Path gradient estimators for CNFs in Lattice Gauge Theory

Lorenz Vaitl, Simone Bacchio, Pan Kessel, Shinichi Nakajima, Kim Nicoli, Stefan Schaefer



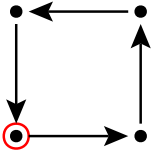



Problem setting

- 4D Yang-Mills Theory:

$$S = \frac{\beta}{3} \sum_{\mu,\nu<\mu,x} Re\mathrm{Tr}\left(I - P(x,\mu,\nu)\right)$$

  - $P(x,\mu,\nu)$: plaquette
  - β: inverse coupling
  - $P(x,\mu,\nu) \in SU(3)$
- Normalizing Flows
  - deep generative model
  - promise embarassingly parallel sampling
  - typically trained with self-sampling



$P(\circ, 1, 2)$

Normalizing Flows

- Diffeomorphism $\mathcal{T}_\theta : \mathcal{Z} \to \mathcal{X}$ from base density $p_0(U_0)$ to approximate target density $p(U)$:

$$q_\theta(U) = p_0(\mathcal{T}_\theta^{-1}(U)) \left| \det \frac{\partial \mathcal{T}_\theta^{-1}(U)}{\partial U} \right|$$

Normalizing Flows

- Diffeomorphism $\mathcal{T}_\theta : \mathcal{Z} \to \mathcal{X}$ from base density $p_0(U_0)$ to approximate target density $p(U)$:

$$q_\theta(U) = p_0(\mathcal{T}_\theta^{-1}(U)) \left| \det \frac{\partial \mathcal{T}_\theta^{-1}(U)}{\partial U} \right|$$

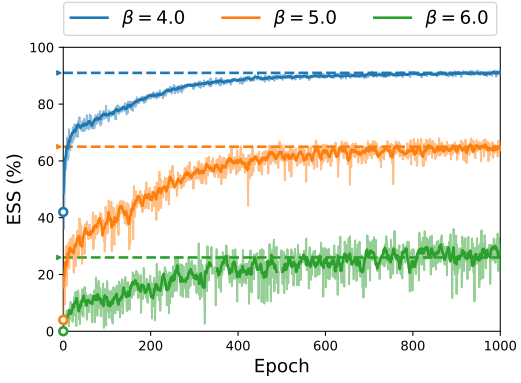
- Can be implemented via ODE (i.e. Continuous Normalizing Flow):

$$\mathcal{T}_\theta(U_0) = U_0 + \int_0^T \mathrm{d}t\, \dot{U}(U_t, t, \theta)$$

$$\left| \det \frac{\partial \mathcal{T}_\theta^{-1}(U)}{\partial U} \right| = \int_0^T \mathrm{tr} \left(\frac{\partial \dot{U}(U_t, t, \theta)}{\partial U_t} \right) \mathrm{d}t$$

Continuous Normalizing Flows

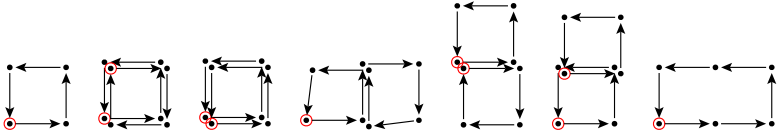
- Easy to incorporate symmetries [Köhler et al., 2020]
- Lüscher proposed continuous flow for LGT with parametrization by perturbative expansion [Lüscher, 2010]
- In recent work [Bacchio et al., 2023] we proposed optimizing the Lüscher's model with gradient descent



2D problem, $L = 16$

Lüscher's approach

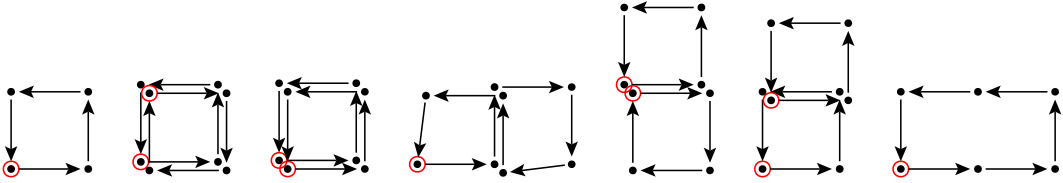
- Define $\dot{U}(U_t, t, \theta) = \partial\tilde{S}(U_t, t) \cdot U_t$, where $\partial\tilde{S}(U_t, t)$ is force of generic action
  - $\tilde{S}$ is scalar & invariant
    - Force equivariant and is element of Lie algebra $\mathfrak{su}(N)$
    - generic ODE for lattice gauge theory
  - $\tilde{S} = \Sigma_i c_i(t) W_i(U_t)$
    - W_i are traces of Wilson loops
    - $c_i(t)$ are time dependent coefficients parametrized by θ
- Lüscher found $W_i \& c_i(t)$ by a perturbative expansion around $t = 0$



