LEARNING DEFORMATION TRAJECTORIES OF BOLTZMANN DENSITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a training objective for continuous normalizing flows that can be used in the absence of samples but in the presence of an energy function. Our method relies on either a prescribed or a learnt interpolation f_t of energy functions between the target energy f_1 and the energy function of a generalized Gaussian $f_0(x) = |x/\sigma|^p$. This then induces an interpolation of Boltzmann densities $p_t \propto e^{-f_t}$ and we aim to find a time-dependent vector field V_t that transports samples along this family of densities. Concretely, this condition can be translated to a PDE between V_t and f_t and we minimize the amount by which this PDE fails to hold. We compare this objective to the reverse KL-divergence on the Boltzmann density of the ϕ^4 theory on the circle.

1 INTRODUCTION

We consider the task of sampling from an unnormalized density $p \propto e^{-f}$ given by an energy function (unnormalized negative log-density) $f : \mathbb{R}^n \to \mathbb{R}$. In particular, there are no samples given, all we have is the ability to evaluate f for any sample candidate. A popular technique (Boyda et al., 2021; Albergo et al., 2021a;b; 2022; Abbott et al., 2022; de Haan et al., 2021; Gerdes et al., 2022; Noé et al., 2018; Köhler et al., 2020; Nicoli et al., 2020; 2021) for attacking this problem is to use a normalizing flow to parametrise a density q_θ and optimize the parameters θ to minimize the reverse KL-divergence

$$KL[q_\theta, p] = \mathbb{E}_{x\sim q_\theta}(\log q_\theta(x) - \log p(x)) = \mathbb{E}_{x\sim q_\theta}(\log q_\theta(x) + f(x)) + Z. \tag{1}$$

Unfortunately, the reverse KL-divergence is mode-seeking, making the training prone to mode-collapse (Fig. 1). To tackle this problem, several works have proposed alternative training objectives. Vaitl et al. (2022) introduce an algorithm that estimates the path gradient of the reverse KL-divergence, while Midgley et al. (2022) replace the reverse KL-divergence with the $\alpha = 2$ divergence and combine it with annealed importance sampling (Neal, 1998). We propose yet another alternative based on infinitesimal deformations of Boltzmann densities (Pfau & Rezende, 2020; Máté & Fleuret, 2022).

The contributions of this work can be summarised as follows.

- In §3 we describe our method which relies on either a prescribed or a learnt interpolation f_t of energy functions between the target energy f_1 and the energy function of a generalized Gaussian $f_0(x) = |x/\sigma|^p$. Given f_t we optimize a vector field V_t to transport samples along the family $p_t(x) \propto e^{-f_t(x)}$ of Boltzmann densities. After translating this condition to a PDE between V_t and f_t we propose to minimize the amount by which this PDE fails to hold.
- In §4 we run experiments on the ϕ^4 lattice field theory on a circle and report improvements in KL-divergence, effective sample size, and mode coverage.

Motivation Consider the following multimodal density

$$\frac{1}{3}\Big(2\,\mathcal{N}\left([-8 \quad -8], 1\right) + \mathcal{N}\left([4 \quad 4], 1\right)\Big), \tag{2}$$

where $\mathcal{N}(\mu, \sigma)$ denotes a normal density centered at μ with covariance matrix $\mathrm{diag}(\sigma^2)$. Fig. 1 shows the mode-collapse of a normalizing flow trained with the reverse KL-divergence on this target.

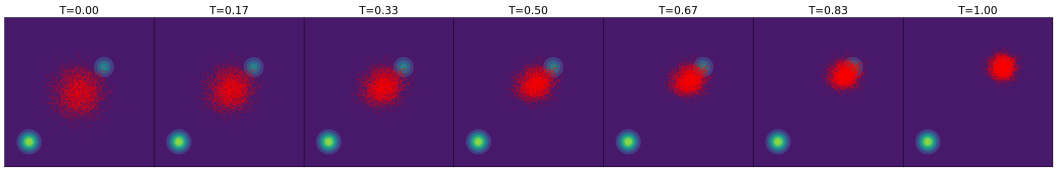

Figure 1: Mode-seeking nature of the reverse KL-divergence. The figures from left to right show how the latent gaussian is transformed by the continuous normalizing flow trained with the reverse KL objective. The green blobs denote the (unnormalized) target density, given by Eq. 2.

