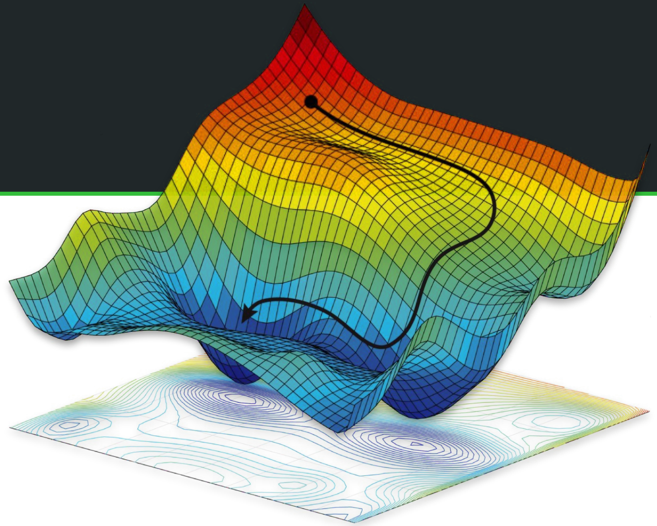
Continuous Normalizing Flows

for Lattice Gauge Theories



Dr. Simone Bacchio

Associate Research Scientist
CaSToRC, The Cyprus Institute

10/12/24 -NGT Algorithm Workshop, CERN

Outline

arXiv:2212.08469

Learning Trivializing Gradient Flows for Lattice Gauge Theories

Simone Bacchio,[1] Pan Kessel,[2,3] Stefan Schaefer,[4] and Lorenz Vaitl[2]
[1] *Computation-based Science and Technology Research Center, The Cyprus Institute, Nicosia, Cyprus*
[2] *Machine Learning Group, Technische Universität Berlin, Berlin, Germany*
[3] *BIFOLD—Berlin Institute for the Foundations of Learning and Data, Berlin, Germany*
[4] *John von Neumann-Institut für Computing NIC, Deutsches Elektronen-Synchrotron DESY, Germany*
(Dated: December 19, 2022)

First application of Continuous Normalizing Flows to Lattice Gauge Theories

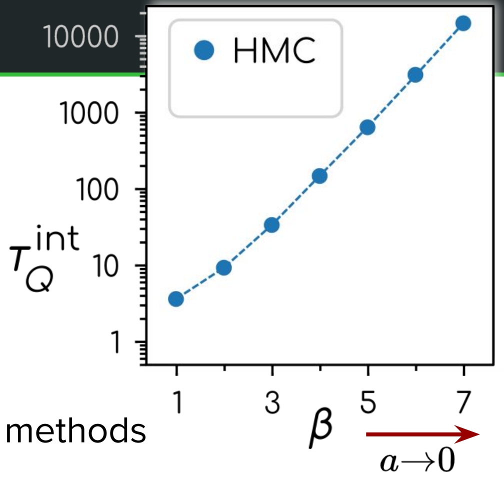
Realistic application of Machine Learning in Lattice QCD

arXiv:2305.07932

A novel approach for computing gradients of physical observables

Simone Bacchio[1]
[1] *Computation-based Science and Technology Research Center, The Cyprus Institute, Nicosia, Cyprus*
(Dated: May 21, 2023)

Motivation: Critical slowing down

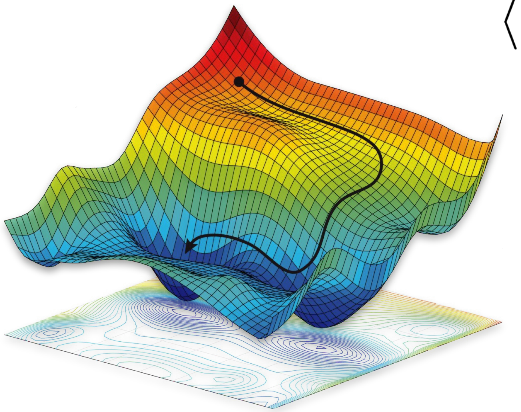


In Lattice Gauge Theories, the expectation value of an observable is defined as

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathrm{D}[U]\, \mathcal{O}(U)\, \exp(-S(U))$$

Computed via importance sampling and using Markov-chain Monte Carlo (MCMC) methods

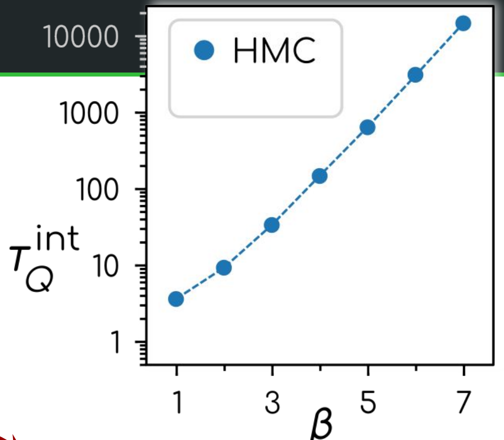
$$\langle \mathcal{O} \rangle = \frac{1}{N} \sum_i^N O(U_i) \quad \textit{with} \quad p(U) = \frac{1}{\mathcal{Z}} \exp(-S(U))$$



- **Requires independent and identically distributed (IID) samples**
- **State-of-the-art:** Hybrid Monte Carlo (HMC) algorithm

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathrm{D}[U]\, \mathrm{D}[p]\, \mathcal{O}(U)\, \exp(-p^2/2 - S(U))$$

Motivation: Critical slowing down

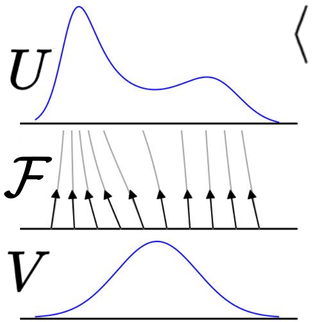


In Lattice Gauge Theories, the expectation value of an observable is defined as

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathrm{D}[U]\, \mathcal{O}(U)\, \exp(-S(U))$$

