

Learning Trivializing Gradient Flows

[arXiv:2212.08469](https://arxiv.org/abs/2212.08469)



Simone Bacchio

Computational Scientist
CaSToRC, The Cyprus Institute

A work in collaboration with
Pan Kessel, Stefan Schaefer, Lorenz Vaitl



Introduction

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

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

If the field is generated by the transformation $U = \mathcal{F}(V)$ $D[U] = D[V] \det \mathcal{F}_*(V)$

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

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

Introduction

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

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

Equivalent to the
KL-divergence

Our goal, then, is to find a *flow*, \mathcal{F} , such that

- $S_{\mathcal{F}}(V)$ is easier to simulate with HMC (e.g. better autocorrelation) but force of $S_{\mathcal{F}}(V)$ is required
- $S_{\mathcal{F}}(V) \equiv S'(V)$, i.e. flows to a different action where it is easier to simulate (or already simulated)
- $S_{\mathcal{F}}(V) \equiv \text{const}$, i.e. all configurations V have the same weight and they are uniformly distributed

→ *Trivializing map* [M. Lüscher, 0907.5491] → *Normalizing flows* [G. Kanwar et al, 2003.06413]

How to construct the flow?

Direct approach

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

Requirements:

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

Approach:

- Affine coupling layers: [\[2008.05456\]](#)



Continuous flow

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

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

Same requirements...

Approach:

- Gradient flow by Lüscher: [\[0907.5491\]](#)
- ... with machine learning: [\[2212.08469\]](#)
- Focus of this presentation

Part 1: Continuous flows for Lattice Gauge Theories

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

[arXiv:0907.5491](https://arxiv.org/abs/0907.5491)

Trivializing maps, the Wilson flow and
the HMC algorithm

Martin Lüscher

CERN, Physics Department, 1211 Geneva 23, Switzerland

ODEs on manifolds

$$\dot{U}_t = \underbrace{g(U_t, t)}_{Z_t} U_t \quad \text{where} \quad \begin{aligned} U_t &\in \text{Group} \\ Z_t &\in \text{Algebra} \end{aligned}$$

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

$$U_\mu(x) \rightarrow \Omega(x)U_\mu(x)\Omega^\dagger(x + \mu) \longrightarrow Z_\mu(x) \rightarrow \Omega(x)Z_\mu(x)\Omega^\dagger(x)$$

- 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](https://arxiv.org/abs/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 an element of the algebra
 - Any force of an action (i.e. gauge invariant quantity) satisfies $Z_\mu(x) \rightarrow \Omega(x) Z_\mu(x) \Omega^\dagger(x)$
- $\dot{U}_t = (\partial \tilde{S}(U_t, t)) U_t$ is the most generic and suitable ODE for lattice gauge theories!

Notation in Continuous Flows

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

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

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

Summing up on Continuous Flows

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

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

- $S_0(U_0) = \text{const.}/S'(U_0)/\dots$ 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 equal to an action, 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](https://arxiv.org/abs/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](https://arxiv.org/abs/2212.08469)

Learning Trivializing Gradient Flows for Lattice Gauge Theories

Simone Bacchio,¹ Pan Kessel,^{2,3} Stefan Schaefer,⁴ and Lorenz Vaitl²

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

²Machine Learning Group, Technische Universität Berlin, Berlin, Germany

³BIFOLD—Berlin Institute for the Foundations of Learning and Data, Berlin, Germany

⁴John von Neumann-Institut für Computing NIC,
Deutsches Elektronen-Synchrotron DESY, Germany
(Dated: December 19, 2022)

Lüscher's t-expansion

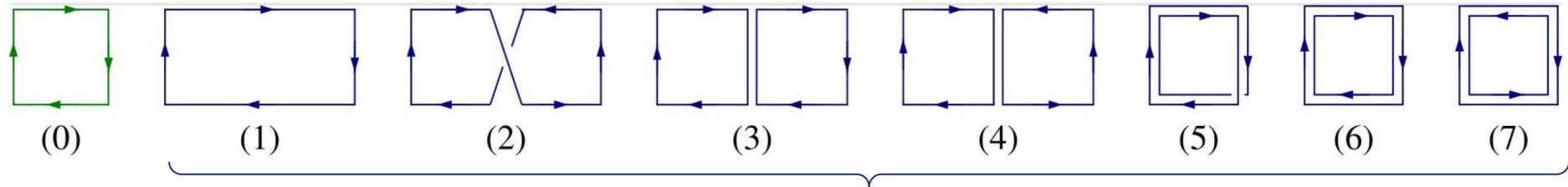
$$\begin{aligned}\tilde{S}(U_t, t) &= \sum_i c_i(t) W_i(U_t) \\ &= \sum_k t^k \tilde{S}^{(k)}(U_t) = \sum_k t^k \sum_i c_i^{(k)} W_i(U_t)\end{aligned}\quad \left.\right\} c_i(t) = \sum_k c_i^{(k)} t^k$$