2 BACKGROUND

Boltzmann densities Let $f : \mathbb{R}^n \to \mathbb{R}$ be an energy function[1] with a finite normalizing constant $Z = \int e^{-f(x)} d^n x$. The function f then induces a Boltzmann density over the configurations $x \in \mathbb{R}^n$, $p(x) = \frac{1}{Z} e^{-S(x)}$. Conversely, given a probability density function $p : \mathbb{R}^n \to \mathbb{R}_{+,0}$ the corresponding energy function can be recovered up to a constant $f = -\log p - \log Z$.

The deformation equation A probability density $p_0 \propto e^{-f_0}$ and a one-parameter family of vector field V_t generates a one-parameter family of probability densities $p_t \propto e^{-f_t}$ simply by following the flow[2] determined by V_t. Conversely, given a one-parameter family of probability densities p_t, we are interested in finding the *transport field* or *deformation field* V_t. Once V_t is found, it can be used to transform samples between different members of the family p_t. Either way, the families f_t and V_t are related by

$$\partial_t f_t + \langle \nabla f_t, V_t \rangle - \nabla \cdot V_t + C_t = 0, \tag{3}$$

where C_t is a spatially constant function, i.e. it only depends on time. Moreover, if we find a triplet (f_t, V_t, C_t) that solves Eq. 4, then C_t necessarily equals $\partial_t \log Z_t = \partial_t Z_t / Z_t$ (up to an additive constant). We refer the reader to the works of Pfau & Rezende (2020, Eq. 6) and Máté & Fleuret (2022, Eq. 16) for details.

3 APPROXIMATING THE TRANSPORT FIELD

From here on, we will use the vector field V_t and the term "continuous normalizing flow" interchangably. Our goal is to sample from a target Boltzmann density $p_{target} \propto e^{-f_{target}}$ by

1. defining a family of energy functions f_t, $0 \leq t \leq 1$ interpolating between the target energy $f_1 = f_{target}$ and the energy function of a generalized Gaussian $f_0(x) = |x/\sigma|^p$,
2. finding a transport field V_t such that (f_t, V_t) "solves" the deformation equation (Eqs. 3).

If we succeed at both of these constructions, then we can obtain samples from p_{target} by sampling from $p_0 \propto e^{-|x/\sigma|^p}$ and let the samples follow the flow of V_t from $t = 0$ to $t = 1$.

Regarding the second item of the above list, an analytical expression for V_t is not easy to find if we are given a family of energy functions f_t. This would amount to solving Eq. 3, which is difficult in general. Therefore we will parametrise V_t with a neural network and train it to minimize the amount by which the pair (f_t, V_t) fails to satisfy the deformation equation. Eq. 3 is equivalent to

$$\partial_t f_t + \langle \nabla f_t, V_t \rangle - \nabla \cdot V_t + C_t = 0, \tag{4}$$

where C_t is a spatially constant function, i.e. only depends on time. In what follows, V_t^θ and C_t^θ are parametrized by neural networks and are optimized to minimize some monotonically increasing function L[3] of the pointwise *deformation error*

$$\mathcal{E}_{\theta,x,t} = \partial_t f_t(x) + \langle \nabla f_t(x), V_t^\theta(x) \rangle - \nabla \cdot V_t^\theta(x) + C_t^\theta. \tag{5}$$

[1] Depending on the field, other authors might call this *action* or *potential*.

[2] The word *flow* is not used here in the usual sense in a machine learning context, but rather in the sense of differential geometry.

[3] In our experiments we tried $L \in \{\mathcal{E} \mapsto |\mathcal{E}|, \mathcal{E} \mapsto |\mathcal{E}|^2, \mathcal{E} \mapsto |\mathcal{E}| + |\mathcal{E}|^2, \mathcal{E} \mapsto \log(1 + |\mathcal{E}|)\}$.

The expression $L(\mathcal{E}_{\theta,x,t})$ measures the incompatibility of f_t and V_t at a single (t, x) pair of coordinates, we will need to optimize some sort of integral of this pointwise error over both t and x.

THE DEFORMATION LOSS

Suppose that we have an interpolation of energy functions f_t. We propose to train V_t^θ and C_t^θ to minimize the deformation error (Eq. 5) along the trajectories of V_t^θ. Formally, let q_θ be a parametric density parametrized by a continuous normalizing V_t^θ. We update the parameters to minimalize the integral of $L(\mathcal{E})$ along the trajectories of the flow, weighting each trajectory by $w(z) = \frac{e^{-S(\gamma_1^\theta(z))}}{q_\theta(\gamma_1^\theta(z))}$,

$$\mathcal{L}(\theta) = \mathbb{E}_{z\sim\mathcal{B}}\left[w(z)\int_0^1 L\left(\mathcal{E}_{\theta,\gamma_t^\theta(z),t}\right)dt\right], \tag{6}$$

where $\mathcal{B}$ denotes the base distribution and $\gamma_\tau^\theta(z)$ is given by transporting z along the V_t^θ between 0 and τ. The standard deviation of the base is an important hyperparameter, its role can be explained as follows. Ideally, we would like to minimize the deformation error everywhere in space, but we can only evaluate it along the trajectories of V_t^θ. To provide better coverage, we can increase the standard deviation of the base density during training. In principle, it is not a requirement to train along trajectories starting from the same base as the one we're trying to match to the target. We didn't explore this possibility, in all our experiments $\mathcal{B}$ is both the distribution along which we optimize the deformation loss and also the base of the flow (i.e. the density that approximates p once pulled through the flow).

