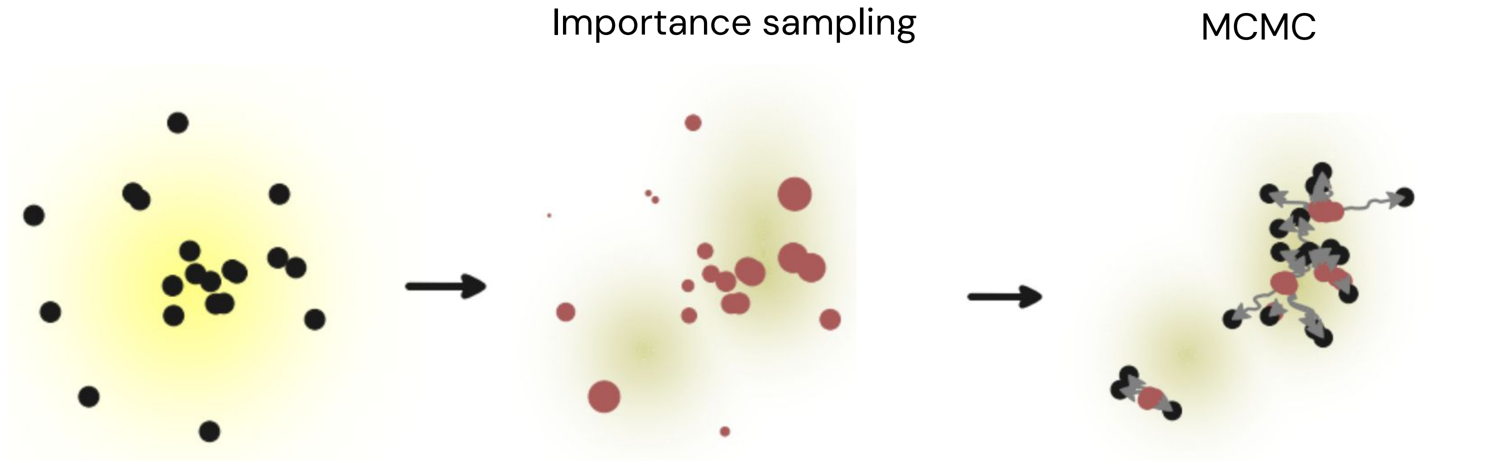
$F_{k+1}^{\text{AIS}}(x_{k+1}|x_k)$ Reversible Markov kernel with respect to $\pi_{k+1}(\mathrm{x}_{k+1})$ (or approximation thereof)

$$B_k^{\text{AIS}}(x_k|x_{k+1}) = \frac{\pi_{k+1}(x_k)}{\pi_{k+1}(x_{k+1})} F_{k+1}^{\text{AIS}}(x_{k+1}|x_k)$$

Everything cancels and it is beautiful!

$$w^{\text{AIS}}(x_{0:K}) = \prod_{k=1}^{K} \frac{\gamma_k(x_{k-1})}{\gamma_{k-1}(x_{k-1})}$$

Annealed Importance Sampling: one step



$$w_k^{\mathrm{AIS}} = w_{k-1}^{\mathrm{AIS}} \frac{\gamma_k(x_{k-1})}{\gamma_{k-1}(x_{k-1})}$$

$$x_k \sim F_k(\cdot|x_{k-1})$$

Plan for the rest of the talk

$$\begin{tikzcd}
 & \text{1) Annealed Importance Sampling} \arrow[dl] \arrow[dr] & \\
\text{2) Adding normalizing flows} & & \text{3) Connections to diffusion models}
\end{tikzcd}$$

4) Outlook for applications

2) Adding normalizing flows / CRAFT

Collaborators

Michael Arbel

Gatsby Unit -> INRIA

Arnaud Doucet

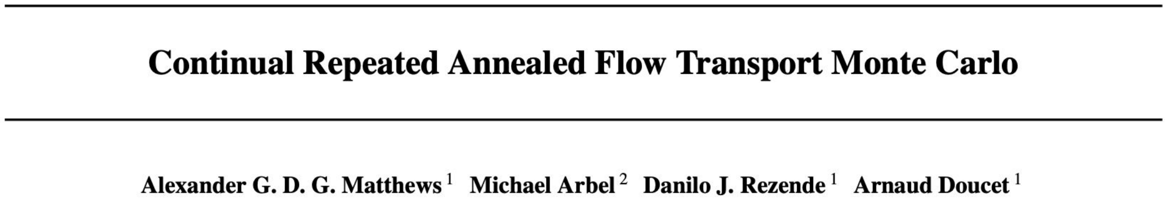
Oxford, DeepMind

Danilo J. Rezende

DeepMind

Papers and code.

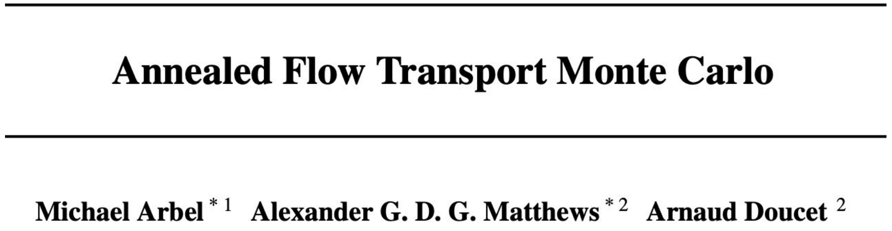
ICML 2022 paper



Continual Repeated Annealed Flow Transport Monte Carlo

Alexander G. D. G. Matthews[1] Michael Arbel[2] Danilo J. Rezende[1] Arnaud Doucet[1]

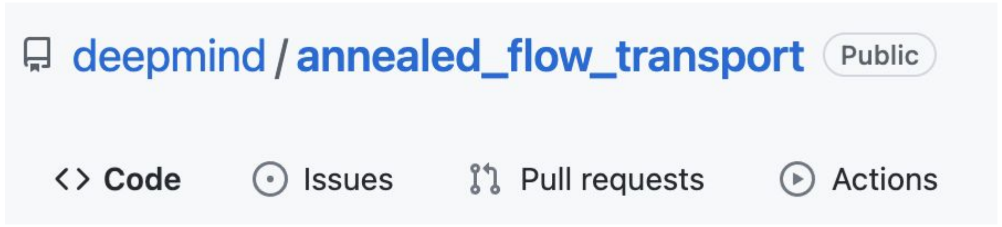
ICML 2021 paper



Annealed Flow Transport Monte Carlo

Michael Arbel[*1] Alexander G. D. G. Matthews[*2] Arnaud Doucet[2]

Open source repo on GitHub.



deepmind / annealed_flow_transport

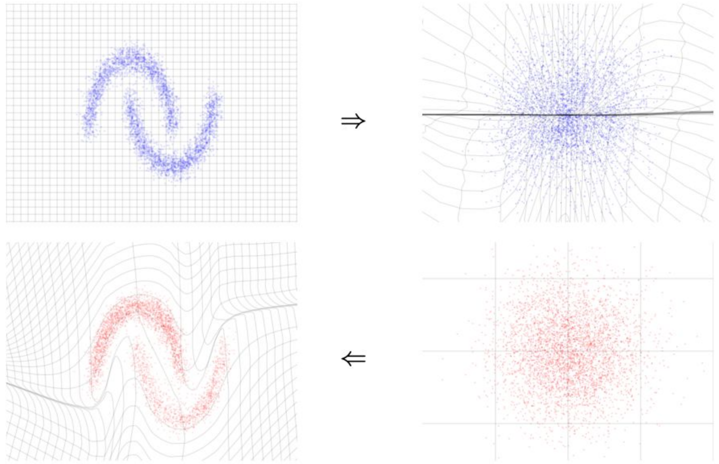
Normalizing flows

Denote these as T_k

These are diffeomorphisms that exploit the change of variables formula for tractability.