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

$$\tilde{S}^{(0)} = \mathcal{L}_0^{-1} S$$

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

Lüscher's t-expansion - Wilson action



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

- $S = \frac{\beta}{6} W_0$
 - $\tilde{S}^{(0)} = \mathcal{L}_0^{-1} S = -\frac{\beta}{32} W_0$
 - $\tilde{S}^{(1)} = \mathcal{L}_0^{-1} \sum_{x,\mu,a} \partial_{x,\mu}^a S \partial_{x,\mu}^a \tilde{S}^{(0)} = \frac{\beta^2}{192} \left(-\frac{4}{33} W_1 + \frac{12}{119} W_2 + \frac{1}{33} W_3 - \frac{5}{119} W_4 + \frac{3}{10} W_5 - \frac{1}{5} W_6 + \frac{1}{9} W_7 \right)$
 - etc... *Things become very difficult... very fast!*

Machine Learning approach

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

- $\vec{\theta}$ 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

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

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

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

 *Adjoint State*

- 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](https://arxiv.org/abs/2212.08469)

Learning Trivializing Gradient Flows for Lattice Gauge Theories

Simone Bacchio,¹ Pan Kessel,^{2,3} Stefan Schaefer,⁴ and Lorenz Vaitl²

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

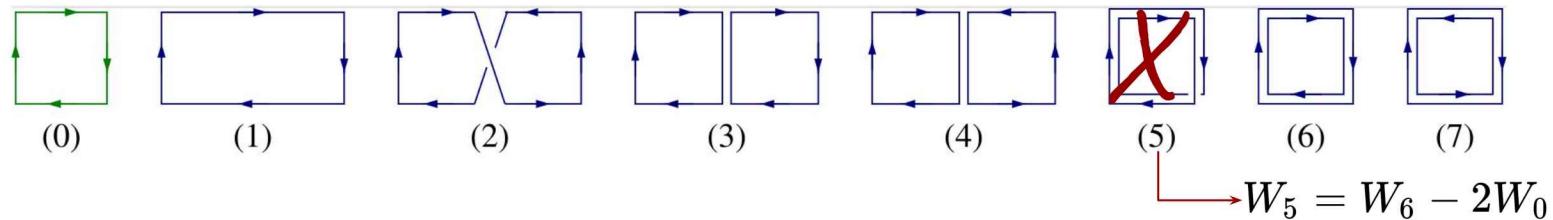
²*Machine Learning Group, Technische Universität Berlin, Berlin, Germany*

³*BIFOLD—Berlin Institute for the Foundations of Learning and Data, Berlin, Germany*

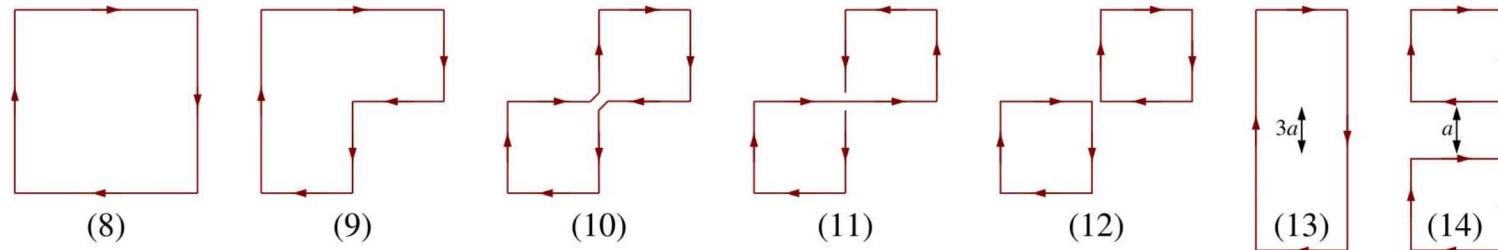
⁴*John von Neumann-Institut für Computing NIC,
Deutsches Elektronen-Synchrotron DESY, Germany*
(Dated: December 19, 2022)