PARAMETRIZING THE INTERPOLATION

We use a neural network to parametrize the interpolation f_t as

$$f_t(x) = (1-t)f_0(x) + tf_1(x) + t(1-t)f_t^\theta(x), \qquad 0 \le t \le 1, \tag{7}$$

where f_t^θ is parametrized by a neural network. This parametrization ensures that the boundary conditions at $t \in \{0, 1\}$ are satisfied, and allows for flexibility for $0 < t < 1$. The parameters of the interpolation are trained together with the those of the flow (and those of C_t) with the objective of minimizing the deformation loss. Fig. 2 shows that this flexibility allows a flow trained with the deformation loss to capture both modes of the distribution given by Eq. 2.

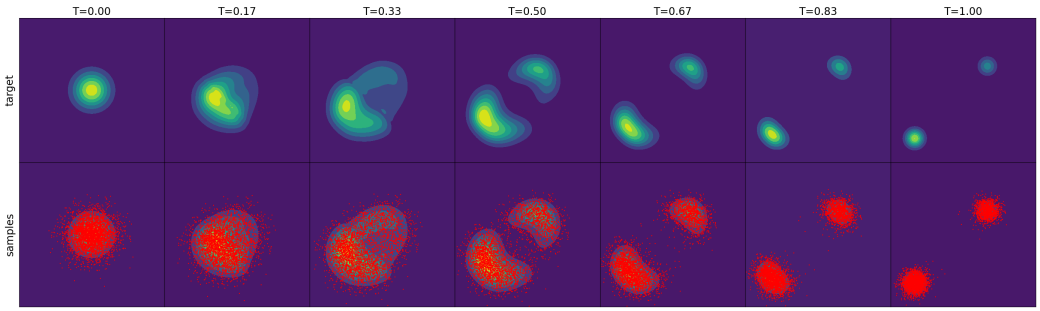

Figure 2: The same continuous normalizing flow as in Fig. 1 trained with the deformation loss using the trainable interpolation. The green blobs denote the (unnormalized) target density, a mixture of 2 Gaussians, given by Eq. 2. The top row shows how the target density evolves under the learned interpolation, while the bottom row shows how the samples from q_θ evolve along V_t as t is varied.

4 THE ϕ^4 LATTICE FIELD THEORY ON A CIRCLE

In this section we consider the ϕ^4 lattice field theory on a circle. After discretizing the circle to the cycle graph C_N with N nodes, the action associated to a state $(x_1, \ldots, x_N)$ reads

$$S(\phi_1, \ldots, \phi_N) = \sum_{i=1}^{N}\Big((\phi_i - \phi_{i+1})^2 - m\phi_i^2 + \lambda(\phi_i - \alpha)^4\Big), \tag{8}$$

where m, λ and α are numerical parameters and the subscript $i+1$ is to be understood modulo N. In all our experiments $N = 16, \lambda = 1/16, \alpha = 10^{-2}$ and $m \in \{0.25, 0.50, 0.75, 1.00\}$. In all cases, the one-dimensional Boltzmann density $e^{m\phi^2 - \lambda(\phi-\alpha)^4}$ has two modes. Their relative weight and separation is controlled by m and α. Intuitively, the second term in Eq. 8 encourages every node to follow the unnormalized density $e^{m\phi^2-(\phi-\alpha)^4}$ at every lattice site, while the first one penalizes neighbors that differ too much (i.e. jump between the modes of $e^{m\phi^2-(\phi-\alpha)^4}$).