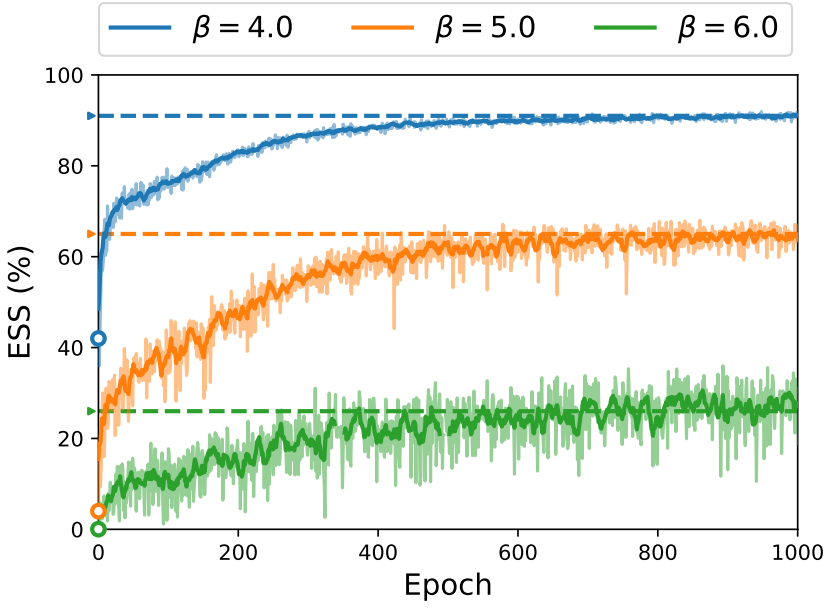
Training Trivializing Maps

- We proposed optimizing θ by minimizing the $\mathrm{KL}(q_\theta|p) \overset{c}{=} \mathbb{E}_{q_\theta(U_T)}\left[\ln q_\theta(U_T) + S(U_T)\right]$
  - Derived adjoint-state method for adjoint state $\in \mathfrak{su}(N)$
  - Implemented CG3 ODE solver, more complex functions for $c_i(t)$
  - Used **path gradients** for low variance gradient estimators

$\Rightarrow$ Expressive model with **few** parameters (14: linear $c_i(t)$ for each Wilson loop W_i)



2D SU(3)
Yang-Mills
Theory, $L = 16$



Path Gradient Estimators

- Low variance [Roeder et al., 2017]
  - efficiently computed with a modified adjoint method [Vaitl et al., 2022b]
- Reverse KL

$$KL(q_\theta|p) = \mathbb{E}_{U_T \sim q_\theta(U_T)} \left[\ln \frac{q_\theta(U_T)}{p(U_T)} \right] = \mathbb{E}_{U_0 \sim p_0} \left[\ln \frac{q_\theta(T_\theta(U_0))}{p(T_\theta(U_0))} \right]$$

Path Gradient Estimators

- Low variance [Roeder et al., 2017]
  - efficiently computed with a modified adjoint method [Vaitl et al., 2022b]
- Reverse KL total gradient

$$\frac{d}{d\theta} KL(q_\theta|p) = \frac{d}{d\theta} \mathbb{E}_{U_T \sim q_\theta(U_T)} \left[\ln \frac{q_\theta(U_T)}{p(U_T)} \right] = \frac{d}{d\theta} \mathbb{E}_{U_0 \sim p_0} \left[\ln \frac{q_\theta(T_\theta(U_0))}{p(T_\theta(U_0))} \right]$$

Path Gradient Estimators

- Low variance [Roeder et al., 2017]
  - efficiently computed with a modified adjoint method [Vaitl et al., 2022b]
- Reverse KL total gradient

$$\frac{d}{d\theta} KL(q_\theta|p) = \frac{d}{d\theta} \mathbb{E}_{U_T \sim q_\theta(U_T)} \left[\ln \frac{q_\theta(U_T)}{p(U_T)} \right] = \mathbb{E}_{U_0 \sim p_0} \left[\frac{d}{d\theta} \ln \frac{q_\theta(T_\theta(U_0))}{p(T_\theta(U_0))} \right]$$

Path Gradient Estimators

- Low variance [Roeder et al., 2017]
  - efficiently computed with a modified adjoint method [Vaitl et al., 2022b]
- Reverse KL total gradient

$$\frac{d}{d\theta} KL(q_\theta|p) = \frac{d}{d\theta} \mathbb{E}_{U_T \sim q_\theta(U_T)} \left[\ln \frac{q_\theta(U_T)}{p(U_T)} \right] = \mathbb{E}_{U_0 \sim p_0} \left[\frac{d}{d\theta} \ln \frac{q_\theta(T_\theta(U_0))}{p(T_\theta(U_0))} \right]$$

$$\frac{d}{d\theta} KL(q_\theta|p) = \mathbb{E}_{U_0 \sim p_0} \left[\frac{\partial}{\partial T_\theta(U_0)} \left(\ln \frac{q_\theta(T_\theta(U_0))}{p(T_\theta(U_0))} \right) \frac{\partial T_\theta(U_0)}{\partial \theta} + \frac{\partial \ln q_\theta(U_T)}{\partial \theta} \Big|_{U_T = T_\theta(U_0)} \right]$$