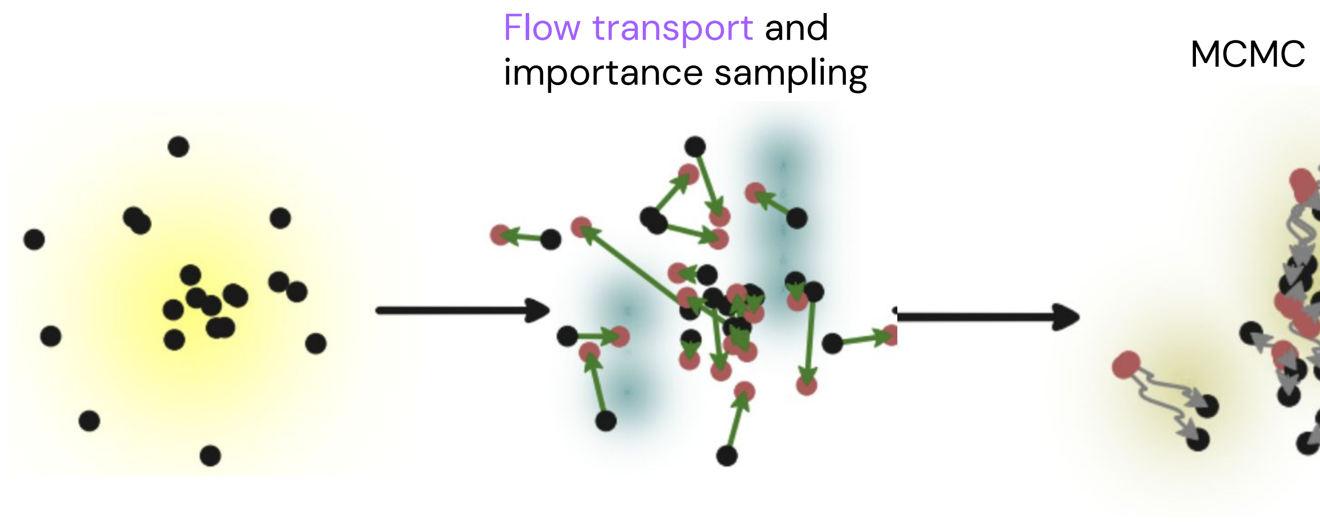
We parameterize them using neural networks and can incorporate symmetries.

Historically they have been trained by variational inference though this has some challenges.



Dinh et al. 2017

CRAFT one step with fixed normalizing flow(simplified)



$$w_k^{\text{CRAFT}} = w_{k-1}^{\text{CRAFT}} \frac{\gamma_k(T_k(x_{k-1}))}{\gamma_{k-1}(x_{k-1})} |\nabla T_k(x_{k-1})|$$

$$x_k \sim F_k(\cdot | T_k(x_{k-1}))$$

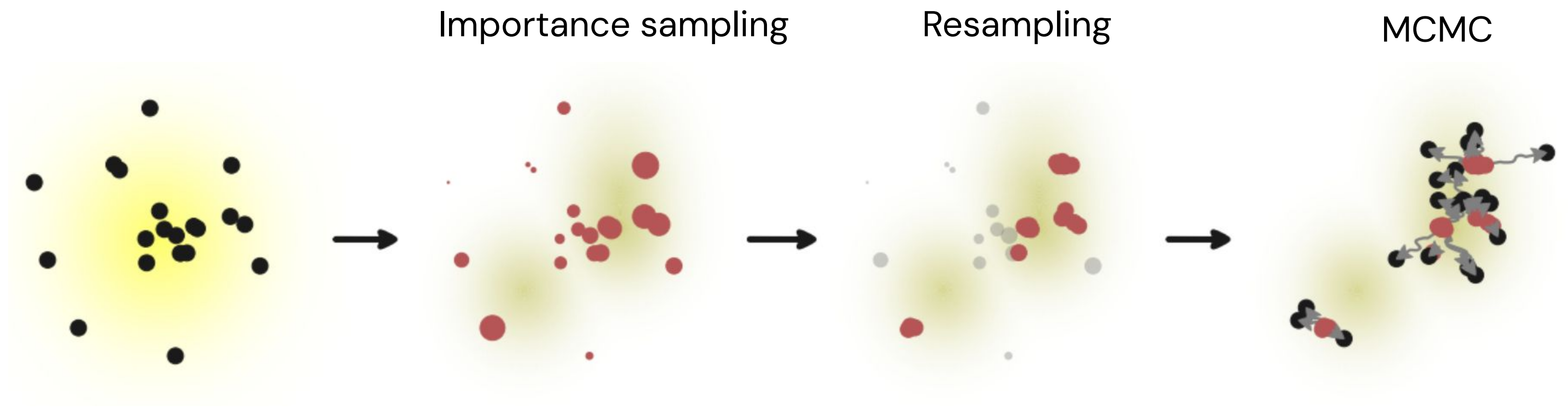
AIS has identity flow.

Optimal and only valid reversal of a flow is its inverse.

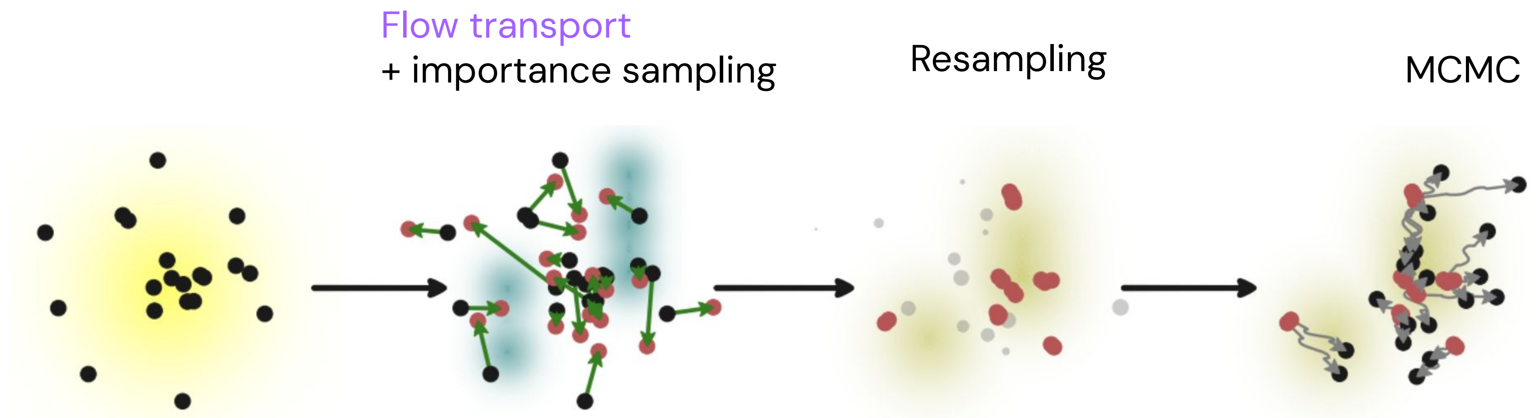
Adding resampling to AIS

Resampling allows to remove unpromising particles.

AIS --> Sequential Monte Carlo (SMC). Sampler Del Moral, Doucet and Jasra (2006)



Full CRAFT step with fixed normalizing flows.



Estimating the flow

In both cases the forward sampler is non-differentiable and not batch parallel because it includes MCMC and resampling. These components are known to help a lot so we want them.

Using multiple KLs also helps reduce mode collapse.