Alternative: If the field is generated by the transformation $U = \mathcal{F}(V)$

$$D[U] = D[V] \det \mathcal{F}_*(V)$$



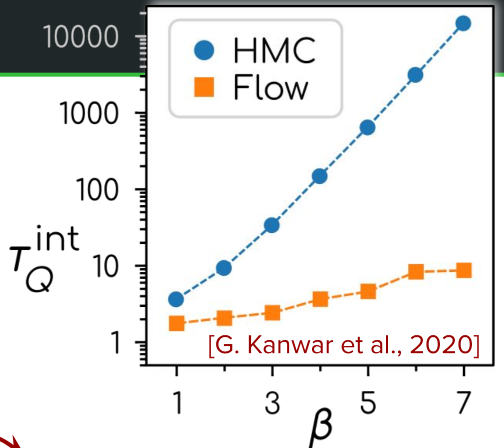
$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathrm{D}[V]\, \mathcal{O}(\mathcal{F}(V))\, \exp(-S_{\mathcal{F}}(V))$$

$$S_{\mathcal{F}}(V) = S(\mathcal{F}(V)) - \log\det \mathcal{F}_*(V)$$

The action is modified by the log-det. of the Jacobian of the transformation

Our goal, then, is to find a *flow*, $\mathcal{F}$, such that $S_{\mathcal{F}}(V)$ is easier to simulate / sample

Motivation: Critical slowing down

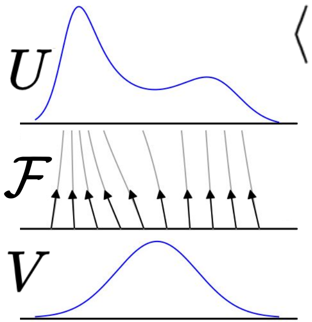


In Lattice Gauge Theories, the expectation value of an observable is defined as

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathrm{D}[U]\, \mathcal{O}(U)\, \exp(-S(U))$$

Alternative: If the field is generated by the transformation $U = \mathcal{F}(V)$

$$D[U] = D[V] \det \mathcal{F}_*(V)$$



$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathrm{D}[V]\, \mathcal{O}(\mathcal{F}(V))\, \exp(-S_{\mathcal{F}}(V))$$

$$S_{\mathcal{F}}(V) = S(\mathcal{F}(V)) - \log \det \mathcal{F}_*(V) = \text{const.}$$

Ultimate goal

Loss function

Our goal, then, is to find a *flow*, $\mathcal{F}$, such that $S_{\mathcal{F}}(V)$ is easier to simulate / sample

How to construct the flow?

Discrete flow

$$U = \mathcal{F}(V) = f_n \circ \cdots \circ f_2 \circ f_1(V)$$

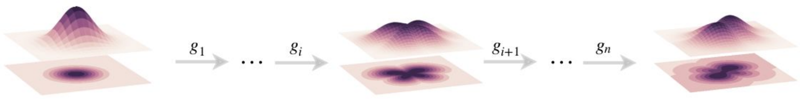
Requirements:

- Cheap calculation of $\det \mathcal{F}_*(V)$
- Preserve symmetries of the field

Approach:

- RealNVP & Equivariant Normalizing Flows

[G. Kanwar et al., 2020]



Continuous flow

$$U = \mathcal{F}(V) = \text{“}\int f(U_t, t)dt\text{”}$$

$$\dot{U}_t = f(U_t, t)$$

Same requirements…

Approach:

- Trivializing maps by Lüscher [M. Lüscher, 2009]
- … with machine learning [**S.B.** *et al.*, 2022]
- Focus of this presentation

Part 1: Continuous flows for Lattice Gauge Theories

How to define $$\dot{U}_t \equiv \frac{dU_t}{dt} = f(U_t, t) \quad ?$$

arXiv:0907.5491

Trivializing maps, the Wilson flow and the HMC algorithm

Martin Lüscher

CERN, Physics Department, 1211 Geneva 23, Switzerland

ODEs on manifolds

$$\dot{U}_t = \underbrace{g(U_t, t)}_{Z_t} U_t$$ where $U_t \in$ Group

$\boxed{Z_t \in \text{Algebra}}$

$$U_t = e^{\sum_a c_a(t)\, T_a}$$
$$\dot{U}_t = \left(\sum_a \dot{c}_a(t)\, T_a\right) U_t$$

- $Z_t = g(U_t, t)$ is an element of the algebra
- Imposing Gauge invariance:

$$U_\mu(x) \to \Omega(x) U_\mu(x) \Omega^\dagger(x + \mu) \quad \Longrightarrow \quad \boxed{Z_\mu(x) \to \Omega(x) Z_\mu(x) \Omega^\dagger(x)}$$

- Strong constraints on Z_t, **how to satisfy these properties?**

Lüscher's ansatz

$$Z_t = \partial \tilde{S}(U_t, t)$$

$$\tilde{S}(U_t, t) = \sum_i c_i(t) W_i(U_t)$$

arXiv:0907.5491

Trivializing maps, the Wilson flow and the HMC algorithm

Martin Lüscher

CERN, Physics Department, 1211 Geneva 23, Switzerland

- Z_t is the **force of a generic action** (i.e. scalar & gauge invariant quantity)
  - Any force (i.e. gradient w.r.t. the field components) is <u>an element of the algebra</u>
  - Any force of an action (i.e. gauge invariant quantity) satisfies $Z_\mu(x) \to \Omega(x) Z_\mu(x) \Omega^\dagger(x)$
- $\dot{U}_t = \left(\partial \tilde{S}(U_t, t)\right) U_t$ is the most generic and suitable ODE for lattice gauge theories!

