Improved Fixed Point Actions from Gauge Equivariant Neural Networks

Andreas Ipp

Institute for Theoretical Physics, TU Wien, Austria

EuCAIFCon 2024
Amsterdam, May 2, 2024

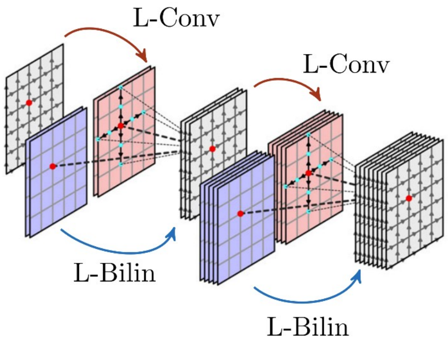
INSTITUTE FOR THEORETICAL PHYSICS

FWF Der Wissenschaftsfonds.

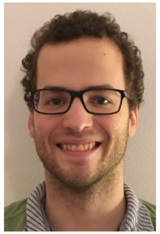
TU WIEN

Overview

Lattice gauge equivariant convolutional neural networks (L-CNNs)



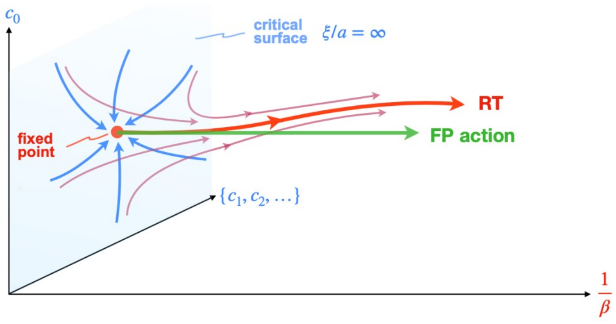
Favoni, AI, Müller, Schuh,
Phys.Rev.Lett. 128 (2022) 032003



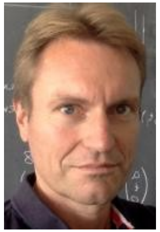
David Müller Matteo Favoni Daniel Schuh

Open source: https://gitlab.com/openpixi/lge-cnn

Learning fixed-point actions



Holland, AI, Müller, Wenger
arXiv:2401.06481



Kieran Holland (U. Pacific) Urs Wenger (U. Bern)

QCD on the lattice

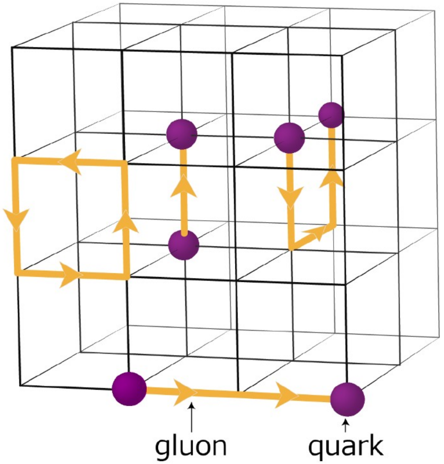
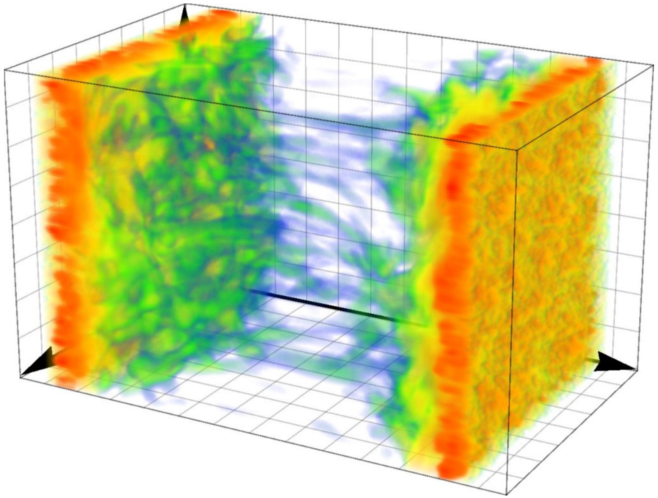


Image from Bi et al. EPJ Web Conf. 245 (2020) 09008

Solve Feynman path integral numerically using Monte Carlo methods

Gauge transformation:
Rotation in internal color space at each space-time point

Real-time lattice gauge theory



Used for describing dynamics of classical gluon fields at earliest stages of heavy ion collisions