Zero when flow transport is perfect at each step

Previous distribution passed through a flow *T*

$$H = \sum_{l=1}^{K} \mathcal{KL}[T_{\#}^{(l)} \pi_{l-1} || \pi_l]$$

Sum over all transitions between temperatures.

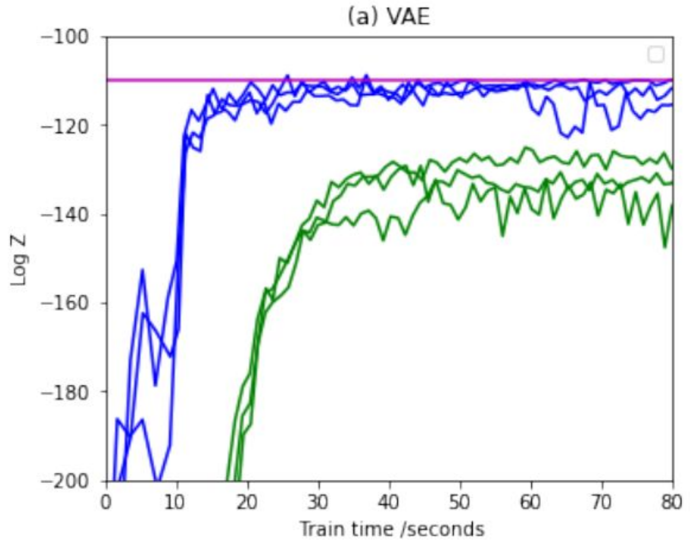
Next distribution.

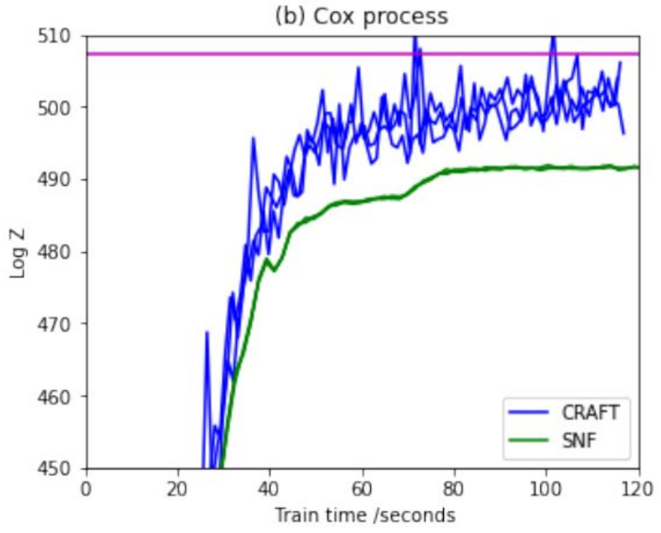
Comparison to Stochastic Normalizing Flows

Wu et al. (2020) uses standard ELBO from AIS with flows (implicit in the paper).

Runs into problems with discrete steps - there is a term ignored from the gradient.

Can also lead to mode collapse.





Experiments with 2D Euclidean ϕ^4 theory

Follow Albergo et al. 2019 use this as a testbed.

Continuum theory:

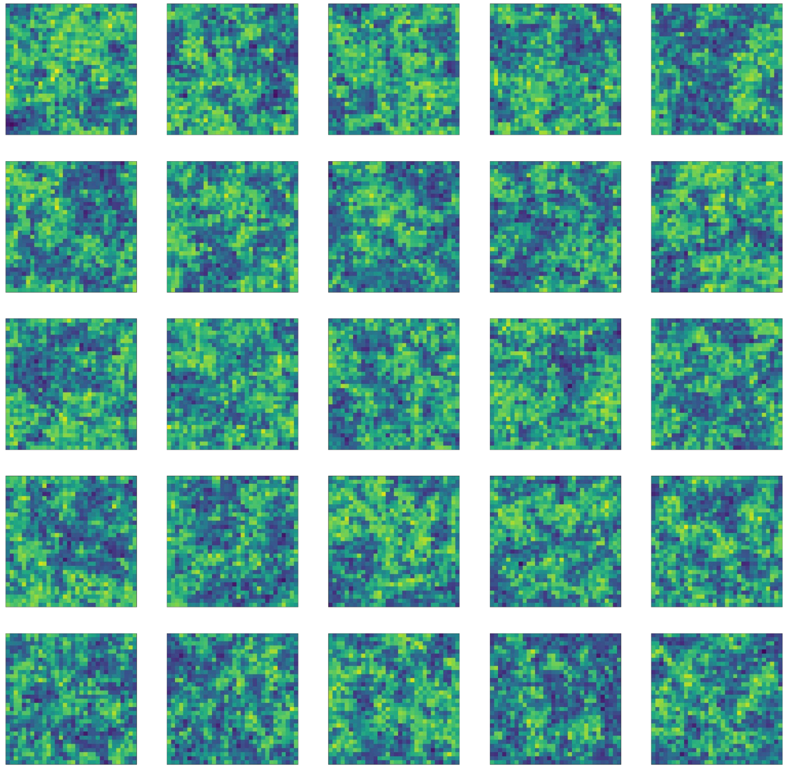
$$S_{\text{cont}}[\phi] = \int \{||\nabla\phi(x)||_2^2 + m^2\phi(x)^2 + \lambda\phi(x)^4\}\, d^2x$$

Discretize on a lattice:

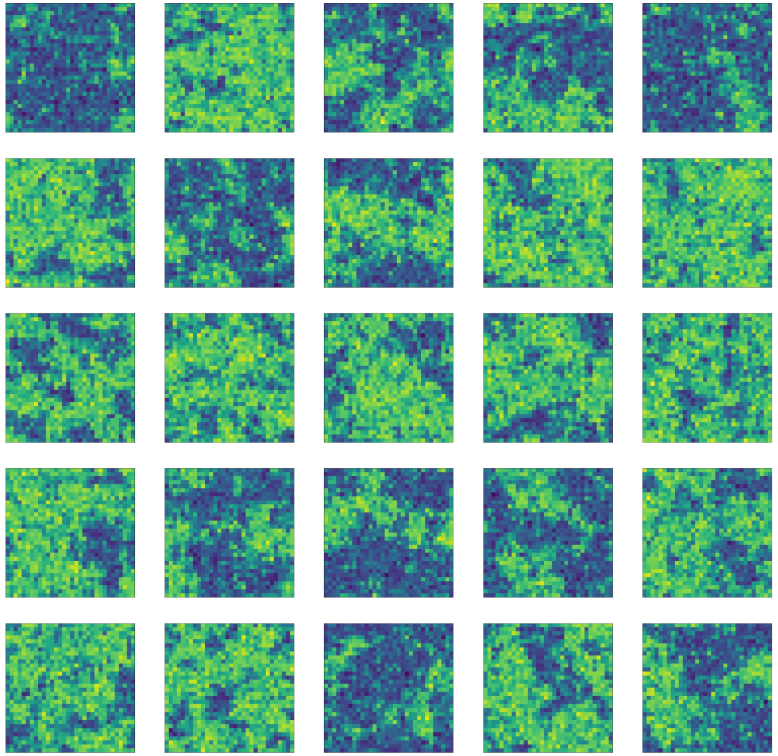
$$S_{\text{latt}}(\phi) = \sum_{\hat{x}} \left\{ \phi(\hat{x}) \sum_{\mu} [2\phi(\hat{x}) - \phi(\hat{x} + \hat{e}_\mu) - \phi(\hat{x} - \hat{e}_\mu)] + m^2\phi(\hat{x})^2 + \lambda\phi(\hat{x})^4 \right\}$$

