THE UNIVERSITY *of* EDINBURGH

Machine Learning Trivializing Maps

Joe Marsh Rossney

The University of Edinburgh, UK

August 8, 2022

a handsome tall scientist giving a presentation on machine learning and lattice field theory containing novel results

THE UNIVERSITY *of* EDINBURGH

Table of Contents

In one slide...the problem

Want to generate representative samples...

$$\{\Phi^{(1)}, \ldots, \Phi^{(N)}\}\,, \qquad \Phi \sim p(\Phi) = \frac{1}{Z} e^{-S(\Phi)} \tag{1}$$

...estimate expectation values...

$$\overline{\mathcal{O}} = \frac{1}{N} \sum_{\{\Phi\}} \mathcal{O}(\Phi)\,, \quad \mathrm{SE}_{\overline{\mathcal{O}}} = \sigma_{\mathcal{O}} \sqrt{\frac{2\tau_{\mathcal{O}}}{N}} \tag{2}$$

...and extrapolate to the continuum limit $\xi \to \infty$. But most MCMC methods suffer from **critical slowing down**

$$\tau_{\mathcal{O}} \propto \xi^z\,, \quad z \simeq 2 \tag{3}$$

In one slide...the idea

CSD occurs because proposals are correlated. Let's train a neural model to generate **independent** proposals with a high probability of acceptance.

Important requirement

For asymptotically exact sampling, the model must permit exact and efficient computation of the proposal density $q(\Phi)$!

Albergo, Kanwar, Shanahan (2019) [1904.12072]: model learns an invertible, differentiable transformation $F : \Psi \mapsto \Phi$ from a set of 'latent variables' Ψ for which independent sampling is trivial — e.g. $\Psi \sim \exp\left(-\frac{1}{2}\sum_i \Psi_i^2\right)$. Proposal density is

$$q(\Phi) = q(\Psi) \left| \frac{\partial F(\Psi)}{\partial \Psi} \right|^{-1} \tag{4}$$

Table of Contents

The model

Let $p(x) \propto e^{-S(x)}$, $q(z)$ denote the **target density** and **prior density** respectively. We seek to approximate $p(x)$ with

$$q(x) = \int \mathrm{d}z\, q(z) q(x \mid z) = q(z) \frac{q(x \mid z)}{q(z \mid x)} \,. \tag{5}$$

$q(x \mid z)$ is a neural model which we can optimise. Usually it has several layers,

$$\begin{aligned} q(x \mid z) = \int \mathrm{d}y^{(1)} \dots \mathrm{d}y^{(T-1)} q(x \mid y^{(T-1)}) q(y^{(T-1)} \mid y^{(T-2)}) \dots \\ \dots q(y^{(2)} \mid y^{(1)}) q(y^{(1)} \mid z) \,. \end{aligned} \tag{6}$$

Each $q(y^{(t+1)} \mid y^{(t)})$ represents a transition probability.

Computing the density

From (5) we can define weights $w(x) \propto p(x)/q(x)$,

$$w(x) = \exp\left(-S(x) - \log q(z) + \log \frac{q(z \mid x)}{q(x \mid z)} \right) \tag{7}$$

$$\log \frac{q(z \mid x)}{q(x \mid z)} = \sum_{t=0}^{T-1} \log \frac{q(y^{(t)} \mid y^{(t+1)})}{q(y^{(t+1)} \mid y^{(t)})} \tag{8}$$

Asymptotically exact sampling via Metropolis test:

$$\Pr(x \to x') = \min\left(1\,,\ \frac{w(x')}{w(x)}\right)\,. \tag{9}$$

Note that $w(x)$ need not be normalised: $-\log p(x) \Leftrightarrow S(x)$.

Training

The ‘reverse’ Kullbach-Leibler divergence,

$$D_{\mathrm{KL}}(\tilde{p} \| p) = \int \mathrm{d}x \, q(x) \log \frac{q(x)}{p(x)} \tag{10}$$

can be estimated using samples generated by the model

$$\hat{D}_{\mathrm{KL}}(\{x\}) = \frac{1}{N} \sum_{\{x \sim q\}} - \log w(x) + \mathrm{const.} \tag{11}$$

Training amounts to

1. Sampling from the model, $z \mapsto x$
2. Computing $\log q(x)$, $S(x)$, and hence $\log w(x)$
3. Backprop and gradient-based update $\theta \leftarrow \theta - \eta \frac{\mathrm{d}}{\mathrm{d}\theta} \hat{D}_{\mathrm{KL}}$

Deterministic flows

Most studies to date have set

$$q(y^{(t+1)} \mid y^{(t)}) = \delta\left(y^{(t+1)} - f_t(y^{(t)})\right) \tag{12}$$

where $f_t : y^{(t)} \mapsto y^{(t+1)}$ is invertible and differentiable.