Notation in Continuous Flows

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathrm{D}[V]\, \mathcal{O}(\mathcal{F}(V)) \, \exp(-S_{\mathcal{F}}(V))$$

$$S_{\mathcal{F}}(V) = S(\mathcal{F}(V)) - \log\det \mathcal{F}_*(V)$$

- $U_0 \equiv V$ base field
- $U_T \equiv \mathcal{F}(V) = \int_0^T \dot{U}_t \, dt$ integrated field
- $\dot{U}_t = \left(\partial \tilde{S}(U_t, t)\right) U_t$ flow ODE
- $\log\det \mathcal{F}_*(V) = \int_0^T \mathcal{L}_o \tilde{S}(U_t, t) dt$ with $\mathcal{L}_0 = -\sum_{x,\mu,a} \partial^a_{x,\mu} \partial^a_{x,\mu}$

Laplacian

Summing up on Continuous Flows

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathrm{D}[U_0]\, \mathcal{O}(U_T)\, \exp(-S_0(U_0))$$

$$\underbrace{S_0(U_0)}_{\text{Base action}} = \underbrace{S(U_T)}_{\text{Target action}} - \int_0^T \mathcal{L}_0 \underbrace{\tilde{S}(U_t, t)}_{\text{Flow action}} dt$$

- $S_0(U_0) = \text{const.}/S'(U_0)/\ldots$ training condition
- $\tilde{S}(U_t, t) = \sum_i c_i(t) W_i(U_t)$ trainable action with $c_i(t)$ free parameters
- $\mathcal{L}_0 \tilde{S}(U_t, t)$ is always an action-like term, e.g. $\mathcal{L}_0 W_0 = \frac{16}{3} W_0$ (but more complicated for loops with repeated links)

Part 2: Building on Lüscher's idea

How to find $c_i(t)$?

Perturbative approach around $\beta = 0$

arXiv:0907.5491

Trivializing maps, the Wilson flow and the HMC algorithm

Martin Lüscher

CERN, Physics Department, 1211 Geneva 23, Switzerland

Machine Learning approach

arXiv:2212.08469

Learning Trivializing Gradient Flows for Lattice Gauge Theories

Simone Bacchio,[1] Pan Kessel,[2,3] Stefan Schaefer,[4] and Lorenz Vaitl[2]

[1] *Computation-based Science and Technology Research Center, The Cyprus Institute, Nicosia, Cyprus*
[2] *Machine Learning Group, Technische Universität Berlin, Berlin, Germany*
[3] *BIFOLD—Berlin Institute for the Foundations of Learning and Data, Berlin, Germany*
[4] *John von Neumann-Institut für Computing NIC, Deutsches Elektronen-Synchrotron DESY, Germany*
(Dated: December 19, 2022)

or with CNN $c_i\left(t, \vec{W}_t(x)\right)$ [M. Gerdes et al., arXiv:2410.13161]

Lüscher's t-expansion

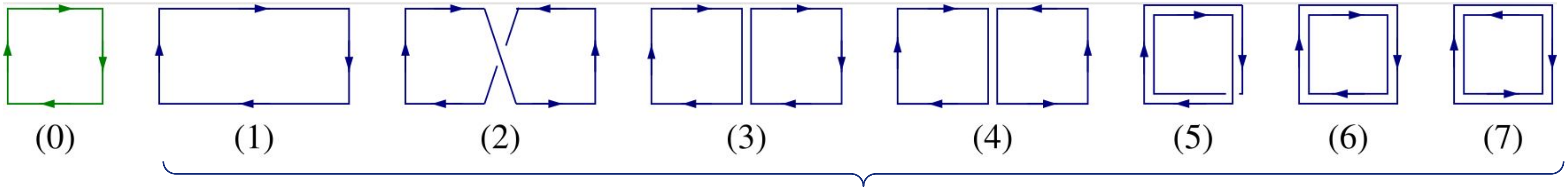
$$\tilde{S}(U_t, t) = \sum_i c_i(t) W_i(U_t)$$
$$= \sum_k t^k \tilde{S}^{(k)}(U_t) = \sum_k t^k \sum_i c_i^{(k)} W_i(U_t)$$

$$c_i(t) = \sum_k c_i^{(k)} t^k$$

- When solving around $\beta = 0$, i.e. $S(U_T) - \int_0^T \mathcal{L}_0 \tilde{S}(U_t, t) dt = \text{const.}$

$$\tilde{S}^{(0)} = \mathcal{L}_0^{-1} S$$
$$\tilde{S}^{(k)} = \mathcal{L}_0^{-1} \sum_{x,\mu,a} \partial^a_{x,\mu} S \, \partial^a_{x,\mu} \tilde{S}^{(k-1)} \quad \text{for } k > 0$$

Lüscher’s t-expansion - Wilson action



Terms appearing in the Next-to-Leading order,
i.e. all combinations of two plaquettes sharing a link

Things become very difficult... very fast!

- $$S = \frac{\beta}{6} W_0$$
- $$\tilde{S}^{(0)} = \mathcal{L}_0^{-1} S = -\frac{\beta}{32} W_0$$
- $$\tilde{S}^{(1)} = \mathcal{L}_0^{-1} \sum_{x,\mu,a} \partial^a_{x,\mu} S \, \partial^a_{x,\mu} \tilde{S}^{(0)} = \frac{\beta^2}{192}\Big(-\frac{4}{33} W_1 + \frac{12}{119} W_2 + \frac{1}{33} W_3 - \frac{5}{119} W_4 + \frac{3}{10} W_5 - \frac{1}{5} W_6 + \frac{1}{9} W_7 \Big)$$
- etc...

Machine Learning approach

$$\tilde{S}(U_t, t) = \sum_i c_i(t) W_i(U_t) \quad \longrightarrow \quad c_i(t, \vec{\theta})$$

- $\vec{\theta}$ are coefficients to train for finding the minimum of our tuning condition, i.e. *cost function*.
- Gradients of the cost function are needed for better & faster convergence

$$\frac{d}{d\vec{\theta}} \left(\underbrace{S(U_T) - S_0(U_0) - \int_0^T \mathcal{L}_0 \tilde{S}(U_t, t) dt}_{\text{Cost function}} \right)$$

Gradient of the cost function

$$\frac{\partial}{\partial \vec{\theta}} C(\vec{\theta}) = \underbrace{\frac{d}{d\vec{\theta}} S(U_T)}_{?} - \underbrace{\frac{d}{d\vec{\theta}} S_0(U_0)}_{0} - \int_0^T \underbrace{\frac{d}{d\vec{\theta}} \mathcal{L}_0 \tilde{S}(U_t, t)}_{?} dt$$

$$\frac{dU_t}{d\theta_i} = Y_t^{(i)} U_t \longrightarrow \frac{d}{d\theta_i} S(U_t) = \left(\partial S(U_t), Y_t^{(i)} \right)$$

Coefficients' ODE

Algebra's scalar product

- **ISSUE:** we have as many fields $Y_t^{(i)}$ as the number of parameters. Suitable only for few parameters...