Target probability distribution : $\dfrac{\exp\{-S_{\text{latt}}[\phi]\}}{Z}$

Samples from ϕ^4 theory:

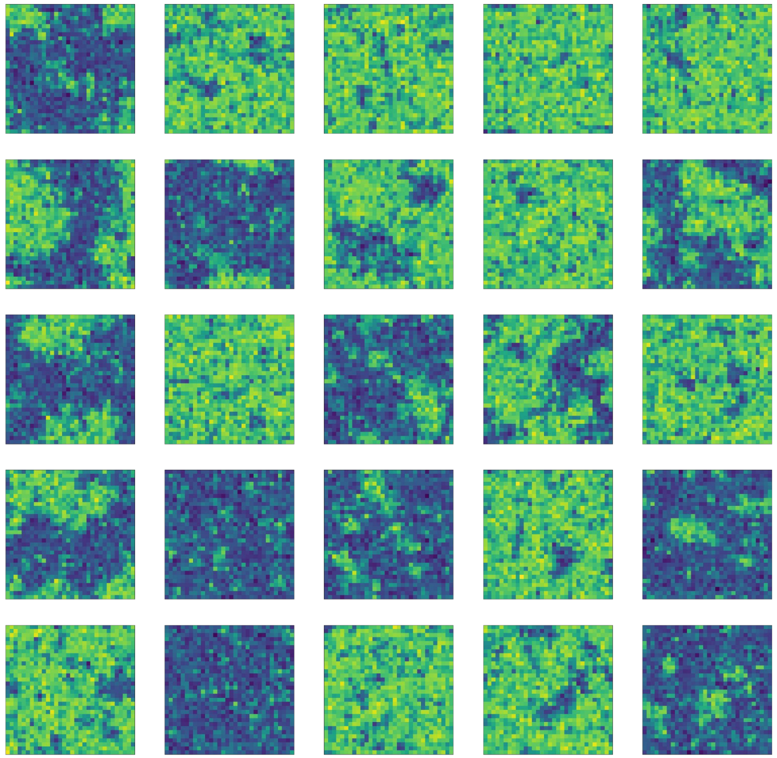


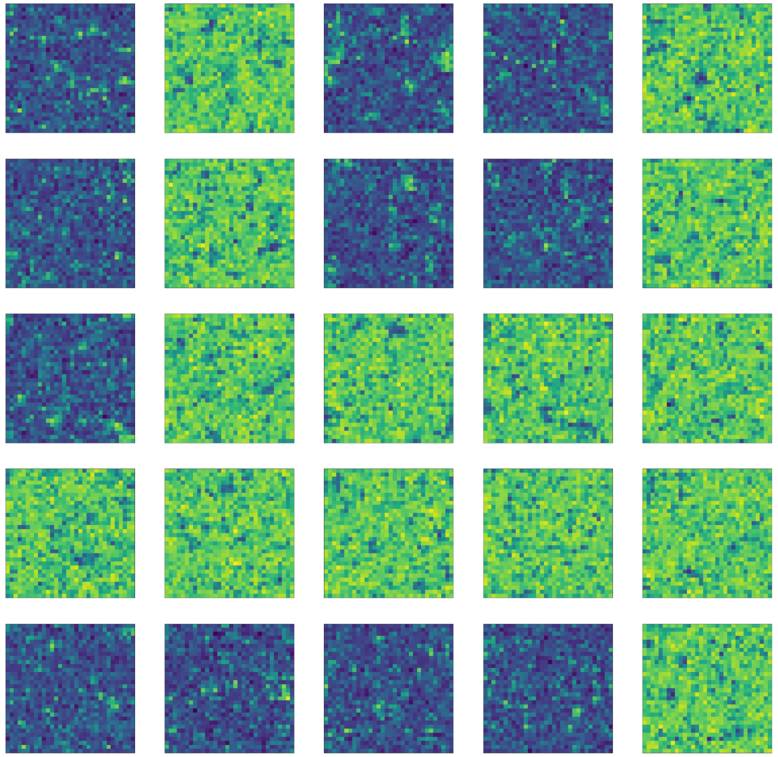
m^2 = -4
Used in original paper with 14x14 lattice



m^2 = -4.5

Samples from ϕ^4 theory:


m^2 = −4.75
Used in the work.


m^2 = −5

Choice of normalizing flow

We use the real NVP normalizing flow with checkerboard masking (left in figure).

Extract is from the original paper by Dinh, Sohl-Dickstein and Bengio ICLR 2017.

Convolution assumption in our case corresponds to translation invariance which is exactly obeyed in ϕ^4 .

This is a natural choice of flow for the problem and is mentioned in existing work.

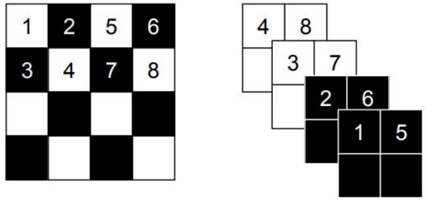


Figure 3: Masking schemes for affine coupling layers. On the left, a spatial checkerboard pattern mask. On the right, a channel-wise masking. The squeezing operation reduces the $4 \times 4 \times 1$ tensor (on the left) into a $2 \times 2 \times 4$ tensor (on the right). Before the squeezing operation, a checkerboard pattern is used for coupling layers while a channel-wise masking pattern is used afterward.

(see Figure 2(b)),

$$\begin{cases} y_{1:d} &= x_{1:d} \\ y_{d+1:D} &= x_{d+1:D} \odot \exp\big(s(x_{1:d})\big) + t(x_{1:d}) \end{cases} \tag{7}$$

$$\Leftrightarrow \begin{cases} x_{1:d} &= y_{1:d} \\ x_{d+1:D} &= \big(y_{d+1:D} - t(y_{1:d})\big) \odot \exp\big(-s(y_{1:d})\big), \end{cases} \tag{8}$$

meaning that sampling is as efficient as inference for this model. Note again that computing the inverse of the coupling layer does not require computing the inverse of s or t, so these functions can be arbitrarily complex and difficult to invert.

3.4 Masked convolution

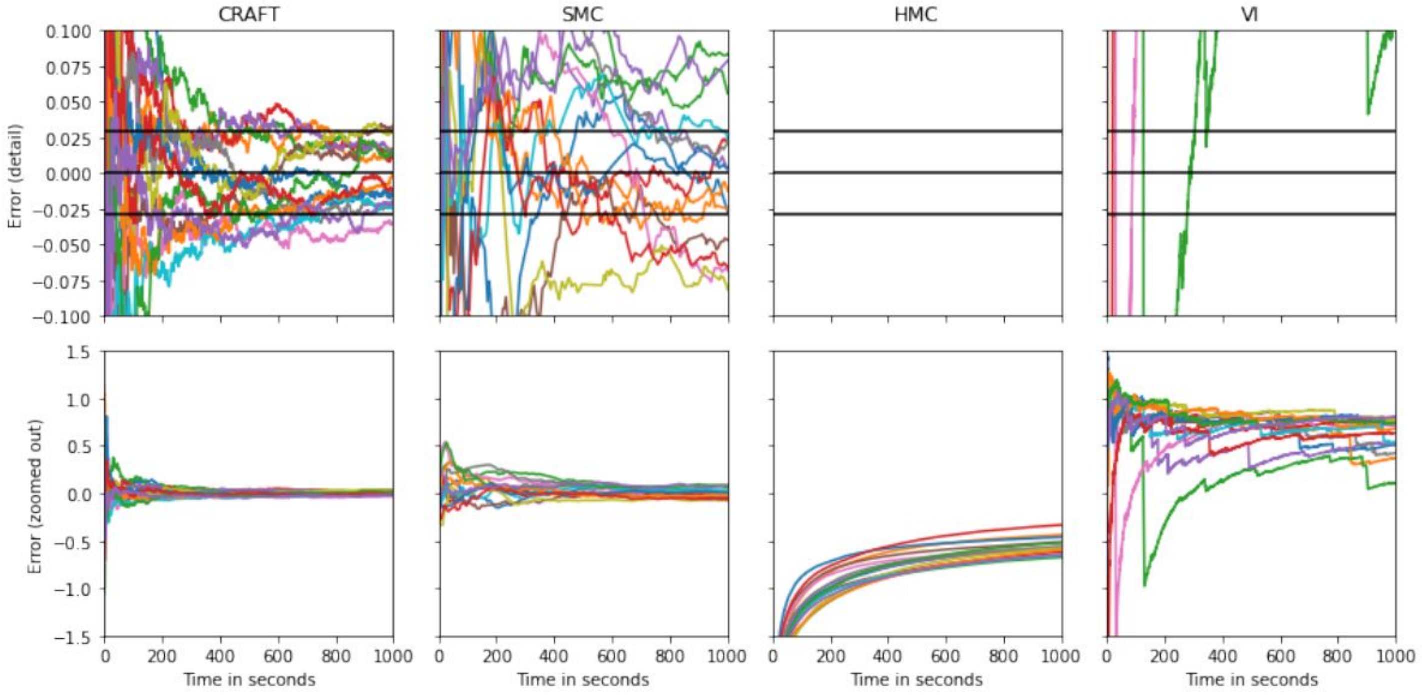
Partitioning can be implemented using a binary mask b, and using the functional form for y,