$$x = F(z) = f_{T-1} \circ f_{T-2} \circ \ldots \circ f_0(z) \tag{13}$$

$$q(x) = q(F^{-1}(x)) \left| \frac{\partial F^{-1}(x)}{\partial x} \right| \tag{14}$$

Perfectly trained flow $F^{-1} : x \mapsto z$, $x \sim p(x), z \sim q(z)$ is a **trivialising map** (Lüscher [0907.5491])

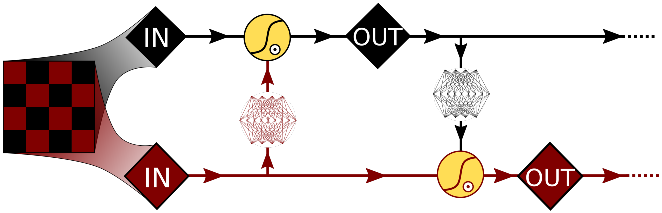
Coupling layers

Triangular Jacobian makes density calculation easy.

$$f(y_i) = \begin{cases} y_i\,, & i \in \mathbb{P} \\ g\big(y_i\,;\mathbf{n}(y_{\mathbb{P}})\big)\,, & i \in \mathbb{A} \end{cases} \qquad \mathbb{A} \cap \mathbb{P} = \emptyset \tag{15}$$

$$\log \left| \frac{\partial f(y)}{\partial y} \right| = \sum_{i \in \mathbb{A}} \log \frac{\mathrm{d}g(y_i)}{\mathrm{d}y_i}\,. \tag{16}$$

E.g. checkerboard partitioning of lattice $\mathbb{A} \cup \mathbb{P} = \Lambda$.



THE UNIVERSITY *of* EDINBURGH

Table of Contents

MIT, Deepmind and collaborators

- Albergo, Kanwar, Shanahan [1904.12072]: Proof of principle on ϕ^4 theory using affine coupling layers
- Kanwar et al. [2003.06413]: $U(1)$ equivariant flows
- Boyda et al. [2008.05456]: $SU(N)$ equivariant flows
- Albergo et al. [2106.05934]: Dynamical fermions
- Hackett et al. [2107.00734]: Optimising flow-based samplers for multi-modal distributions
- Boyda et al. [2202.05838]: ML applications for lattice field theory white paper
- Albergo et al. [2202.11712]: Schwinger model
- Abbott et al. [2207.08945]: Pseudofermions

Other contributions

- Nicoli et al. [2007.07115]: Another proof of principle on ϕ^4 theory, but using additive coupling layers
- Gabrié, Rotskoff, Vanden-Eijnden [2105.12603]: Combines flow-based moves with local updates
- Del Debbio, Marsh Rossney, Wilson [2105.12481]: Spline flows for ϕ^4, measured scaling with lattice size
- De Haan et al. [2110.02673]: Continuous normalising flows
- Foreman et al. [2112.01586]: HMC with normalising flows
- Finkerath [2201.02216]: Flow-based updates on subvolumes of the lattice
- Caselle et al. [2201.08862]: Improve CNN-based flows by interleaving stochastic layers

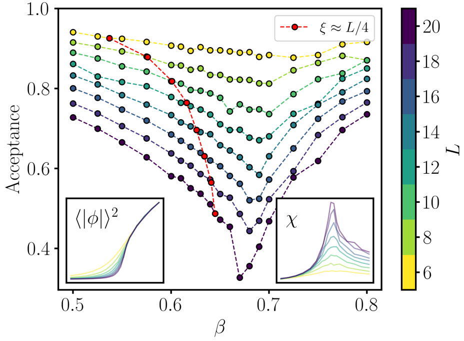
Table of Contents

THE UNIVERSITY *of* EDINBURGH

ϕ^4 Theory

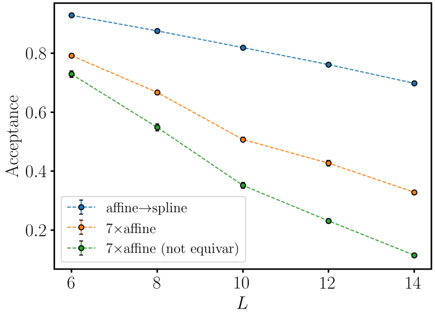
$$S(\phi) = \sum_{x\in\Lambda} \left[-\beta \sum_{\mu=1}^{2} \phi_{x+e_\mu}\phi_x + \phi_x^2 + \lambda(\phi_x^2 - 1)^2 \right]. \qquad (17)$$

Fix $\lambda = 0.5$, vary β.