Path Gradient Estimators

- Low variance [Roeder et al., 2017]
  - efficiently computed with a modified adjoint method [Vaitl et al., 2022b]
- Reverse KL total gradient

$$\frac{d}{d\theta} KL(q_\theta|p) = \frac{d}{d\theta} \mathbb{E}_{U_T \sim q_\theta(U_T)} \left[\ln \frac{q_\theta(U_T)}{p(U_T)} \right] = \mathbb{E}_{U_0 \sim p_0} \left[\frac{d}{d\theta} \ln \frac{q_\theta(T_\theta(U_0))}{p(T_\theta(U_0))} \right]$$

- Path Gradient / Sticking-the-Landing

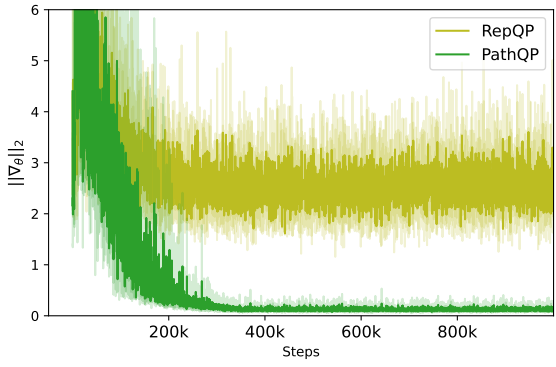
$$\frac{d}{d\theta} KL(q_\theta|p) = \mathbb{E}_{U_0 \sim p_0} \left[\frac{\partial}{\partial T_\theta(U_0)} \left(\ln \frac{q_\theta(T_\theta(U_0))}{p(T_\theta(U_0))} \right) \frac{\partial T_\theta(U_0)}{\partial \theta} + \cancel{\frac{\partial \ln q_\theta(U_T)}{\partial \theta}} \Big|_{U_T = T_\theta(U_0)} \right]$$

[Roeder et al., 2017]

- Path Gradient

$$\frac{d}{d\theta} KL(q_\theta|p) = \mathbb{E}_{U_0 \sim p_0}\left[\frac{\partial}{\partial T_\theta(U_0)} \left(\ln \frac{q_\theta(T_\theta(U_0))}{p(T_\theta(U_0))} \right) \frac{\partial T_\theta(U_0)}{\partial \theta} + \cancel{\frac{\partial \ln q_\theta(U_T)}{\partial \theta}}\Big|_{U_T = T_\theta(U_0)} \right]$$

Favourable behavior seen in many contexts e.g.
[Roeder et al., 2017, Tucker et al., 2019, Agrawal et al., 2020, Vaitl et al., 2022a]



Score term as control variate

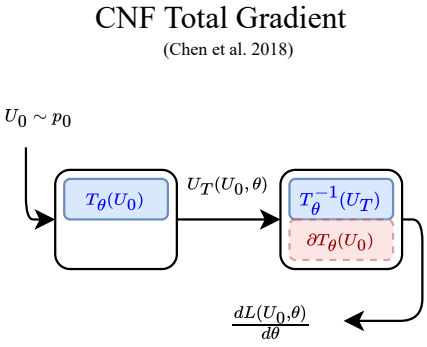
- Path gradients can be seen as akin to using the score term $\mathcal{G}_{\text{score}}$ as a control variate with a constant factor of 1

$$\underbrace{\frac{\partial \left(\ln \frac{q_\theta(T_\theta(U_0))}{p(T_\theta(U_0))} \right)}{\partial T_\theta(U_0)} \frac{\partial T_\theta(U_0)}{\partial \theta}}_{\mathcal{G}_{\text{path}}} = \underbrace{\frac{d}{d\theta} \ln \frac{q_\theta(T_\theta(U_0))}{p(T_\theta(U_0))}}_{\mathcal{G}_{\text{total}}} - 1 \cdot \underbrace{\frac{\partial \ln q_\theta(U_T)}{\partial \theta}\Big|_{U_T = T_\theta(U_0)}}_{\mathcal{G}_{\text{score}}}$$

- Remarks:
  - Empirically we did not see an improvement when estimating cv factor
  - Can also be interpreted as double reparametrization [Tucker et al., 2019]
  - The variance of the score term $\mathcal{G}_{\text{score}}$ is the Fisher Information $\mathcal{I}(\theta)$ divided by the batch-size
  - Path gradients work on a per sample basis