$$y = b \odot x + (1-b) \odot \Big(x \odot \exp\big(s(b \odot x)\big) + t(b \odot x)\Big). \tag{9}$$

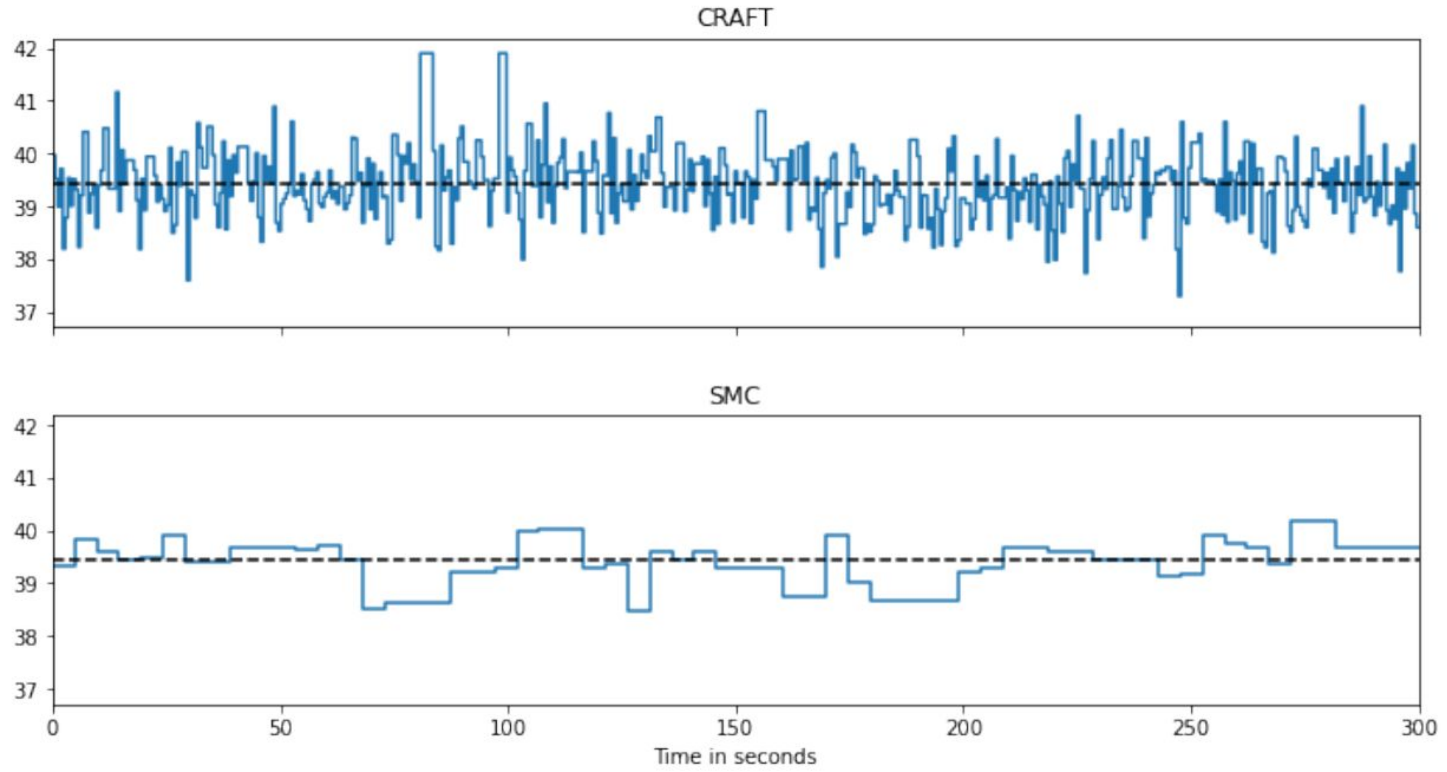
We use two partitionings that exploit the local correlation structure of images: spatial checkerboard patterns, and channel-wise masking (see Figure 3). The spatial checkerboard pattern mask has value 1 where the sum of spatial coordinates is odd, and 0 otherwise. The channel-wise mask b is 1 for the first half of the channel dimensions and 0 for the second half. For the models presented here, both $s(\cdot)$ and $t(\cdot)$ are rectified convolutional networks.

Debiasing proposals using MCMC

We can correct bias in observables for VI, SMC and CRAFT proposals using Metropolis correction.



Raw MCMC chains



Plan for the rest of the talk

$$\begin{tikzcd} & \text{1) Annealed Importance Sampling} \arrow[dl] \arrow[dr] & \\ \text{2) Adding normalizing flows} & & \text{3) Connections to diffusion models} \end{tikzcd}$$

4) Outlook for applications

3) Connections to diffusion models

Collaborators and paper

Arnaud Doucet

Oxford, DeepMind

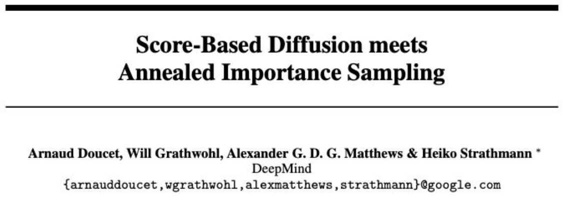
Will Grathwohl

DeepMind

Heiko Strathmann

DeepMind

NeurIPS 2022



Score-Based Diffusion meets Annealed Importance Sampling

Arnaud Doucet, Will Grathwohl, Alexander G. D. G. Matthews & Heiko Strathmann *
DeepMind
{arnauddoucet,wgrathwohl,alexmatthews,strathmann}@google.com

Reversing diffusions

Forward equation. t goes from $0 \to T$. Standard Brownian motion w.

$$x(0) \sim p_o$$

$$dx(t) = f(x,t)dt + g(t)dw$$

p_t is the marginal distribution of the forward SDE at time t.

Reverse equation. Reverse time $\tilde{t}$ goes from $0 \to T$. Different random variable $\tilde{x}$. Standard standard Brownian notion $\tilde{w}$.

$$\tilde{x}(0) \sim p_T$$

$$d\tilde{x}(\tilde{t}) = \left[g(T-\tilde{t})^2 \nabla_x \log p_{T-\tilde{t}}(\tilde{x}) - f\left(\tilde{x}, T-\tilde{t}\right)\right] d\,\tilde{t} + g\left(T-\tilde{t}\right) d\tilde{w}$$

Note that we use a different notation convention from Song et al.