Considered Flow Actions

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



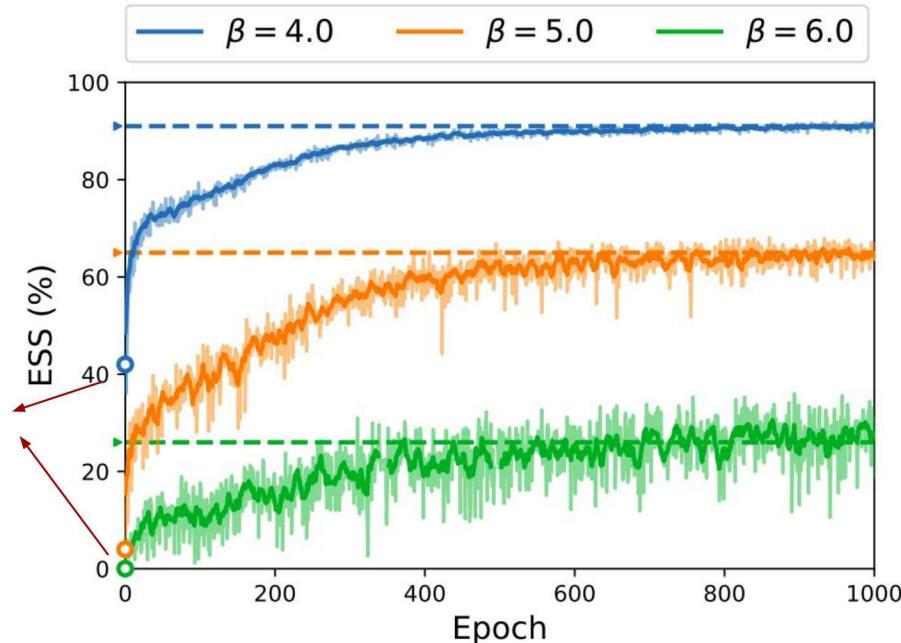
- **Model B:** 42 Wilson loops x 10 time points (interpolated by a cubic spline)



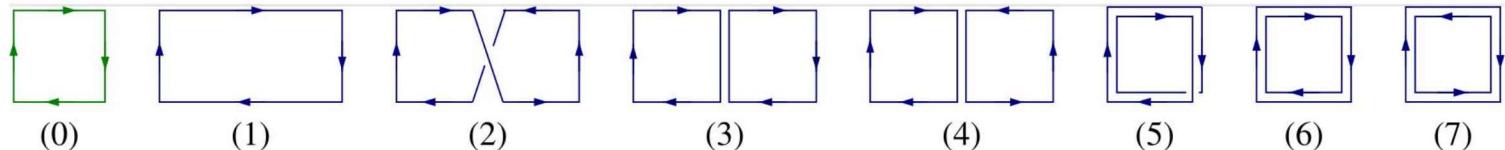
Results on a 16^2 lattice

Ref.	N_{params}	ESS at β		
		4.0	5.0	6.0
Lüscher, NL [3]	8 non-zero values	42%	4%	<1%
This work	A	91%	65%	26%
	B	98%	88%	70%
Boyda et al. [8]	$\mathcal{O}(10^6)$ estimated	88%	75%	48%

- We can use Lüscher's t-expansion as initial guess
- We can get compatible results to deep learning with as few as 14 parameters (vs 4M)
- Model B represents the state-of-the-art in 2D SU(3)