Adjoint State method

$$\frac{\partial}{\partial\vec{\theta}} C(\vec{\theta}) = \int_0^T \left[\left(\underbrace{\lambda_t}_{\text{Adjoint State}}, \frac{\partial}{\partial\vec{\theta}} \partial\tilde{S}_t \right) - \frac{\partial}{\partial\vec{\theta}} \mathcal{L}_0 \tilde{S}_t \right] dt$$

- We use the adjoint state method to remove any dependence on $Y_t^{(i)}$. See [2212.08469] for details.

Adjoint State ODE

$$\dot{\lambda}_t = \partial\mathcal{L}_0 \tilde{S}_t + [\partial\tilde{S}_t, \lambda_t] - \sum_{y,\nu} \lambda_t^a(y,\nu) \underbrace{\partial\, \partial_{y,\nu}^a \tilde{S}_t}_{\text{Second derivative required}}$$

$$\lambda_T = \partial S(U_T)$$

The adjoint state is defined at time *T*, so we have to integrate backwards

Part 3: Numerical Results

2 Dimensional SU(3) Yang-Mills Theory

arXiv:2212.08469

Learning Trivializing Gradient Flows for Lattice Gauge Theories

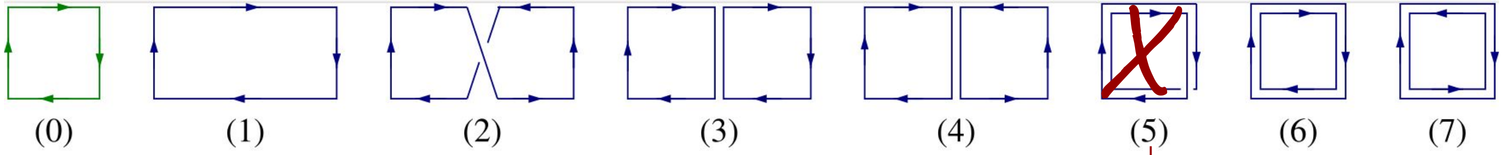
Simone Bacchio,[1] Pan Kessel,[2,3] Stefan Schaefer,[4] and Lorenz Vaitl[2]

[1] *Computation-based Science and Technology Research Center, The Cyprus Institute, Nicosia, Cyprus*
[2] *Machine Learning Group, Technische Universität Berlin, Berlin, Germany*
[3] *BIFOLD—Berlin Institute for the Foundations of Learning and Data, Berlin, Germany*
[4] *John von Neumann-Institut für Computing NIC, Deutsches Elektronen-Synchrotron DESY, Germany*

(Dated: December 19, 2022)

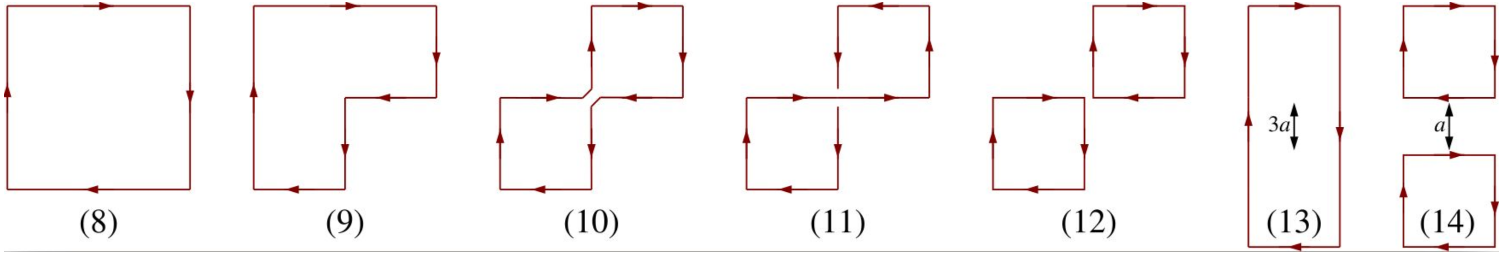
Considered Flow Actions

- **Model A:** Next-to-leading order of t-expansion, 7 Loops x Linear coefficients (2 params.)