Adjoint state method CNF

- Gradients in standard neural ODEs computed using adjoint state method



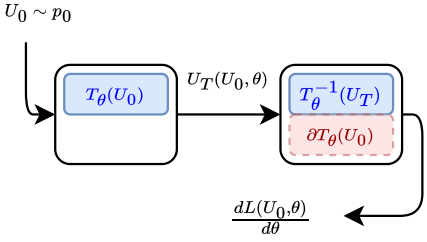
Adjoint state method CNF

- Gradients in standard neural ODEs computed using adjoint state method

$$U_T = U_0 + \int_0^T \dot{U}(U_t, t, \theta)dt \quad \Big\} \quad T_\theta(U_0)$$

CNF Total Gradient

(Chen et al. 2018)



Adjoint state method CNF

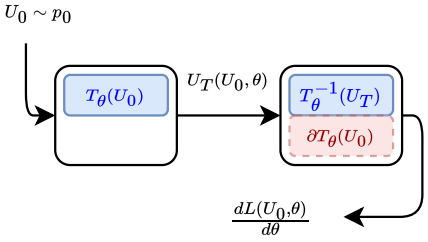
- Gradients in standard neural ODEs computed using adjoint state method

$$
\left.
\begin{aligned}
U_T &= U_0 + \int_0^T \dot{U}(U_t, t, \theta) dt
\end{aligned}
\right\} \quad T_\theta(U_0)
$$

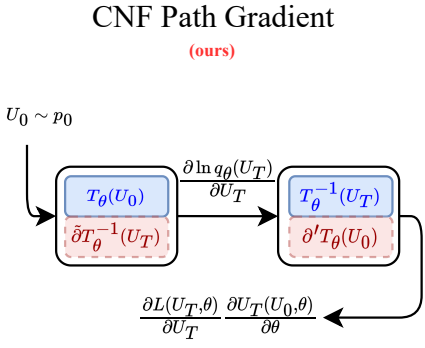
$$
\left.
\begin{aligned}
\lambda(T) &= \frac{\partial \ln p(U_T)}{\partial U_T} \\
\dot{\lambda}(t) &= -\lambda(t) \frac{\partial \dot{U}(U_t, t, \theta)}{\partial U_t} \\
\frac{dL}{d\theta} &= -\int_T^0 \lambda(t) \frac{\partial \dot{U}(U_t, t, \theta)}{\partial \theta} \\
&\quad -\partial_\theta \mathrm{tr} \left(\frac{\partial \dot{U}(U_t, t, \theta)}{\partial U_t} \right) dt
\end{aligned}
\right\} \quad \partial T_\theta(U_0)
$$

CNF Total Gradient

(Chen et al. 2018)



Modified adjoint state method [Vaitl et al., 2022b]



Modified adjoint state method [Vaitl et al., 2022b]

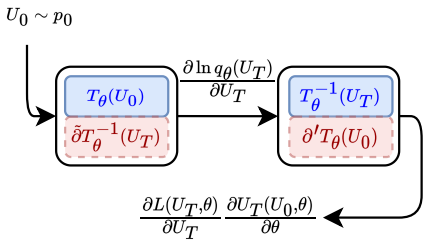
$$\tilde{\lambda}(0) = \partial_{U_0} \ln p_0(U_0)$$

$$\dot{\tilde{\lambda}}(t) = -\tilde{\lambda}(t) \frac{\partial \dot{U}(U_t, t, \theta)}{\partial U_t} - \partial_{U_t} \mathrm{tr} \left(\frac{\partial \dot{U}(U_t, t, \theta)}{\partial U_t} \right)$$

$$\Big\} \; \tilde{\partial} T_\theta^{-1}(U_T)$$

CNF Path Gradient

(ours)



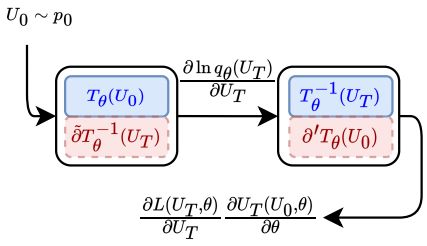
Modified adjoint state method [Vaitl et al., 2022b]

$$\left.\begin{aligned}
\tilde{\lambda}(0) &= \partial_{U_0} \ln p_0(U_0) \\
\dot{\tilde{\lambda}}(t) &= -\tilde{\lambda}(t) \frac{\partial \dot{U}(U_t, t, \theta)}{\partial U_t} \\
&\quad -\partial_{U_t} \mathrm{tr} \left(\frac{\partial \dot{U}(U_t, t, \theta)}{\partial U_t} \right)
\end{aligned}\right\} \tilde{\partial} T_\theta^{-1}(U_T)$$

$$\left.\begin{aligned}
\lambda'(T) &= \frac{\partial \ln q_\theta(U_T) - \ln p(U_T)}{\partial U_T} \\
\dot{\lambda}'(t) &= -\lambda'(t) \frac{\partial \dot{U}(U_t, t, \theta)}{\partial U_t} \\
\frac{\partial L}{\partial U_T} \frac{\partial U_T}{\partial \theta} &= -\int_T^0 \lambda'(t) \frac{\partial \dot{U}(U_t, t, \theta)}{\partial \theta} dt
\end{aligned}\right\} \partial' T_\theta(U_0)$$