Example 1: Brownian motion (plus initial jitter).

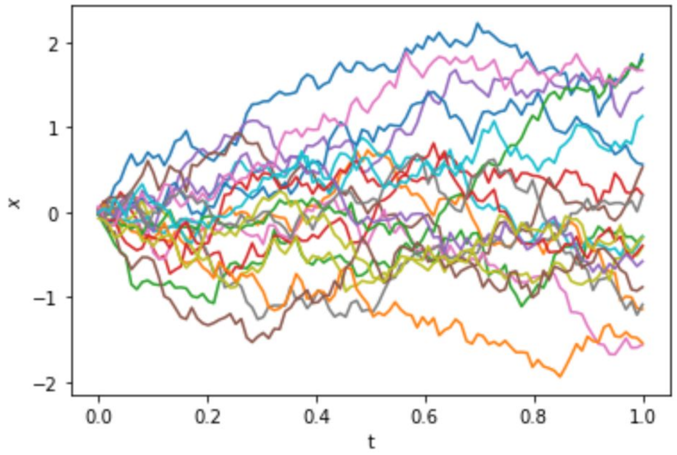
Forward equation.

$$x(0) \sim \mathcal{N}(0, 0.01)$$

$$dx(t) = dw$$

Marginal distribution.

$$p_t(x) = \mathcal{N}(x|0, t + 0.01)$$



Example 1: Brownian motion (plus initial jitter).

Forward equation.

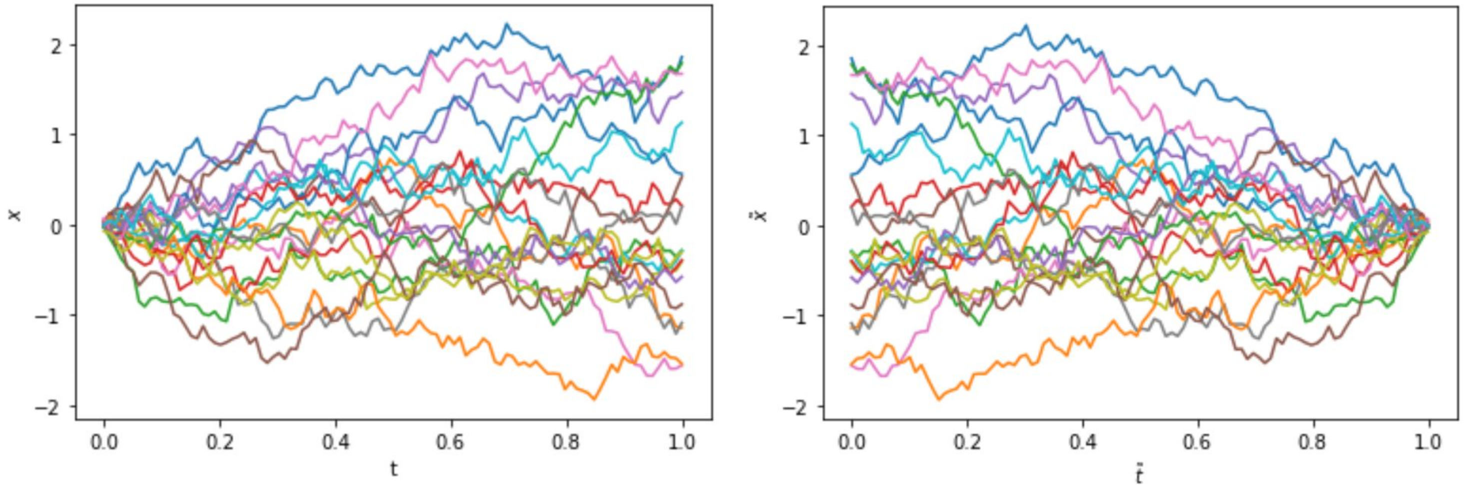
$$x(0) \sim \mathcal{N}(0, 0.01)$$
$$dx(t) = dw$$

Marginal distribution.

$$p_t(x) = \mathcal{N}(x|0, t + 0.01)$$

Reverse equation.

$$\tilde{x}(0) \sim \mathcal{N}(0, 1 + 0.01)$$
$$d\tilde{x}(\tilde{t}) = -\frac{2\tilde{x}}{(T - \tilde{t} + 0.01)^2} d\tilde{t} + d\tilde{w}$$



Example 2: Homogeneous Langevin diffusion initialized at target distribution.

Forward equation.

$$x(0) \sim p(x)$$

$$dx(t) = \nabla_x \log p(x)dt + \sqrt{2}dw$$

Marginal distribution.

$$p_t(x) = p(x)$$

Reverse equation.

$$\tilde{x}(0) \sim p(\tilde{x})$$

$$d\tilde{x}(\tilde{t}) = \nabla_{\tilde{x}} \log p(\tilde{x})d\tilde{t} + \sqrt{2}d\tilde{w}$$

Example 2: Homogeneous Langevin diffusion initialized at target distribution.

Forward equation.

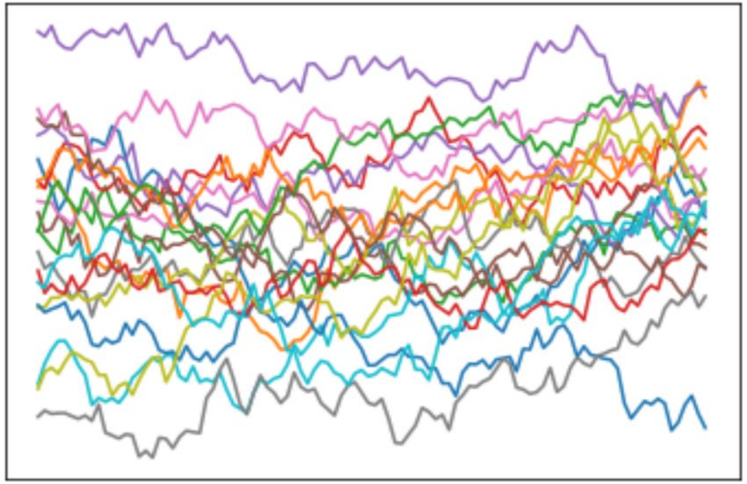
$$x(0) \sim p(x)$$
$$dx(t) = \nabla_x \log p(x)dt + \sqrt{2}dw$$

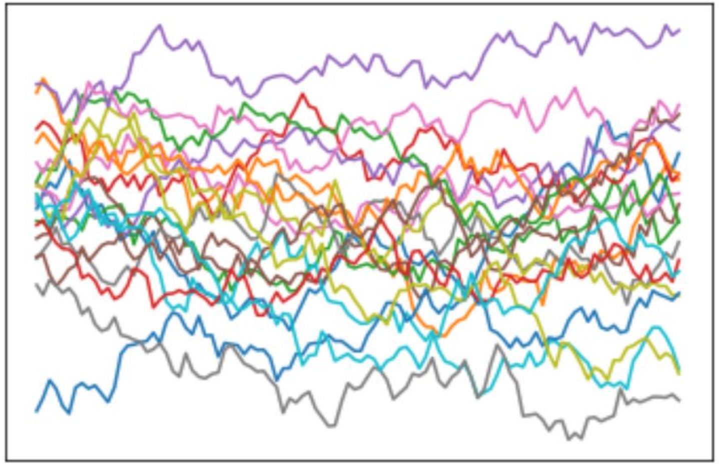
Marginal distribution.