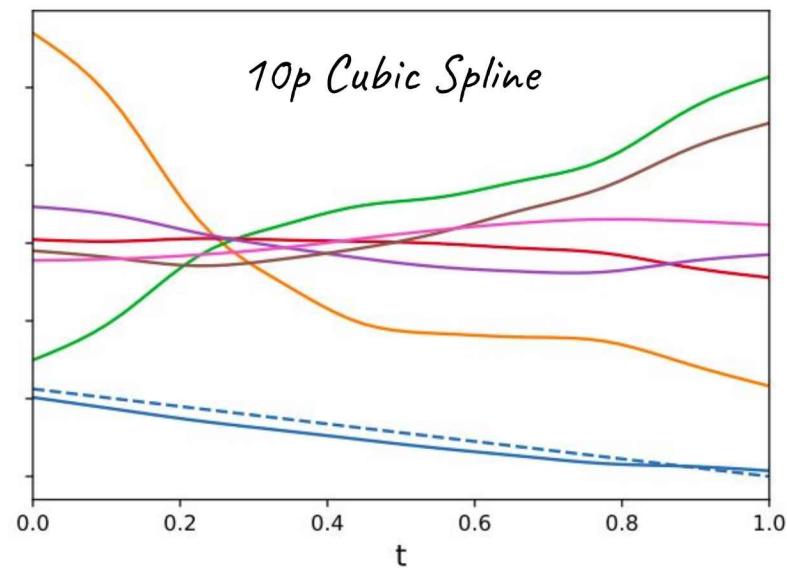
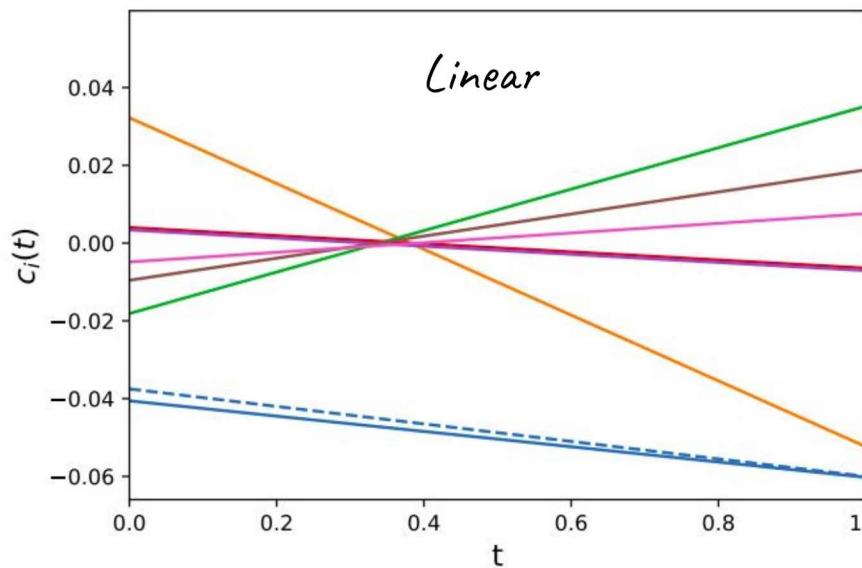


Time dependence of the coefficients is fundamental



Legend:

- $c_0 / 5$ (init) (dashed blue line)
- $c_0 / 5$ (solid blue line)
- c_1 (orange line)
- c_2 (green line)
- c_3 (red line)
- c_4 (purple line)
- c_6 (brown line)
- c_7 (pink line)



Summary of arXiv:2212.08469

- We have successfully applied the adjoint state method in learning gradient flows
- This is the first successful application of gradient flows as trivializing map since 2009
- We have set a new state-of-the-art in generative models for Yang-Mills theories with
 - Higher ESS
 - Less number of parameters (5 orders of magnitude)
 - Faster convergence thanks to initial guess (hundred iteration)
 - All symmetries preserved
- Does the approach solve the volume scaling issue? No, as far as we saw so far...
- What about 4D results? Working on it, an outlook follows...

Part 4: Outlook on 4D results

4 Dimensional $SU(3)$ Yang-Mills Theory



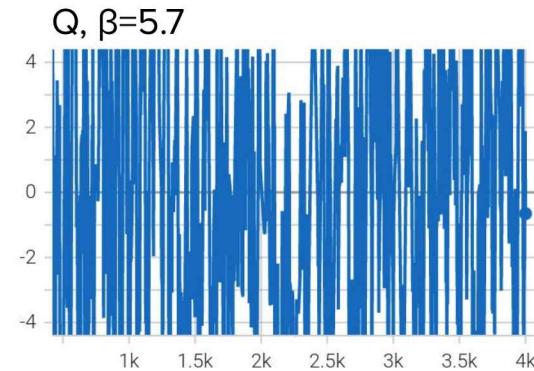
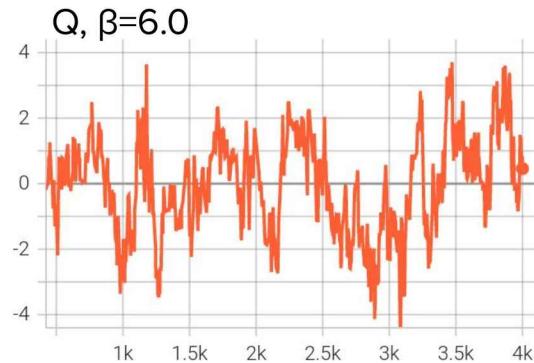
Target problem