CNF Path Gradient

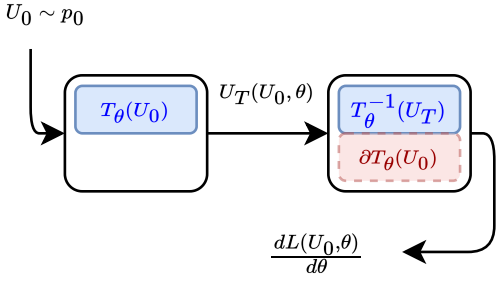
(ours)



Path gradients for CNFs

CNF Total Gradient

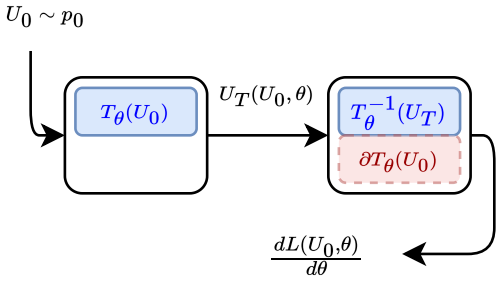
(Chen et al. 2018)



Path gradients for CNFs

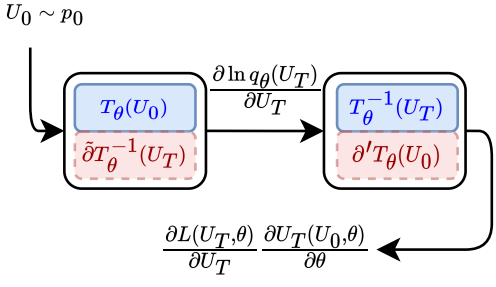
CNF Total Gradient

(Chen et al. 2018)



CNF Path Gradient

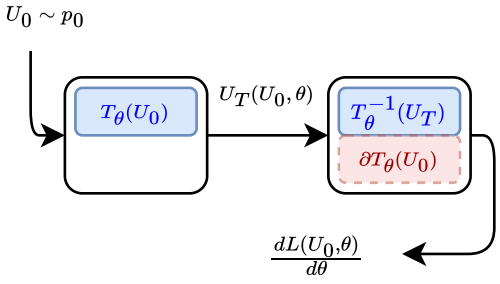
(ours)



Path gradients for CNFs

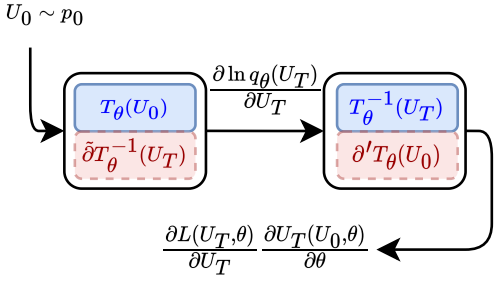
CNF Total Gradient

(Chen et al. 2018)



CNF Path Gradient

(ours)

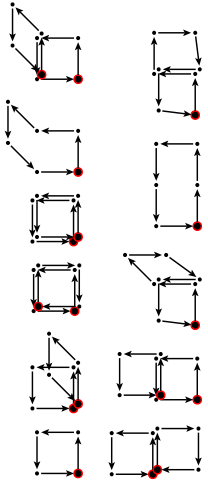


- Same memory requirements, 33% runtime increase per iteration

Results Trivializing Map in 4D

Training

- 4D SU(3) Yang-Mills Theory
- 11 Wilson loops
- Target $\beta \in \{1, 2, 3, 4\}$
  - $c_i(t)$ cubic splines with 2,5,7,10 knots
  - 5,10,15,20 ODE steps
- Lattice size 8, base-density uniform
- Batch-size 1, Adam, learning-rate 10^{-4}, trained on 1 A100
  - Trained on Juwels-Booster