$$p_t(x) = p(x)$$

Reverse equation.

$$\tilde{x}(0) \sim p(\tilde{x})$$
$$d\tilde{x}(\tilde{t}) = \nabla_{\tilde{x}} \log p(\tilde{x})d\tilde{t} + \sqrt{2}d\tilde{w}$$





Song et al. 2021. Continuous time diffusion from data

Also see Sohl-Dickstein et al. 2015

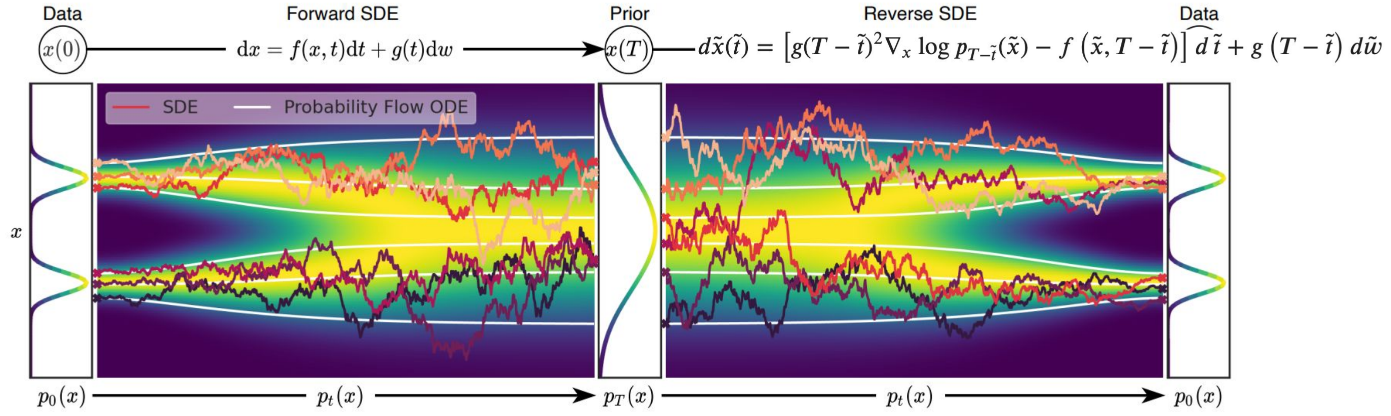
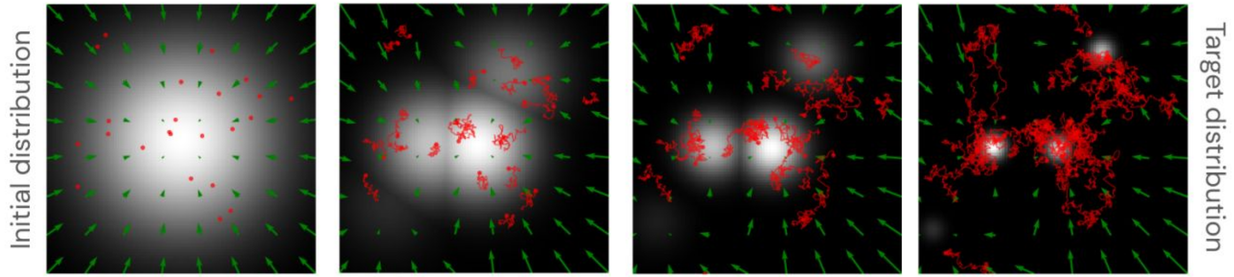

Data
Forward SDE
Prior
Reverse SDE
Data
$x(0)$
$\mathrm{d}x = f(x,t)\mathrm{d}t + g(t)\mathrm{d}w$
$x(T)$
$d\tilde{x}(\tilde{t}) = \left[g(T-\tilde{t})^2\nabla_x \log p_{T-\tilde{t}}(\tilde{x}) - f(\tilde{x}, T-\tilde{t})\right] d\tilde{t} + g(T-\tilde{t})\, d\tilde{w}$
SDE
Probability Flow ODE
x
$p_0(x)$
$p_t(x)$
$p_T(x)$
$p_t(x)$
$p_0(x)$

Figure 2: **Overview of score-based generative modeling through SDEs.** We can map data to a noise distribution (the prior) with an SDE (Section 3.1), and reverse this SDE for generative modeling (Section 3.2). We can also reverse the associated probability flow ODE (Section 4.3), which yields a deterministic process that samples from the same distribution as the SDE. Both the reverse-time SDE and probability flow ODE can be obtained by estimating the score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ (Section 3.3).

This works really well for learning from data. Some recent papers generate from energy and then learn from the samples.
But can't we use the energy inside the algorithm?

Doucet et al. 2022. Continuous time diffusion from energies

$$X_0 \sim \pi_0 \xrightarrow{\mathrm{d}X_t = \nabla \log \pi_t(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t} X_T \approx \pi$$

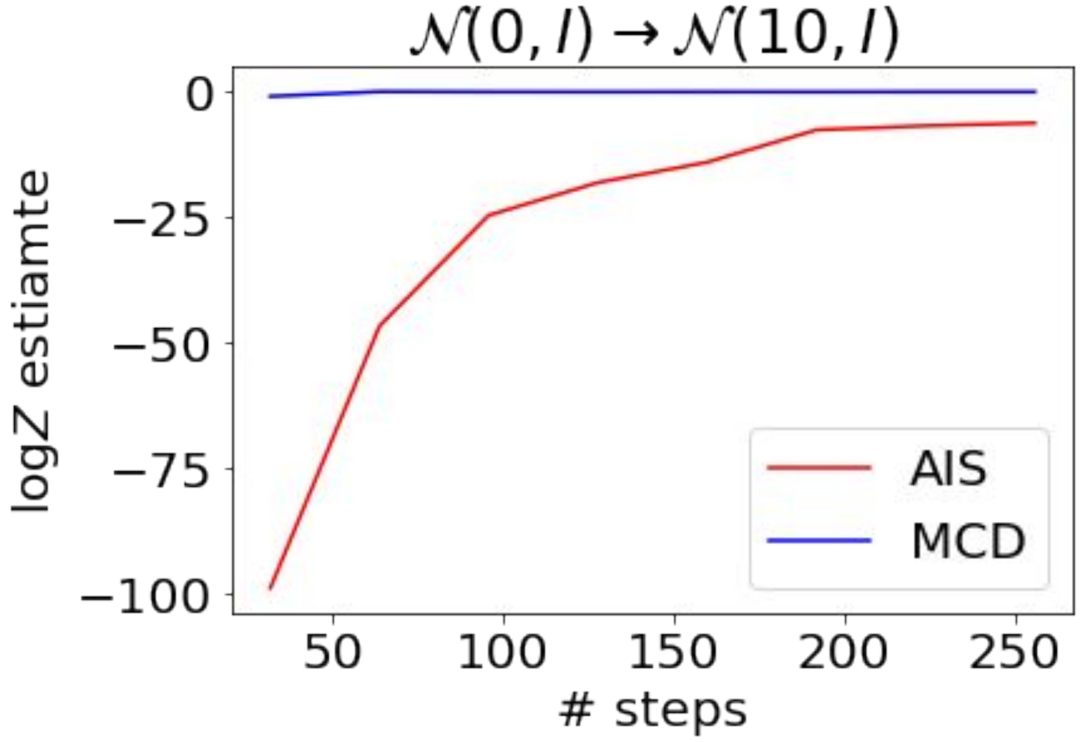


$$\bar{X}_T \approx \pi_0 \xleftarrow{\mathrm{d}\bar{X}_t = \{-\nabla \log \pi_{T-t}(\bar{X}_t) + 2\nabla \log q_{T-t}(\bar{X}_t)\}\,\mathrm{d}t + \sqrt{2}\mathrm{d}\bar{B}_t} \bar{X}_0 \sim \pi$$