PERFORMANCE METRICS

To quantify the results of the experiments, for each model we report a subset of the following metrics. For all runs we report the reverse KL-divergence (minus $\log Z$), $\mathbb{E}_{x\sim q_\theta}(\log q_\theta(x) - \log p(x))$ and the effective sample size,

$$ESS_r = \frac{\left(\frac{1}{N}\sum_i p(x_i)/q_\theta(x_i)\right)^2}{\frac{1}{N}\sum_i (p(x_i)/q_\theta(x_i))^2}, \qquad x_i \sim q_\theta, \tag{9}$$

where N is the number of samples. These metrics capture how good a fit q_θ is for p, but only in those regions from which samples are available. They are therefore insensitive to mode collapse. To compensate for this, we compute the Hausdorff distance between the means of the modes $M = \{m_1, ..., m_k\}$ and a batch of N samples $X = \{x_1, ...x_N\}$ from the model,

$$H(M, X) = \max_{m\in M} \min_{x\in X} \sqrt{||m - x||^2}. \tag{10}$$

In the case of Gaussian mixtures, the means M are the means of mixture components whereas in the case of the ϕ^4 theory, the means are $(\phi_1, \phi_2, ..., \phi_L) = (a, a, ..., a)$ and $(\phi_1, \phi_2, ..., \phi_L) = (b, b, ..., b)$ where a and b are the two local minima of $-m\phi^2 + \lambda\phi^4$. The Hausdorff distance is a good metric for measuring mode coverage but is insensitive to the shape of the distributions.

EXPERIMENTS

We compare the deformation loss with the trainable interpolation to the reverse KL objective. The quantitative results are summarized in Table 1.

	$m = 0.25$		$m = 0.5$	
	$KL(q_\theta, p)$	Deformation Loss	$KL(q_\theta, p)$	Deformation Loss
Hausdorff ↓	$1.15 \pm .095$	$1.15 \pm .101$	$0.87 \pm .032$	$0.98 \pm .023$
Rev. KL ↓	$-9.78 \pm .001$	$-9.78 \pm .001$	$-18.7 \pm .018$	$-18.8 \pm .001$
Rev. ESS ↑	$0.98 \pm .000$	$0.99 \pm .000$	$0.70 \pm .209$	$0.99 \pm .001$
	$m = 0.75$		$m = 1.00$	
	$KL(q_\theta, p)$	Deformation Loss	$KL(q_\theta, p)$	Deformation Loss
Hausdorff ↓	$16.7 \pm .160$	$\mathbf{0.84} \pm .022$	$20.1 \pm .190$	$\mathbf{0.77} \pm .042$
Rev. KL ↓	$-36.4 \pm .529$	$\mathbf{-37.0} \pm .001$	$-63.1 \pm .844$	$\mathbf{-63.9} \pm .001$
Rev. ESS ↑	$0.97 \pm .008$	$\mathbf{0.99} \pm .006$	$0.94 \pm .042$	$\mathbf{0.99} \pm .002$

Table 1: Results of training the same flow with two different objectives: the reverse KL-divergence and the deformation loss with the trainable interpolation. Mean and standard deviation values over 3 seeds are reported.

5 CONCLUSION

We introduced an alternative training objective of continuous normalizing flows that uses an interpolation of energy functions. We've demonstrated empirically that the proposed objective outperforms the reverse KL-divergence when the target density has multiple modes.

REFERENCES

Ryan Abbott, Michael S. Albergo, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, Gurtej Kanwar, Sébastien Racanière, Danilo J. Rezende, Fernando Romero-López, Phiala E. Shanahan, Betsy Tian, and Julian M. Urban. Gauge-equivariant flow models for sampling in lattice field theories with pseudofermions, 2022. URL https://arxiv.org/abs/2207.08945.

Michael S. Albergo, Denis Boyda, Daniel C. Hackett, Gurtej Kanwar, Kyle Cranmer, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E. Shanahan. Introduction to normalizing flows for lattice field theory, 2021a. URL https://arxiv.org/abs/2101.08176.

Michael S. Albergo, Gurtej Kanwar, Sé bastien Racanière, Danilo J. Rezende, Julian M. Urban, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, and Phiala E. Shanahan. Flow-based sampling for fermionic lattice field theories. *Physical Review D*, 104(11), dec 2021b. doi: 10.1103/physrevd.104.114507. URL https://doi.org/10.1103%2Fphysrevd.104.114507.

Michael S. Albergo, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, Gurtej Kanwar, Sébastien Racanière, Danilo J. Rezende, Fernando Romero-López, Phiala E. Shanahan, and Julian M. Urban. Flow-based sampling in the lattice schwinger model at criticality, 2022. URL https://arxiv.org/abs/2202.11712.

Denis Boyda, Gurtej Kanwar, Sébastien Racanière, Danilo Jimenez Rezende, Michael S. Albergo, Kyle Cranmer, Daniel C. Hackett, and Phiala E. Shanahan. Sampling using $\mathrm{SU}(n)$ gauge equivariant flows. *Phys. Rev. D*, 103:074504, Apr 2021. doi: 10.1103/PhysRevD.103.074504. URL https://link.aps.org/doi/10.1103/PhysRevD.103.074504.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Pim de Haan, Corrado Rainone, Miranda C. N. Cheng, and Roberto Bondesan. Scaling up machine learning for quantum field theory with equivariant continuous flows, 2021. URL https://arxiv.org/abs/2110.02673.