Moving away from toy models, our current target problem is

- $\beta=6.0 \rightarrow 0.093$ fm lattice spacing [[1009.5228](#)]
- 16^4 lattice \rightarrow 1.5 fm lattice size
- Long autocorrelation already visible in the smeared Q

For the base distribution then we can consider

- Ideally, uniform \rightarrow most probably impossible
- Realistically, $\beta=5.7$ \rightarrow double lattice spacing
- Practically, ...



Flowing in β

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

Gradient
Flow

$$S_\tau(V) = S(U) - \int_\tau^1 dt \mathcal{L}_0 \tilde{S}(U_t, t) \quad \left\{ \begin{array}{l} \dot{U}_t = -\partial \tilde{S}_t U_t \\ V \equiv U_\tau, \quad U \equiv U_1 \end{array} \right.$$

Desired
base action

$$S_\tau = \tau S + C_\tau = -\tau \frac{\beta}{6} W_0 + C_\tau = -\frac{\beta_0}{6} W_0 + C_{\beta_0/\beta}$$



$\tau = 1 \quad \text{Target action}$
 $\tau = 0 \quad \text{Trivial action (constant)}$

$\tau = \frac{\beta_0}{\beta} \quad \text{Wilson action}$

Flowing in β

$$\frac{d}{d\tau} \left(S_\tau(V) = S(U) - \int_\tau^1 dt \mathcal{L}_0 \tilde{S}(U_t, t) \right) \quad \left\{ \begin{array}{l} \dot{U}_t = -\partial \tilde{S}_t U_t \\ V \equiv U_\tau, \quad U \equiv U_1 \end{array} \right.$$
$$\frac{\partial S_\tau}{\partial \tau} - (\partial S_\tau, \partial \tilde{S}_\tau) = \mathcal{L}_0 \tilde{S}_\tau \quad S_\tau = \tau S + C_\tau$$
$$Wilson action, \quad S = -\frac{\beta}{6} W_0 \quad S + \dot{C}_\tau - \tau (\partial S, \partial \tilde{S}_\tau) = \mathcal{L}_0 \tilde{S}_\tau$$
$$\mathcal{L}_0 \tilde{S}_\tau - \frac{\tau \beta}{6} (\partial W_0, \partial \tilde{S}_\tau) + \frac{\beta}{6} W_0 = \text{const.}$$

Flowing in β

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

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

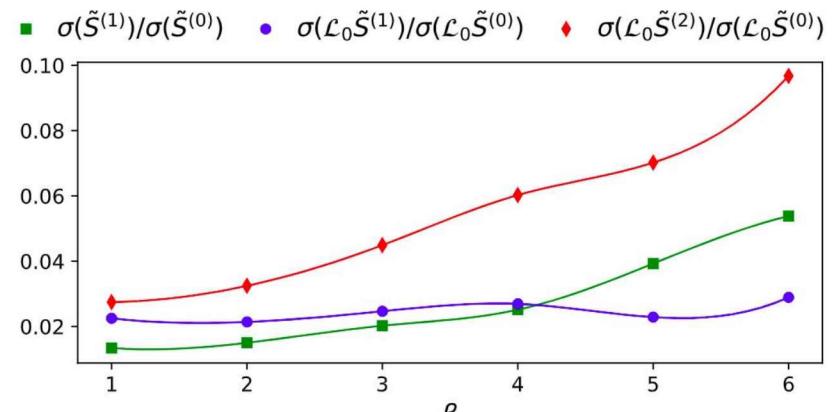
$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int D[U] \mathcal{O}(U) \exp(-S_\beta(U)) = \frac{1}{Z} \int D[V] \mathcal{O}(U) \exp(-S_{\beta_0}(V))$$

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

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 $\tilde{S}^{(k)}$??