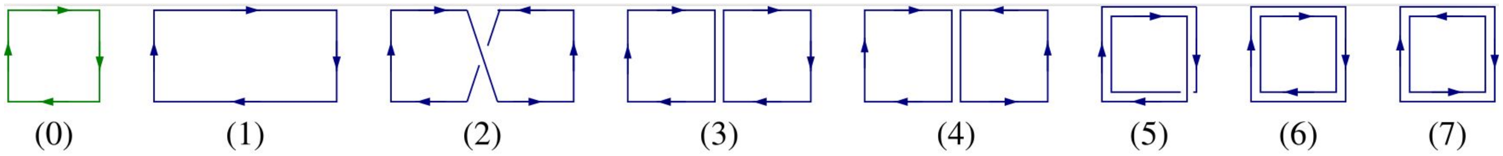


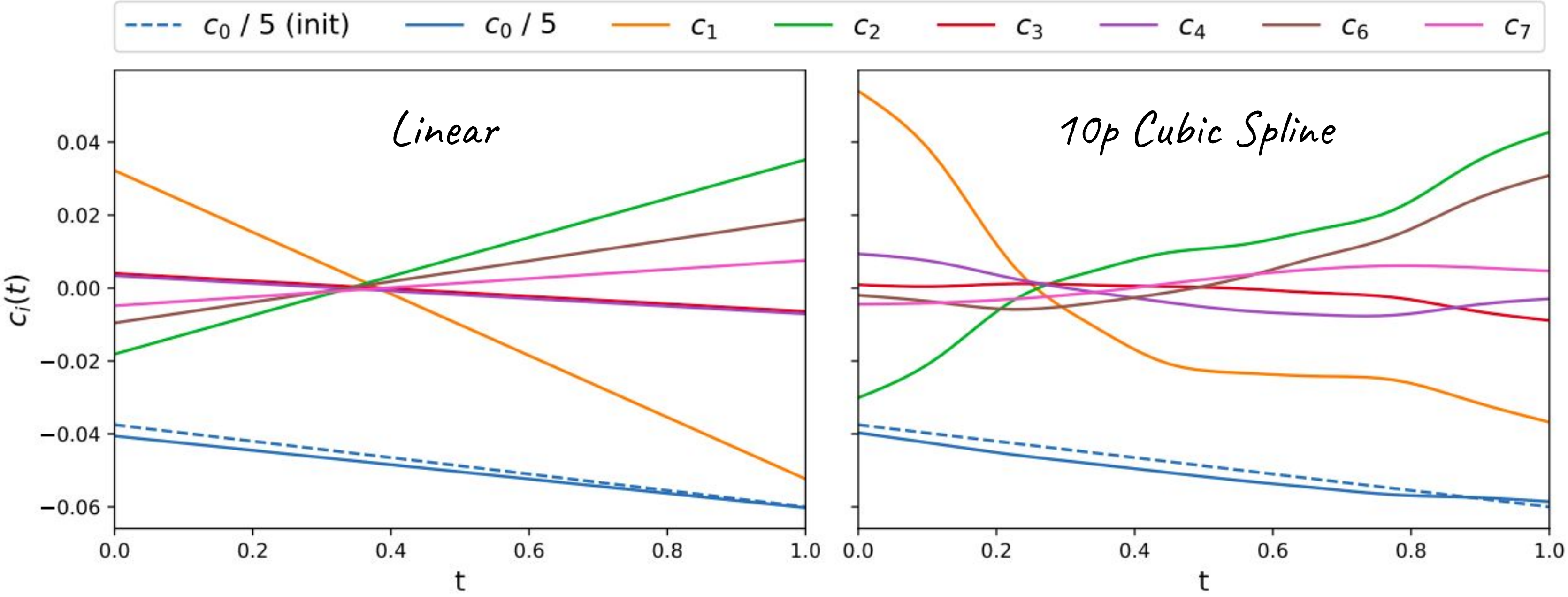
$W_5 = W_6 - 2W_0$

- **Model B:** 42 Wilson loops x 10 time points (interpolated by a cubic spline)



Time dependence of the coefficients is fundamental





Main Result: Machine-learned Gradient Flows

arXiv:2212.08469

Learning Trivializing Gradient Flows for Lattice Gauge Theories

Simone Bacchio,[1] Pan Kessel,[2,3] Stefan Schaefer,[4] and Lorenz Vaitl[2]

[1] *Computation-based Science and Technology Research Center, The Cyprus Institute, Nicosia, Cyprus*
[2] *Machine Learning Group, Technische Universität Berlin, Berlin, Germany*
[3] *BIFOLD—Berlin Institute for the Foundations of Learning and Data, Berlin, Germany*
[4] *John von Neumann-Institut für Computing NIC, Deutsches Elektronen-Synchrotron DESY, Germany*

(Dated: December 19, 2022)

Ref.		N_{params}	ESS at β 4.0	5.0	6.0
Lüscher		8 non-zero values	42%	4%	<1%
Our work	A	$14 \equiv 2_t \times 7_W$	91%	65%	26%
	B	$420 \equiv 10_t \times 42_W$	98%	88%	70%
MIT & DeepMind		$\gtrsim 4\,000\,000$	88%	75%	48%

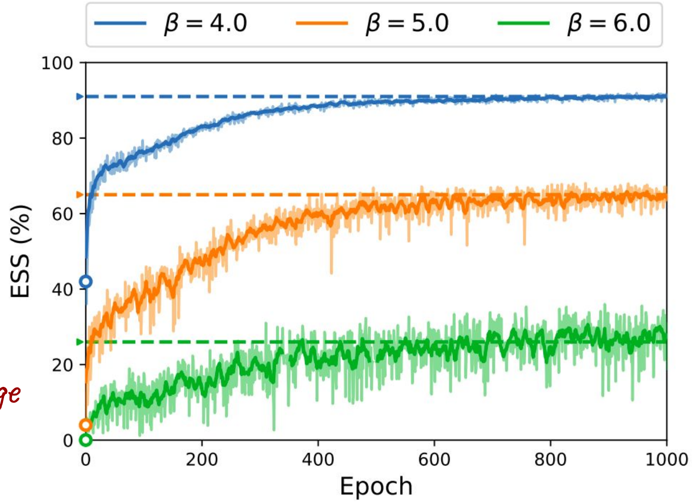
[arXiv:2410.13161] obtained ESS compatible to ours

We have developed the first continuous flow for Lattice QCD.

Advantages compared to "discrete" flows are

- Highest ESS and better scaling
- 5 orders of magnitude less parameters
- All symmetries of the theory preserved
- Faster training, 100x less iterations
- Initial guess from perturbation theory

NOTE: Results for SU(3) gauge theories on a 2D 16^2 lattice.