AI, Müller, Eur.Phys.J.A 56 (2020) 9, 243
AI, Müller, Eur.Phys.J. C78 (2018) no.11, 884
AI, Müller, Phys. Lett. B 771 (2017) 74
Gelfand, AI, Müller, Phys. Rev. D94 (2016) no.1, 014020

Wilson action for gluons:

$$S_W[U] = \frac{2}{g^2} \sum_{x\in\Lambda} \sum_{\mu<\nu} \mathrm{Tr}\left[\mathbb{1} - U_{x,\mu\nu}\right]$$

Gauge transformation of link variables:

$$U_\mu(n) \;\rightarrow\; U'_\mu(n) = \Omega(n)\, U_\mu(n)\, \Omega(n+\hat{\mu})^\dagger$$

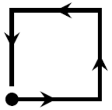
Examples of Wilson loops on the lattice

Wilson action

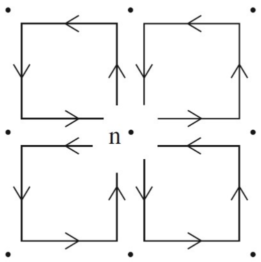
$$S_W[U] = \frac{2}{g^2} \sum_{x \in \Lambda} \sum_{\mu<\nu} \mathrm{Tr}\left[\mathbb{1} - U_{x,\mu\nu}\right]$$

Plaquette

$$U_{x,\mu\nu} = U_{x,\mu} U_{x+\mu,\nu} U^\dagger_{x+\nu,\mu} U^\dagger_{x,\nu} =$$

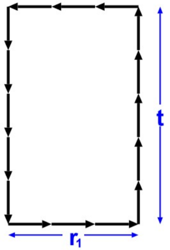


Symanzik improved clover action



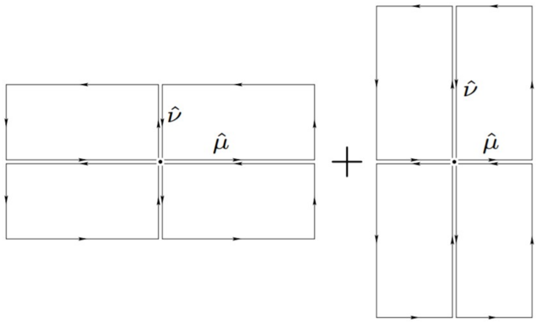
from: Gattringer, Lang (2010)

Potential of static quark pair



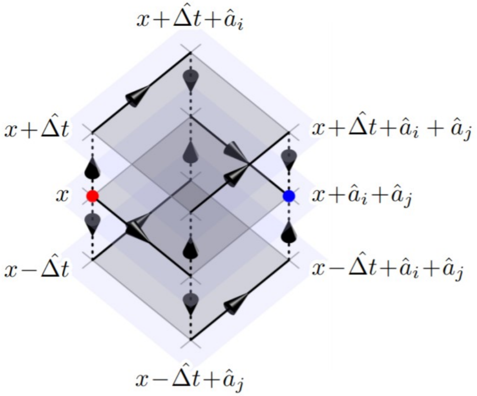
from: Bali, Phys.Rept. 343:1 (2001)