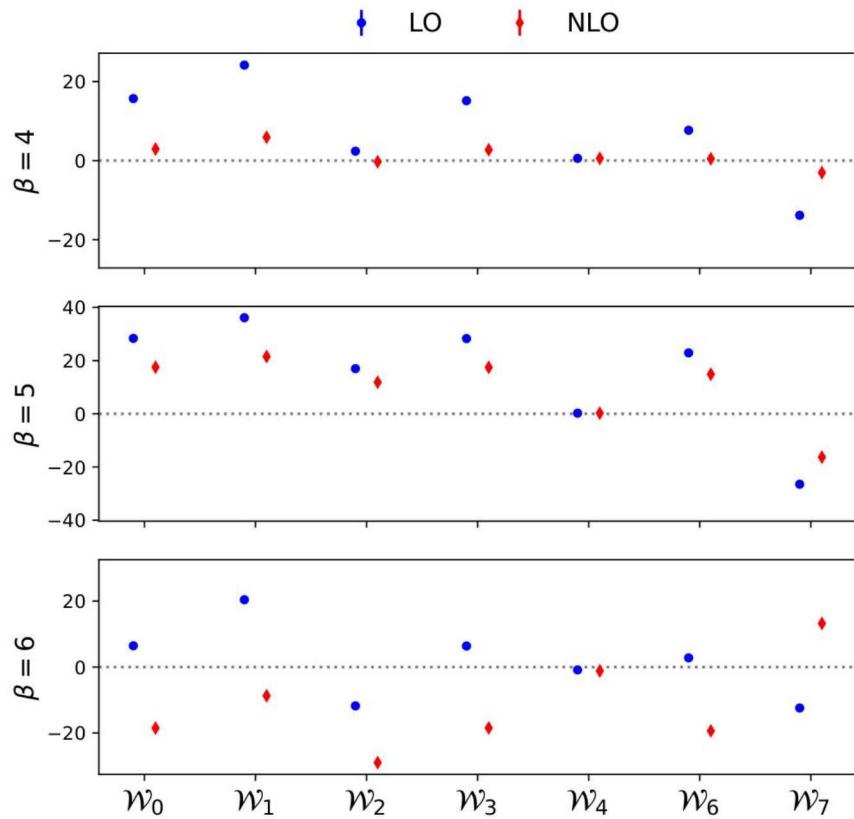
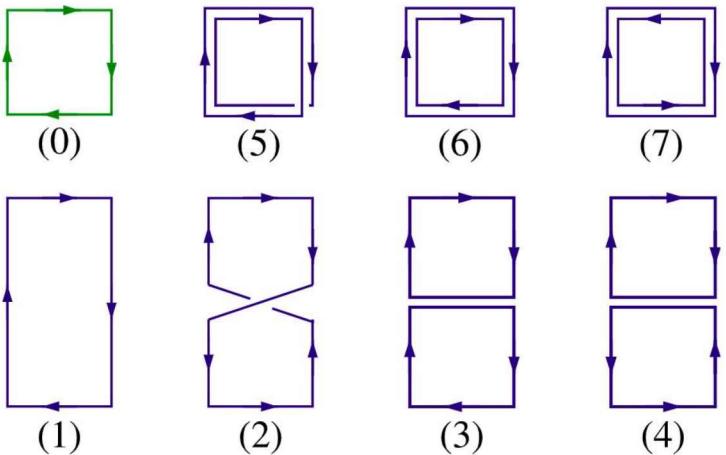


$$\mathcal{L}_0 \tilde{S}^{(2)} = (\partial W_0, \partial \tilde{S}^{(1)})$$

Quality of LO and NLO

$$\frac{dW_i}{d\beta} = (\partial W_i, \partial \tilde{S}) \xrightarrow{\text{Difference}}$$

iff \tilde{S} is the exact action for flowing in β



Conclusions

- It is very challenging to flow in β with target $\beta \gtrsim 6$
- The t-expansion suggests that the flow action is not local anymore
- Ways forwards?
 - Go higher in order in the t-expansion... More work needed, very expensive and when to stop?
 - Non-local terms in the flow action... Probably needed also when including fermions
 - Not flowing in β ... Looking for an action that is easy to simulate and easy to flow
 - Or extending the algorithm with new ideas... Many to try, which one is the best?

Learning Trivializing Gradient Flows

arXiv:2212.08469



Simone Bacchio

Computational Scientist
CaSToRC, The Cyprus Institute

A work in collaboration with
Pan Kessel, Stefan Schaefer, Lorenz Vaitl

Thank you for your attention!