Figure 1: **Top**: Samples X_t from an AIS proposal (red) obtained by sampling initially from a Gaussian at $t = 0$ and diffusing through Langevin dynamics on intermediate targets π_t (white). The intermediate marginals of the proposal, q_t, approximated by the samples are such that $q_T \approx \pi$ for a reasonably fast mixing diffusion. **Bottom**: Computing importance weights. The optimal extended target used to compute the weights is the distribution obtained by initializing $\bar{X}_0$ exactly from π and then following the reverse-time dynamics of the forward AIS proposal. This requires access to score vectors of the marginals q_t.

$$\begin{tikzcd} \text{Optimal reversal of importance sampler} \arrow[r, "\text{Continuum limit}"] & \text{Reversal of diffusions} \end{tikzcd}$$

This informs discretized algorithms



Plan for the rest of the talk

$$\begin{tikzcd}
 & \text{1) Annealed Importance Sampling} \arrow[dl] \arrow[dr] & \\
\text{2) Adding normalizing flows} & & \text{3) Connections to diffusion models} \\
 & \text{4) Outlook for applications} &
\end{tikzcd}$$

4) Outlook for applications

Practical observations

SMC/AIS is a strong baseline but can struggle in hard cases.

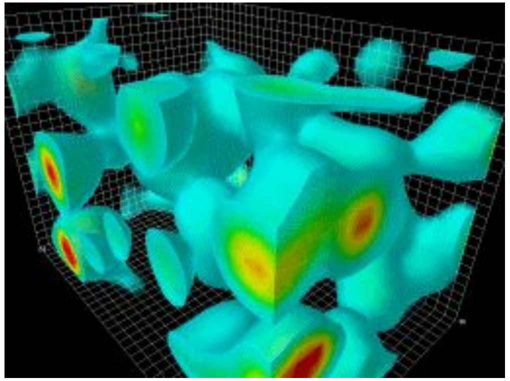
CRAFT currently outperforms MCD in most cases.

MCD may need some more methodological improvements to bridge the gap - see paper.

Think about when ML will help generally…

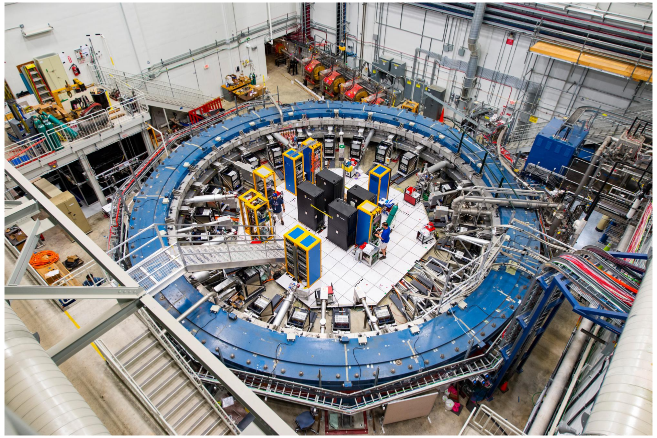
1) When classical algorithms are failing but there is knowledge ML can incorporate.
2) When amortization is important e.g for multiple related systems.
3) When very high accuracy is needed overhead of training is more likely to be worth it.

Long term motivating example: the Muon g-2 experiment



?

New physics

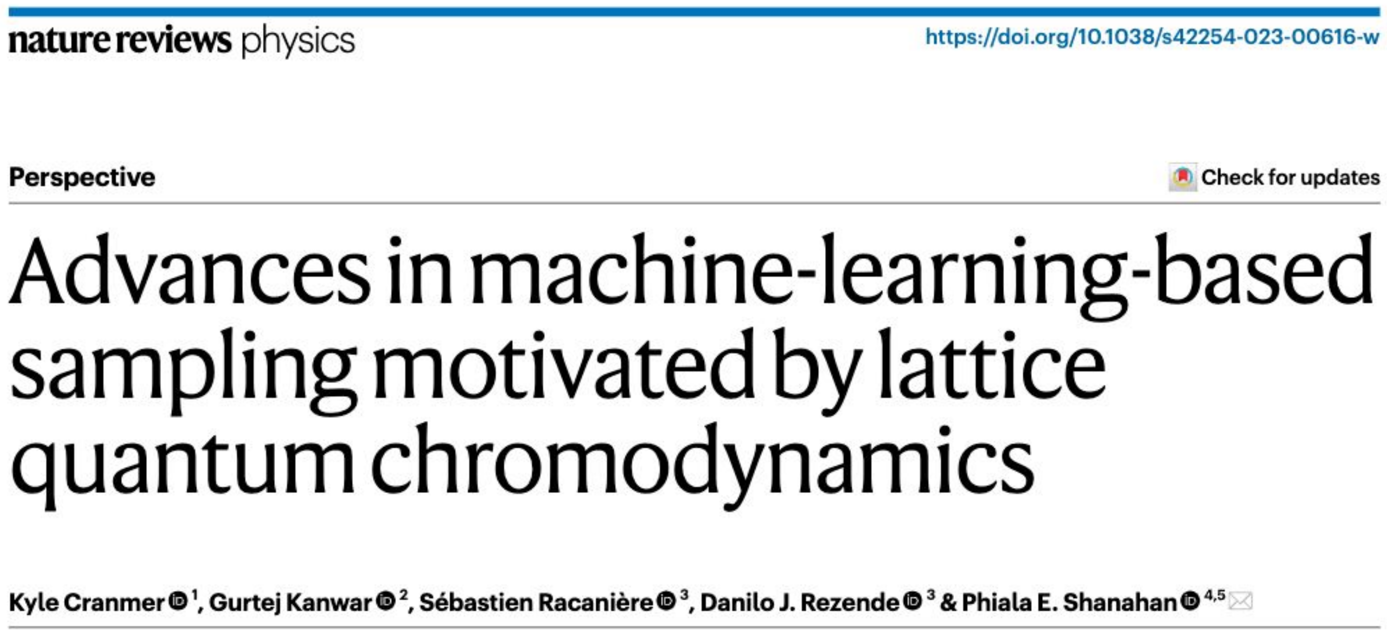


Lattice QCD is a dominant error term in standard model background.

Massive amounts of supercomputer time.

Calculate the muon magnetic moment to astonishing precision.

Animation credit: Derek B. Leinweber



nature reviews physics

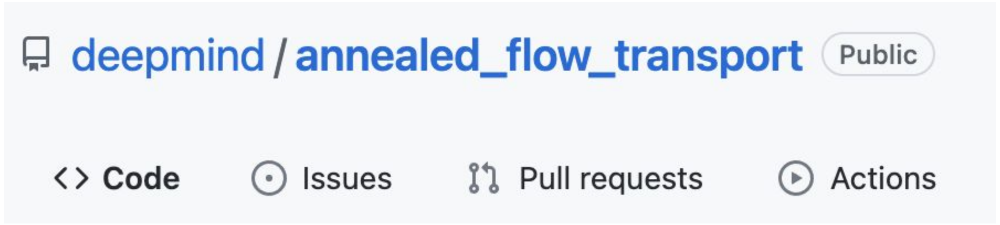
https://doi.org/10.1038/s42254-023-00616-w

Perspective

Check for updates

Advances in machine-learning-based sampling motivated by lattice quantum chromodynamics

Kyle Cranmer[1], Gurtej Kanwar[2], Sébastien Racanière[3], Danilo J. Rezende[3] & Phiala E. Shanahan[4,5]


deepmind / annealed_flow_transport Public