Improved topological charge



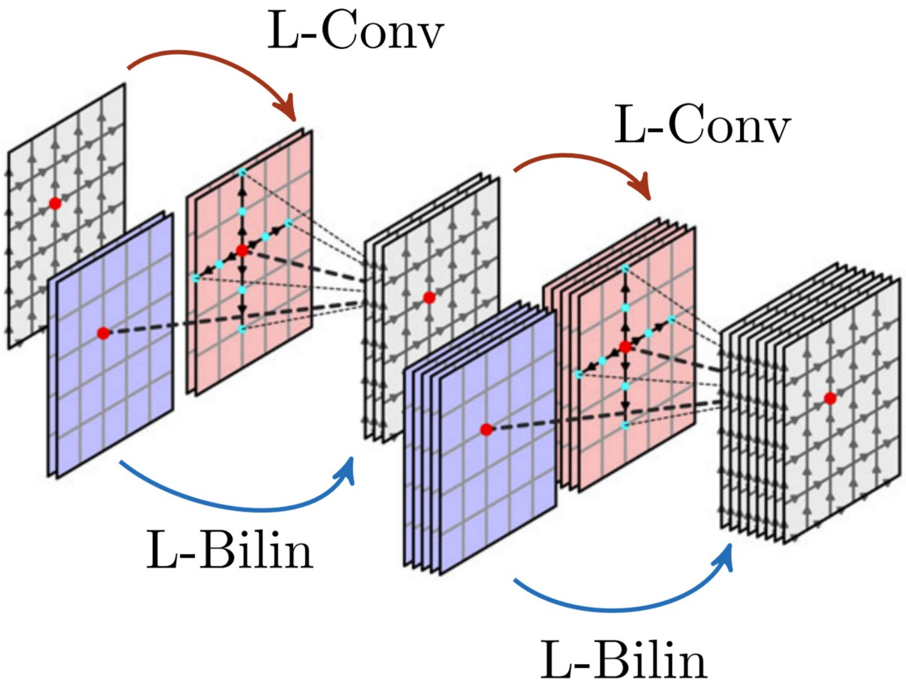
from: Alexandrou et al., Eur.Phys.J.C 80 (2020) 5, 424

Improved real-time lattice actions



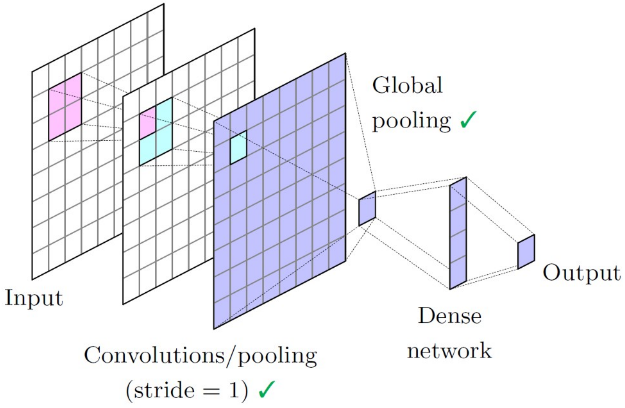
AI, Müller, Eur.Phys.J. C78 (2018) no.11, 884

Lattice gauge symmetry



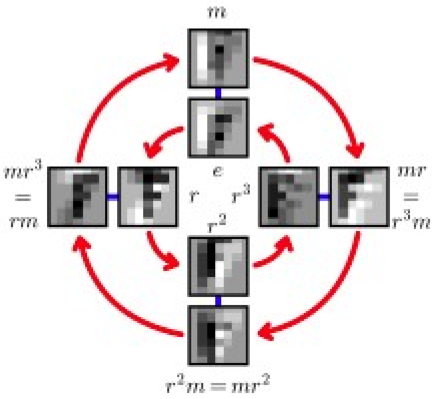
Symmetries on the lattice

Translational symmetry
→ Convolutional neural networks (CNNs)



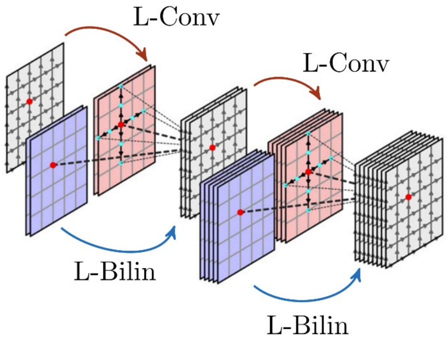
Bulusu, Favoni, AI, Müller, Schuh,
Phys. Rev. D 104 (2021) 074504

Rotation, mirror symmetry
→ Group equivariant CNNs (G-CNNs)



Cohen, Welling, ICML 2016

Lattice gauge symmetry
→ Lattice gauge equivariant CNNs (L-CNNs)