Mathis Gerdes, Pim de Haan, Corrado Rainone, Roberto Bondesan, and Miranda C. N. Cheng. Learning lattice quantum field theories with equivariant continuous flows, 2022. URL https://arxiv.org/abs/2207.00283.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: Exact likelihood generative learning for symmetric densities, 2020. URL https://arxiv.org/abs/2006.02425.

Bálint Máté and François Fleuret. Deformations of boltzmann distributions. *arXiv preprint arXiv:2210.13772*, 2022.

Laurence Illing Midgley, Vincent Stimper, Gregor NC Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. *arXiv preprint arXiv:2208.01893*, 2022.

Radford M. Neal. Annealed importance sampling, 1998. URL https://arxiv.org/abs/physics/9803008.

Kim A. Nicoli, Shinichi Nakajima, Nils Strodthoff, Wojciech Samek, Klaus-Robert Müller, and Pan Kessel. Asymptotically unbiased estimation of physical observables with neural samplers. *Physical Review E*, 101(2), feb 2020. doi: 10.1103/physreve.101.023304. URL https://doi.org/10.1103%2Fphysreve.101.023304.

Kim A Nicoli, Christopher J Anders, Lena Funcke, Tobias Hartung, Karl Jansen, Pan Kessel, Shinichi Nakajima, and Paolo Stornati. Estimation of thermodynamic observables in lattice field theories with deep generative models. *Physical review letters*, 126(3):032001, 2021.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators – sampling equilibrium states of many-body systems with deep learning, 2018. URL https://arxiv.org/abs/1812.01729.

David Pfau and Danilo Rezende. Integrable nonparametric flows. *arXiv preprint arXiv:2012.02035*, 2020.

Lorenz Vaitl, Kim A Nicoli, Shinichi Nakajima, and Pan Kessel. Gradients should stay on path: better estimators of the reverse- and forward kl divergence for normalizing flows. *Machine Learning: Science and Technology*, 3(4):045006, oct 2022. doi: 10.1088/2632-2153/ac9455. URL https://dx.doi.org/10.1088/2632-2153/ac9455.

A IMPLEMENTATION AND TRAINING DETAILS

Every trainable object in our experiments is parametrized by a weighted sum of MLPs. The weighting is done by evenly spaced RBF time-kernels, one for each model. Importantly, our architecture is completely oblivious to the $\mathbb{Z}_2 \ltimes C_n$-symmetry of the ϕ^4 theory and computes the divergence numerically. We leave the exploitation of symmetries and the use of architectures with analytic expressions for the divergence of V_t (Köhler et al., 2020; Gerdes et al., 2022), as well as for ∇f_t and $\partial_t f_t$, for future work. The choices of hyperparameters are given in Table 2. Everything was implemented in JAX (Bradbury et al., 2018) and executed on one of eight A100 GPUs.

number of RBF-kernels in time	8
hidden layers per model	3
neurons per hidden layer	128
activation function	swish
base distribution	$\propto e^{-(x/2)^4}$
bath size during training	256
batch size during evaluation	4096
number of train steps	10^4
initial learning rate	3×10^{-3}
number of integration steps	50
deformation loss	$\lvert\mathcal{E}_{def}\rvert + \lvert\mathcal{E}_{def}\rvert^2$

Table 2: Hyperparamers and desing choices for our experiments. The learning rate was initialized to the value shown in the table and annealed to 0 following a cosine schedule.

Computational costs To optimize the "baseline" reverse KL objective one needs a parametrization of V_t and to integrate $\nabla \cdot V_t$ along the trajectories. In addition to this, the deformation loss also needs a parametrization for C_t and one for f_t and to integrate an expression depending on $\nabla \cdot V_t, \partial_t f, \nabla f$ and C_t. This means that the same number of training steps are more expensive computationally when using the deformation loss. The experiments lasted ~ 1.85-times longer when using the deformation loss instead of KL-divergence.

Weighting of the trajectories An important detail of the implementation is related to the weighting of the trajectories. In Eq. 6 we introduced $w(z) = \frac{e^{-S(\gamma_1(z))}}{q_\theta(\gamma_1(z))}$ assigning weights to trajectories. If q_θ is far from p then the value of this weight can be large, leading to numerical issues. To avoid this, we normalize these weights per batch.