Part 4: Towards 4D results

4 Dimensional SU(3) Yang-Mills Theory

Target problem

Results on a 8^4 lattice	β	1.0	2.0	3.0	4.0
	ESS	91%	49%	1%	0%

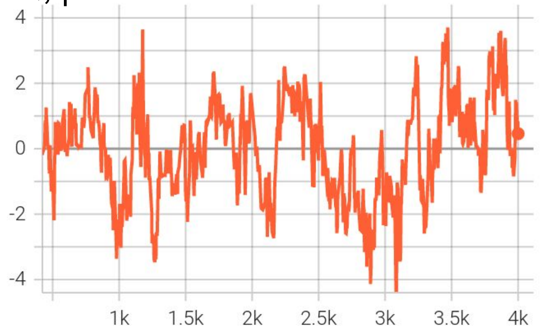
Moving away from toy models, a good target problem is

- **β=6.0** → 0.093 fm lattice spacing [1009.5228]
- **16^4 lattice** → 1.5 fm lattice size
- Long autocorrelation already visible in the smeared Q

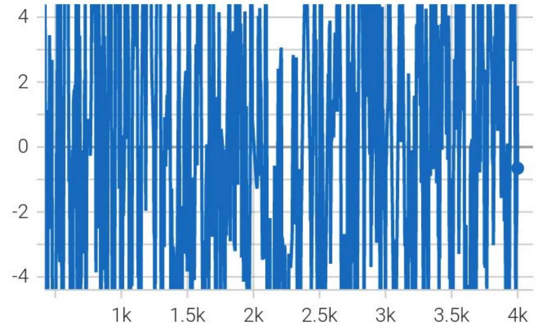
For the base distribution then we can consider

- Ideally, uniform → most probably impossible
- Realistically, β=5.7 → double lattice spacing
- Practically, …

Q, β=6.0



Q, β=5.7



Flowing in β

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathrm{D}[U]\, \mathcal{O}(U)\, \exp(-S(U)) = \frac{1}{\mathcal{Z}} \int \mathrm{D}[U_\tau]\, \mathcal{O}(U)\, \exp(-S_\tau(U_\tau))$$

Gradient Flow

$$S_\tau(U_\tau) = S(U) - \int_\tau^1 \mathrm{d}t\, \mathcal{L}_0 \tilde{S}(U_t, t) \quad \begin{cases} \dot{U}_t = -\partial \tilde{S}_t\, U_t \\ U \equiv U_1 \end{cases}$$

Desired base action

$$S_\tau = \tau S \qquad = -\tau \frac{\beta}{6} W_0 \qquad = -\frac{\beta_0}{6} W_0$$

$\tau = 1$ Target action

$\tau = 0$ Trivial action

$S = -\frac{\beta}{6} W_0$

Wilson action

$\tau = \frac{\beta_0}{\beta}$

Flowing in β

$$S_\tau(U_\tau) = S(U) - \int_\tau^1 \mathrm{d}t\, \mathcal{L}_0 \tilde{S}(U_t, t) \qquad \begin{cases} \dot{U}_t = -\partial \tilde{S}_t\, U_t \\ U \equiv U_1 \end{cases}$$

$$\frac{d}{d\tau}$$

$$\frac{\partial S_\tau}{\partial \tau} - \left(\partial S_\tau, \partial \tilde{S}_\tau\right) = \mathcal{L}_0 \tilde{S}_\tau \qquad S_\tau = \tau S$$

$$S - \tau\left(\partial S, \partial \tilde{S}_\tau\right) = \mathcal{L}_0 \tilde{S}_\tau$$

Wilson action $S = -\frac{\beta}{6} W_0$

$$\mathcal{L}_0 \tilde{S}_\tau - \frac{\tau\beta}{6}\left(\partial W_0, \partial \tilde{S}_\tau\right) + \frac{\beta}{6} W_0 = \text{const.}$$

Flowing in β

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathrm{D}[U]\, \mathcal{O}(U)\, \exp(-S_\beta(U)) = \frac{1}{\mathcal{Z}} \int \mathrm{D}[V]\, \mathcal{O}(U)\, \exp(-S_{\beta_0}(V))$$

Target

$$S_\beta(U) - S_{\beta_0}(V) - \int_{\beta_0/\beta}^{1} \mathrm{d}t\, \mathcal{L}_0 \tilde{S}(U_t, t) = \text{const.}$$

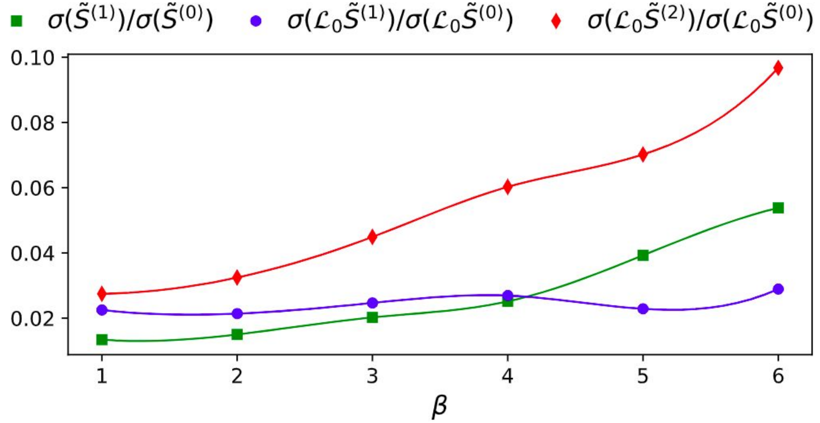
Solution

$$\mathcal{L}_0 \tilde{S}_t - \frac{t\beta}{6}\left(\partial W_0, \partial \tilde{S}_t\right) + \frac{\beta}{6} W_0 = \text{const.}$$

$$\tilde{S}(U_t, t) = \frac{\beta}{6} \sum_{k=0}^{\infty} \left(\frac{t\beta}{6}\right)^k \tilde{S}^{(k)}(U_t) \qquad \begin{cases} \tilde{S}^{(0)} = -\mathcal{L}_0^{-1} W_0 = -\frac{3}{16} W_0 \\ \tilde{S}^{(k)} = \mathcal{L}_0^{-1}\left(\partial W_0, \partial \tilde{S}^{(k-1)}\right) \end{cases}$$

Convergence of the flow action

$$\tilde{S}(U_t,t) = \frac{\beta}{6}\sum_{k=0}^{\infty}\left(\frac{t\beta}{6}\right)^k \tilde{S}^{(k)}(U_t)$$