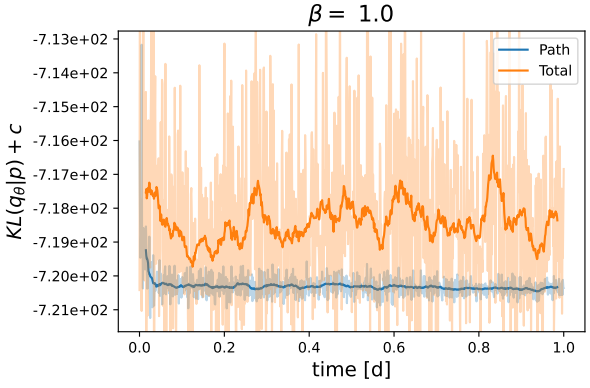


Results in 4D LGT

Effective Sampling Ratio

Estimated on 1k samples

β	Path	Total	days trained
1	96.6 %	13.7 %	1

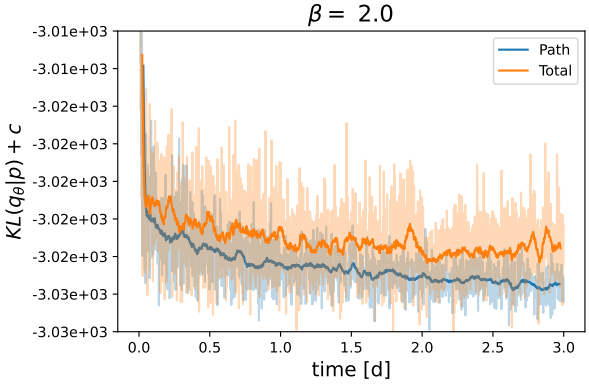


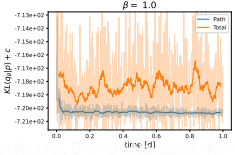
Results in 4D LGT

Effective Sampling Ratio

Estimated on 1k samples

β	Path	Total	days trained
1	96.6 %	13.7 %	1
2	40.1 %	16.7 %	3





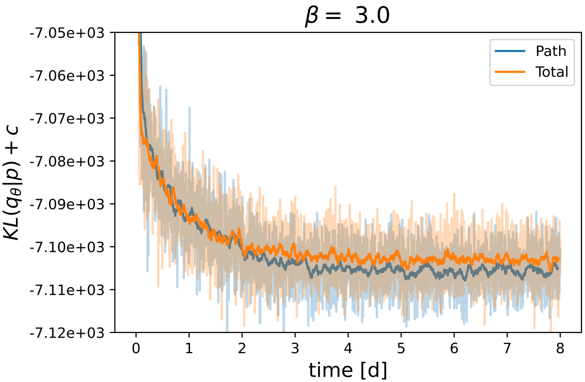
Lorenz Vaitl, Simone Bacchio, Pan Kessel, Stefan Schaefer, Jan Nicoll, Stefan Schaefer

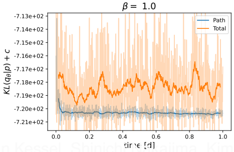
Results in 4D LGT

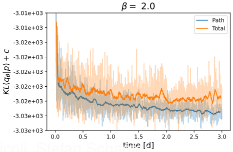
Effective Sampling Ratio

Estimated on 1k samples

β	Path	Total	days trained
1	96.6 %	13.7 %	1
2	40.1 %	16.7 %	3
3	00.8 %	00.4 %	8





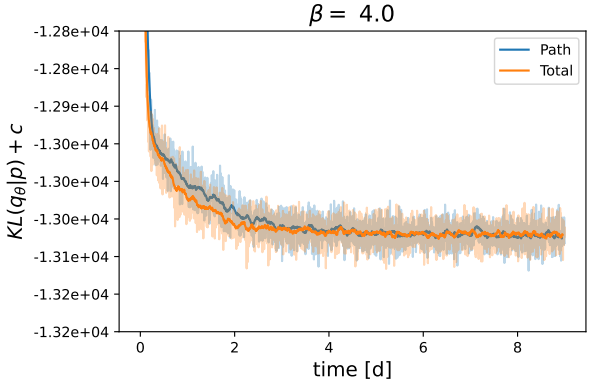


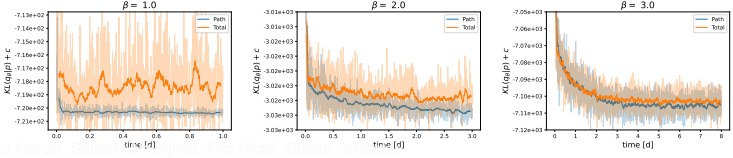
Results in 4D LGT

Effective Sampling Ratio

Estimated on 1k samples

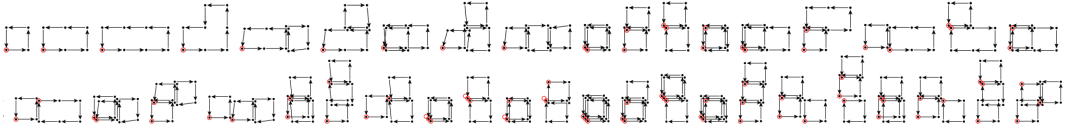
β	Path	Total	days trained
1	96.6 %	13.7 %	1
2	40.1 %	16.7 %	3
3	00.8 %	00.4 %	8
4	00.2 %	00.1 %	9





Summary

- Path gradients help training
- As is, the proposed CNF is not able to scale up to interesting β and lattice size
  - Possible to make flow more complex (e.g. NNLO basis), but drastic increase in runtime
  - Problem becomes exponentially more complex with increasing target β and lattice size



Thank you for your attention

References i

Agrawal, A., Sheldon, D. R., and Domke, J. (2020).
Advances in black-box VI: normalizing flows, importance weighting, and optimization.
In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Bacchio, S., Kessel, P., Schaefer, S., and Vaitl, L. (2023).
Learning trivializing gradient flows for lattice gauge theories.
Physical Review D, 107(5):L051504.