Reproducing original results

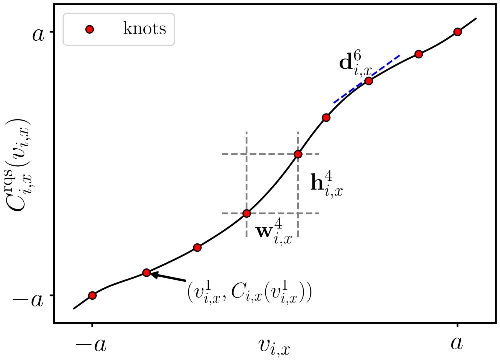
We found low acceptance rates when trying to reproduce original results of Albergo, Kanwar, Shanahan for ϕ^4 (at fixed $\xi = L/4$). But...



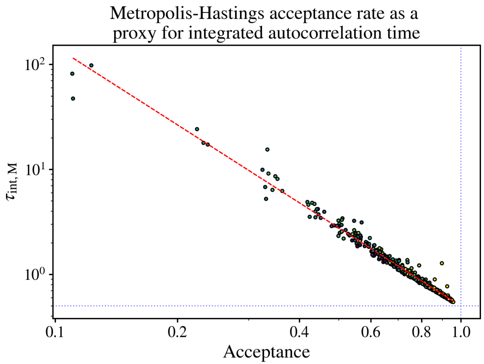
- Introducing more flexible ‘spline’ transformations led to a huge improvement over affine layers
- Enforcing $\mathbb{Z}_2$ equivariance in the affine layers helped

Rational quadratic splines

Network generates segments widths $\mathbf{w}^k_{i,x}$, heights $\mathbf{h}^k_{i,x}$ and knot derivatives $\mathbf{d}^k_{i,x}$. a & $-a$ are fixed points of the transformation.



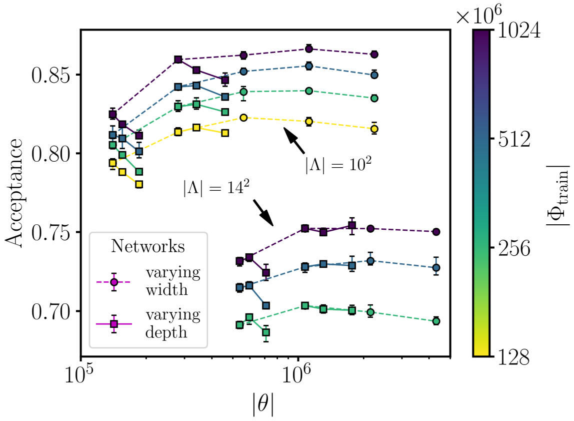
How to compare models?



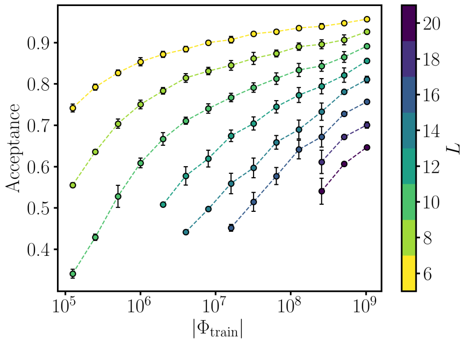
- $|\theta|$: total number of trainable parameters (network weights and biases)
- $|\Phi_{\text{train}}|$: number of configurations generated during training, i.e. batch size $\times$ number of training steps

Dependence on model size $|\theta|$

For fully-connected networks: shallow outperforms deep, but quickly diminishing returns.



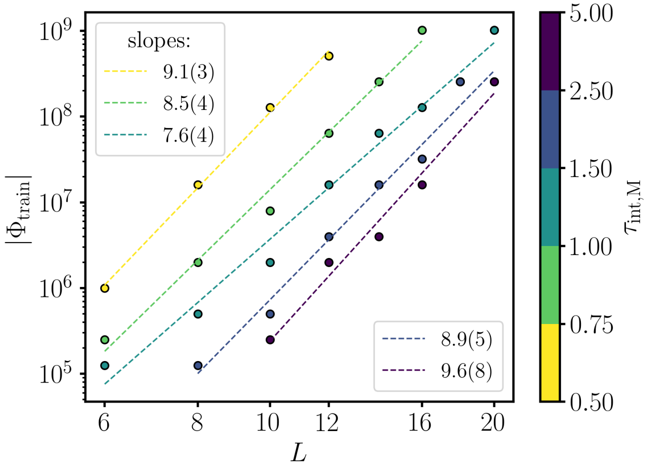
Dependence on training $|\Phi_{\text{train}}|$



Conclusion

Model expressivity is no longer the limiting factor. Acceptances are dictated by the total number of configurations exposed during training.