- $t \leq 1$ ✔ Limit of the integral
- $\beta < 6$ ✗ Region of physical interest at $\beta \gtrsim 6$
- Order of magnitude of $\tilde{S}^{(k)}$??

$$\mathcal{L}_0\tilde{S}^{(2)} = \left(\partial W_0, \partial\tilde{S}^{(1)}\right)$$

- **Our conclusion:** <u>most probably it does not converge!</u>

DEAD END

Final Part: A realistic application

A novel approach for computing gradients of Physical observables

The standard approach for computing gradients

When taking a gradient of a physical observable, we have three contributions:

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathrm{D}[U]\, \mathcal{O}(U)\, \exp(-S(U))$$

$$\frac{d\langle \mathcal{O} \rangle}{d\theta} = \langle \underbrace{\frac{\partial \mathcal{O}}{\partial \theta}}_{\text{Valence contribution (usually connected)}} - \underbrace{\mathcal{O}\frac{\partial S}{\partial \theta} \rangle + \langle \mathcal{O} \rangle \langle \frac{\partial S}{\partial \theta} \rangle}_{\text{Sea contribution (disconnected)}}$$

The novel approach for computing gradients

Alternatively, we can compute gradients in the gradient flow formulation:

$$\left.\begin{aligned}\langle \mathcal{O} \rangle &= \frac{1}{\mathcal{Z}_{d\theta}} \int \mathrm{D}[U_{d\theta}]\, \mathcal{O}(U)\ \exp(-S_{d\theta}(U_{d\theta})) \\ S_{d\theta}(U_{d\theta}) &= S(U) - \int_0^{d\theta} \mathcal{L}_0 \tilde{S}(U_t, t) dt\end{aligned}\right\}$$

Infinitesimal flow

$$\left.\begin{aligned}S_{d\theta}(U_{d\theta}) - S(U) &= d\theta\, \frac{\partial S}{\partial \theta} + d\theta\, (\partial S, \partial \tilde{S}) \\ \int_0^{d\theta} \mathcal{L}_0 \tilde{S}(U_t, t) dt &= d\theta\, \mathcal{L}_0 \tilde{S} + \mathrm{const.}\end{aligned}\right\}$$

Leading-order expansion

$$\boxed{\mathcal{L}_0 \tilde{S} + (\partial S, \partial \tilde{S}) + \frac{\partial S}{\partial \theta} = \mathrm{const.}}$$

New loss function

Theorem and Corollaries

The generic formula, demonstrated in the paper, is

$$\frac{d\langle\mathcal{O}\rangle}{d\theta} = \langle\frac{\partial\mathcal{O}}{\partial\theta} + \left(\partial\mathcal{O}, \partial\tilde{S}\right) - \mathcal{O}\mathcal{C}\rangle + \langle\mathcal{O}\rangle\langle\mathcal{C}\rangle$$ with

$$\mathcal{C} = \mathcal{L}_0\tilde{S} + \left(\partial S, \partial\tilde{S}\right) + \frac{\partial S}{\partial\theta}$$

Corollary A: "The new approach"

If $\mathcal{C}$ = const. then

$$\frac{d\langle\mathcal{O}\rangle}{d\theta} = \langle\frac{\partial\mathcal{O}}{\partial\theta} + (\partial\mathcal{O}, \partial\tilde{S})\rangle$$

Corollary B: "The standard approach"

If $\tilde{S}$ = const. then

$$\frac{d\langle\mathcal{O}\rangle}{d\theta} = \langle\frac{\partial\mathcal{O}}{\partial\theta} - \mathcal{O}\frac{\partial S}{\partial\theta}\rangle + \langle\mathcal{O}\rangle\langle\frac{\partial S}{\partial\theta}\rangle$$

Opportunity? Calculation of gradients!

arXiv:2305.07932

A novel approach for computing gradients of physical observables

Simone Bacchio[1]

[1] *Computation-based Science and Technology Research Center, The Cyprus Institute, Nicosia, Cyprus*

(Dated: May 21, 2023)

- **Standard finite-difference approach:**

$$\frac{d\langle\mathcal{O}\rangle}{d\theta} = \langle \frac{\partial \mathcal{O}}{\partial\theta} - \underbrace{\mathcal{O}\frac{\partial S}{\partial\theta}\rangle + \langle\mathcal{O}\rangle\langle\frac{\partial S}{\partial\theta}\rangle}$$

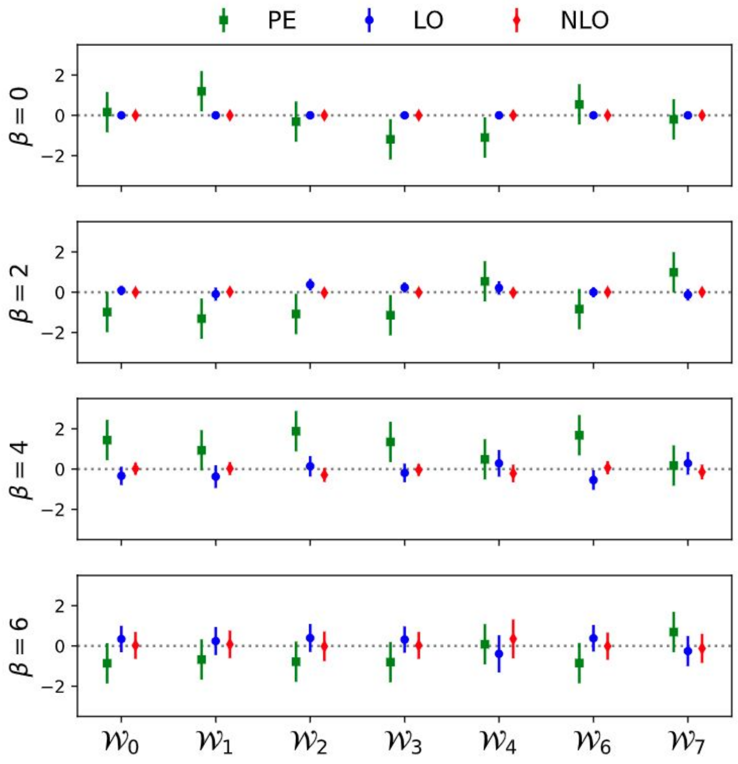
noisy disconnected contributions

- **Novel approach presented:**

$$\frac{d\langle\mathcal{O}\rangle}{d\theta} = \langle \frac{\partial \mathcal{O}}{\partial\theta} + \left(\partial\mathcal{O}, \partial\tilde{S}\right)\rangle$$