References ii

de Haan, P., Rainone, C., Cheng, M. C. N., and Bondesan, R. (2021).
Scaling up machine learning for quantum field theory with equivariant continuous flows.
CoRR, abs/2110.02673.

Köhler, J., Klein, L., and Noé, F. (2020).
Equivariant flows: Exact likelihood generative learning for symmetric densities.
In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5361–5370. PMLR.

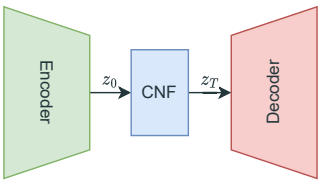
References iii

- Lüscher, M. (2010).
  Trivializing maps, the wilson flow and the hmc algorithm.
  Communications in mathematical physics, 293:899–919.
- Roeder, G., Wu, Y., and Duvenaud, D. (2017).
  Sticking the landing: Simple, lower-variance gradient estimators for variational inference.
  In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6925–6934.

References iv

- Tucker, G., Lawson, D., Gu, S., and Maddison, C. J. (2019). **Doubly reparameterized gradient estimators for monte carlo objectives.** In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Vaitl, L., Nicoli, K. A., Nakajima, S., and Kessel, P. (2022a). **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.
- Vaitl, L., Nicoli, K. A., Nakajima, S., and Kessel, P. (2022b). **Path-gradient estimators for continuous normalizing flows.** In *International Conference on Machine Learning*, pages 21945–21959. PMLR.

Results VAE



ELBO	Path	Total
MNIST	**82.09** $\pm$ .04	82.82 $\pm$ .01
Omniglot	**96.61** $\pm$ .17	98.33 $\pm$ .09
Caltech Silhouettes	**101.93** $\pm$ .63	104.03 $\pm$ .43
Frey Faces	**4.35** $\pm$ .00	4.39 $\pm$ .01

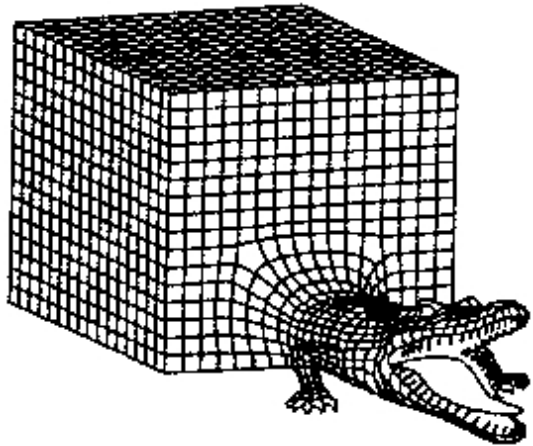
Lattice Field Theory:

- Target

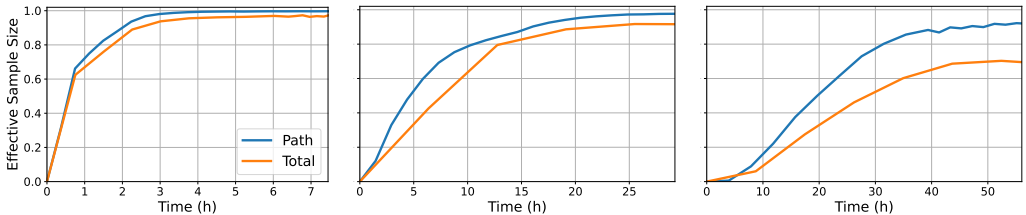
$$p(x) = \frac{1}{Z} e^{-S(x)}$$

  - Intractable
  - Known in closed form

- Can be approximated by CNF with inductive biases
  [de Haan et al., 2021]

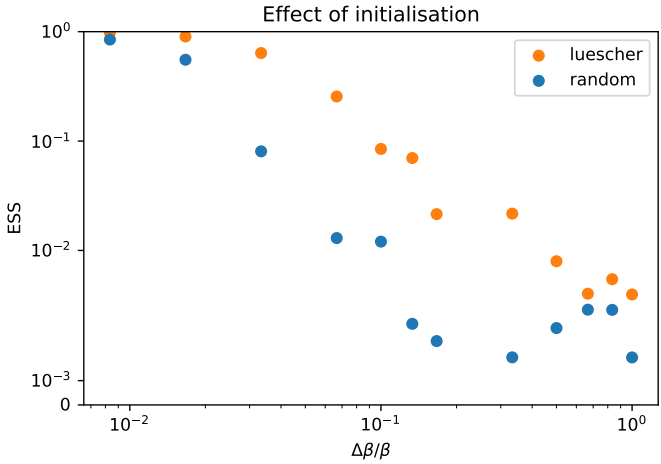


Lattice size	Path	Total
12x12	**99.66**$\% \pm 0.07$	$98.01\% \pm 0.44$
20x20	**97.65**$\% \pm 0.14$	$91.56\% \pm 1.13$
32x32	**91.81**$\% \pm 1.32$	$69.53\% \pm 5.59$