Attempt to quantify scaling of training cost



Caveats

These results look pretty bad, but...

- Part of the poor scaling can be attributed to fully-connected neural networks; CNNs should scale better
- We made no attempt to augment the training strategy, but it is well known that reverse-KL training becomes exponentially slow to fit tails of ill-conditioned target densities

Outstanding questions

- What is the origin of this bad scaling? A pathology of reverse KL training for ill-conditioned target densities is a candidate, but this requires verification.
- To what extent do alternative / equivariant architectures alleviate the poor scaling we observed? (Requires systematic study on larger lattices.)
- Are flow-based samplers effective at mitigating CSD of topological modes in QCD-like models (CP^{N-1})?
- Are there more efficient models than the deterministic, coupling layer based flows?
- Is it worth looking for better priors than the isotropic Gaussian or uniform distribution?

Reference

- "Efficient modeling of trivializing maps for lattice ϕ^4 theory using normalizing flows: A first look at scalability"
- arxiv:2105.12481 / Phys. Rev. D 104, 094507 (2021)
- With Prof. Luigi Del Debbio and Michael Wilson.

Table of Contents

Next steps

Currently pursuing four non-orthogonal directions:

1. Build better models
2. Test efficacy on topologically non-trivial theory – CP^{N-1}
3. Increase the lattice size (correlation length)
4. Look at other ways to use flows in sampling algorithms (see next talk!)

XY model

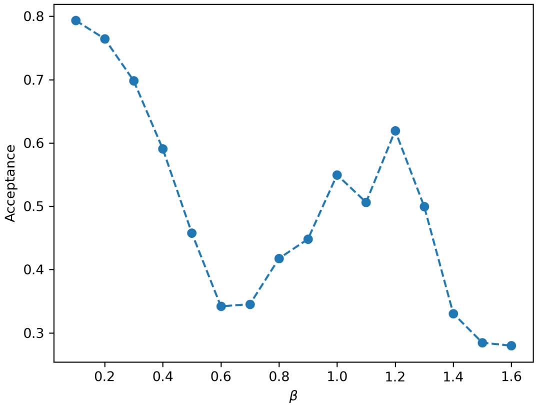
Simplest, $O(2)$-invariant action for a set of 2-spins $\sigma = (\cos\phi, \sin\phi)$ on a lattice Λ

$$
\begin{aligned}
S(\sigma) &= -\beta \sum_{x\in\Lambda} \sum_{\mu=1}^{d} \sigma_x \cdot \sigma_{x+\hat{\mu}} \\
&= -\beta \sum_{x\in\Lambda} \sum_{\mu=1}^{d} \cos\left(\phi_x - \phi_{x+\hat{\mu}}\right) \qquad (18)
\end{aligned}
$$

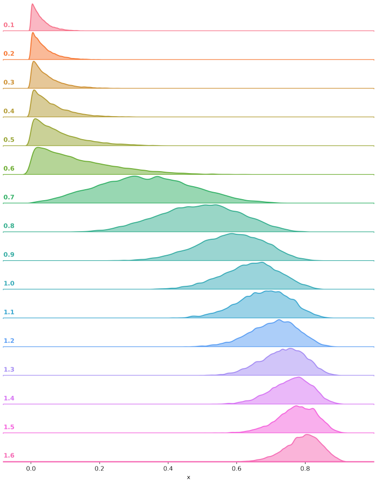
In $d = 2$ Mermin-Wagner theorem forbids SSB, but there is a Kosterlitz-Thouless transition at $\beta \approx 1.1$.

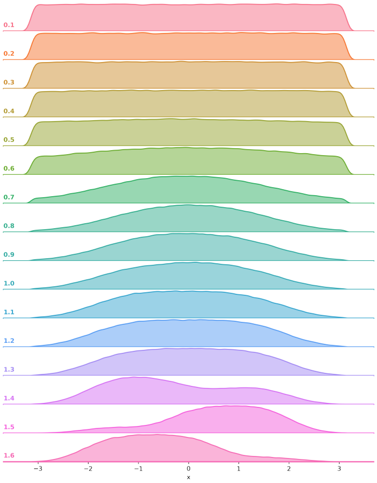
Context

Poor results (versus ϕ^4) using checkerboard partitioning & spline coupling layers...



A snag: $O(2)$ symmetry broken by the splines...





Concluding remarks

- Huge strides made in developing machinery needed to apply Normalising Flows to the sampling problem.
- Our work in [2105.12481] raises interesting questions regarding the scaling of training costs, but a large-scale systematic study with more sophisticated architectures is now needed.

Punchline

Flow-based sampling is promising but (from my perspective at least) work remains to establish which, if any, architectures are scalable!