up to 1000x more precise

$$\tilde{S} = \frac{1}{6}\sum_{k=0}^{\infty}\left(\frac{\beta}{6}\right)^k \tilde{S}^{(k)}$$ *with* $$\tilde{S}^{(0)} = \mathcal{L}_0^{-1}\mathcal{W}_0 = \frac{3}{16}\mathcal{W}_0$$ $$\tilde{S}^{(k)} = \mathcal{L}_0^{-1}\left(\partial\mathcal{W}_0, \partial\tilde{S}^{(k-1)}\right) \quad \text{for } k > 0\,,$$



NOTE: Results for 4D pure-gauge SU(3), $V=16^4$

Remarks & Outlook

$$\mathcal{L}_0 \tilde{S} + \left(\partial S, \partial \tilde{S}\right) + \frac{\partial S}{\partial \theta} = \text{const.}$$

A new loss function for ML approaches

Applications:

- Calculation of gradients and leading-order effects (QED, Θ-term, isospin-breaking, etc.)
- Feynman-Hellmann approach for computing observables (g_A, σ-terms, gluons loops, etc.)

Approach:

- Any ML tools developed for LQCD. This is not a flow anymore, only one $\tilde{S}$ is needed!
- Generic analytic solution:

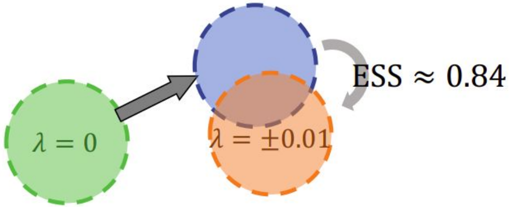
$$\tilde{S} = \sum_{k=0}^{\infty} \tilde{S}^{(k)} \quad \begin{cases} \tilde{S}^{(0)} = -\mathcal{L}_0^{-1} \frac{\partial S}{\partial \theta} \\ \tilde{S}^{(k)} = -\mathcal{L}_0^{-1} \left(\partial S, \partial \tilde{S}^{(k-1)}\right) \end{cases}$$

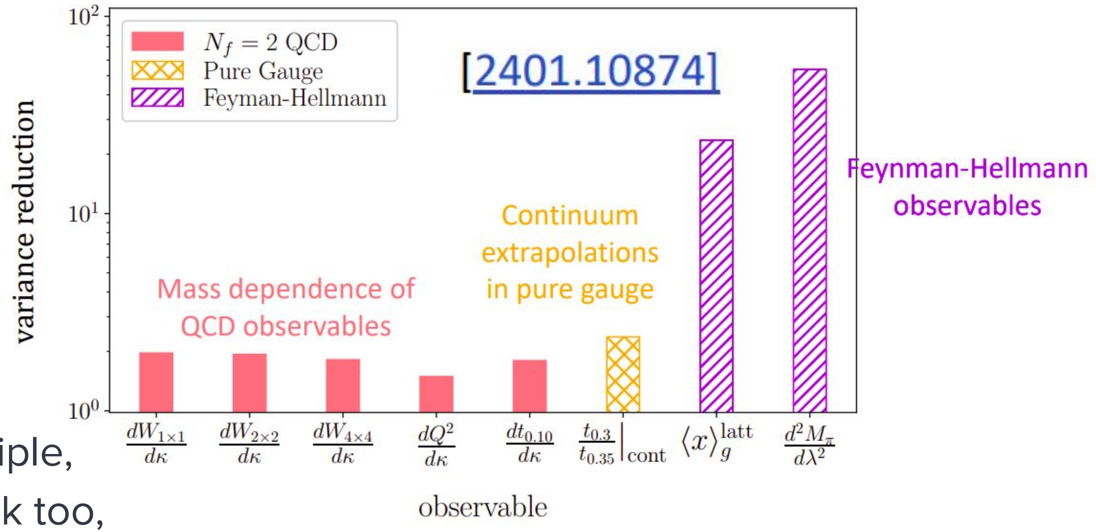
Can be used as initial guess

Critical for convergence

Comparison with related work

Similar benefits have been shown by R. Abbott et al. in the generation of correlated ensembles.





The two approaches are same in principle, i.e. where one works the other will work too, with few key differences:

- ✓ It is an exact gradients, no small λ needs to be introduced
- ✓ We have an analytics solution that can be used as initial guess for the training
- ! We require to compute the force of the observable, of the flow action and its Laplacian

Thank you for your attention!

arXiv:2212.08469

Learning Trivializing Gradient Flows for Lattice Gauge Theories

Simone Bacchio,[1] Pan Kessel,[2,3] Stefan Schaefer,[4] and Lorenz Vaitl[2]

[1] *Computation-based Science and Technology Research Center, The Cyprus Institute, Nicosia, Cyprus*
[2] *Machine Learning Group, Technische Universität Berlin, Berlin, Germany*
[3] *BIFOLD—Berlin Institute for the Foundations of Learning and Data, Berlin, Germany*
[4] *John von Neumann-Institut für Computing NIC, Deutsches Elektronen-Synchrotron DESY, Germany*

(Dated: December 19, 2022)

First application of Continuous Normalizing Flows to Lattice Gauge Theories

Realistic application of Machine Learning in Lattice QCD

arXiv:2305.07932

A novel approach for computing gradients of physical observables

Simone Bacchio[1]

[1] *Computation-based Science and Technology Research Center, The Cyprus Institute, Nicosia, Cyprus*

(Dated: May 21, 2023)