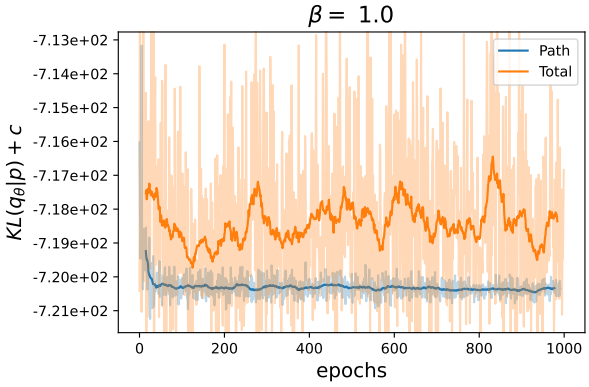
Training from non-trivial distribution

- 2D lattice, $L = 32$, target $\beta = 6$
- 1k epochs, batchsize 512



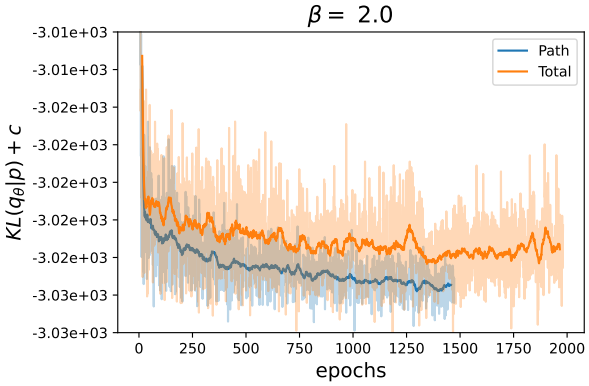
Results in 4D LGT

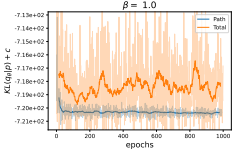
β	Path	Total	days trained
1	96.6 %	13.7 %	1



Results in 4D LGT

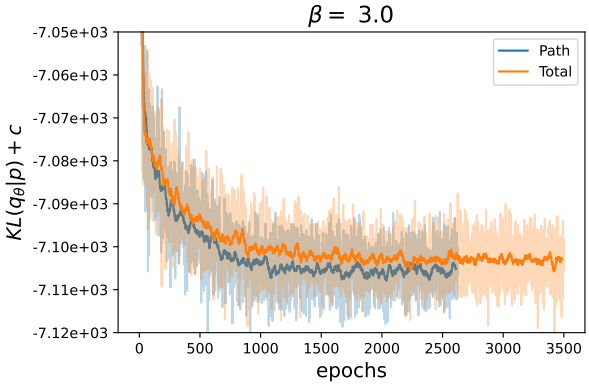
β	Path	Total	days trained
1	96.6 %	13.7 %	1
2	40.1 %	16.7 %	2

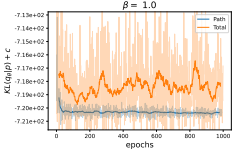


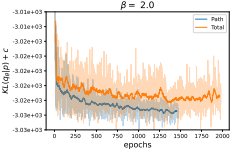


Results in 4D LGT

β	Path	Total	days trained
1	96.6 %	13.7 %	1
2	40.1 %	16.7 %	2
3	00.8 %	00.4 %	8

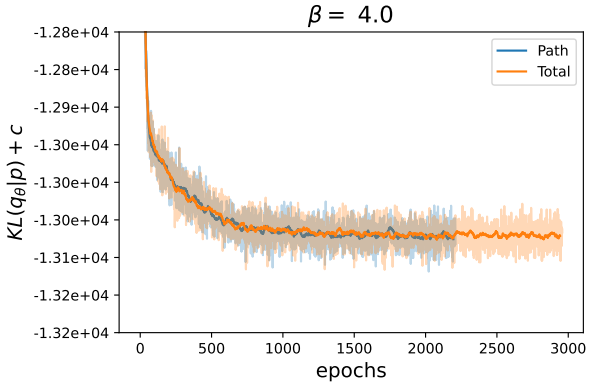


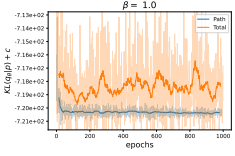


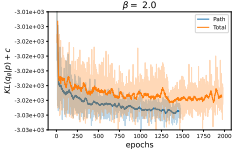


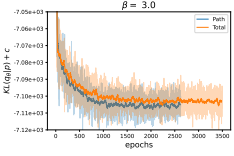
Results in 4D LGT

β	Path	Total	days trained
1	96.6 %	13.7 %	1
2	40.1 %	16.7 %	2
3	00.8 %	00.4 %	8
4	00.2 %	00.1 %	9









Acceptance rate, 4D LGT

Acceptance rate

Estimated on 1k samples

β	Path	Total	days trained
1	91 %	28 %	1
2	49 %	25 %	3
3	1 %	1 %	8
4	0 %	0 %	9

2D coefficients