Favoni, AI, Müller, Schuh,
Phys.Rev.Lett. 128 (2022) 032003

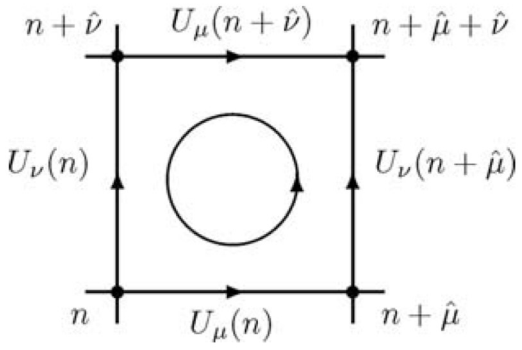
L-CNN data

Combine lattice links *U*
and locally transforming objects *W*

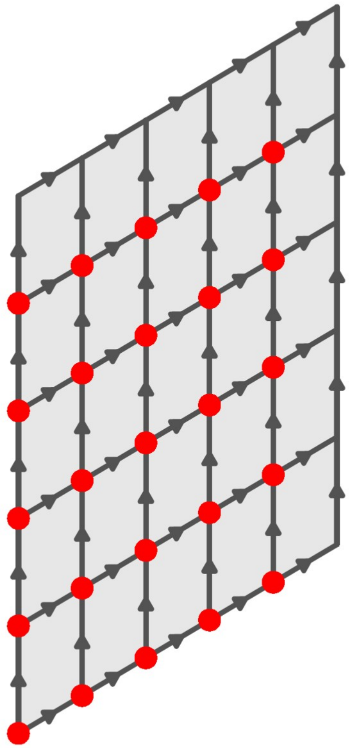
$$\text{tuple} \ \ (\mathcal{U}, \mathcal{W})$$

$$\mathcal{U} \ = \ \{U_{x,\mu}\} \ \ \mathrm{SU}(N_c) \text{ matrices}$$

$$\mathcal{W} \ = \ \{W_{x,i}\} \ \text{ with } \ W_{x,i} \ \in \ \mathbb{C}^{N_c \times N_c}$$



from: Gattringer, Lang (2010)



$$\mathcal{U} = \{U_{\mathbf{x},\mu}\}$$

$$\mathcal{W} = \{W_{\mathbf{x},\mu\nu}\}$$

Gauge transformation

$$T_\Omega U_{x,\mu} = \Omega_x U_{x,\mu} \Omega^\dagger_{x+\mu}$$

$$T_\Omega W_{x,i} = \Omega_x W_{x,i} \Omega^\dagger_x$$

Gauge equivariant (gauge covariant) function

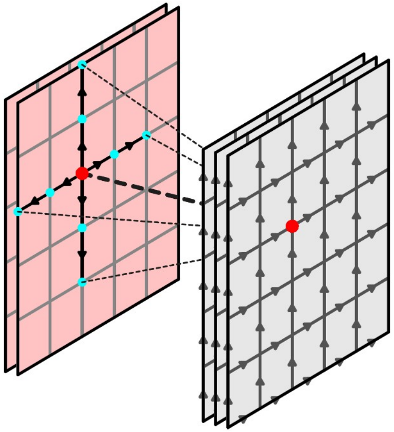
$$f(T_\Omega \mathcal{U}, T_\Omega \mathcal{W}) \ = \ T'_\Omega f(\mathcal{U}, \mathcal{W})$$

Gauge invariant function

$$f(T_\Omega \mathcal{U}, T_\Omega \mathcal{W}) \ = \ f(\mathcal{U}, \mathcal{W})$$

Lattice gauge equivariant layers

Convolution (L-Conv)

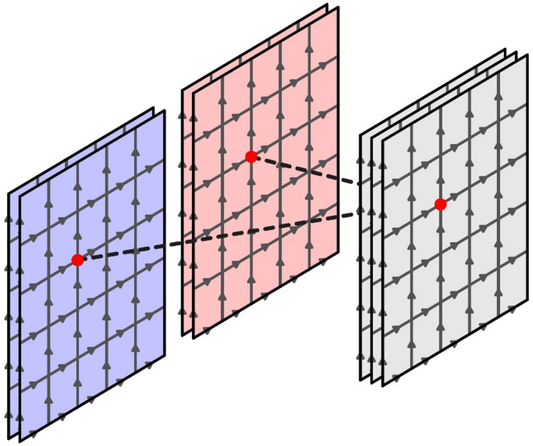


Convolution wish shared weights and proper parallel transport along coordinate axes

$$(\mathcal{U}, \mathcal{W}) \to (\mathcal{U}, \mathcal{W}')$$

$$W'_{\mathbf{x},i} = \sum_{j,\mu,k} \omega_{i,j,\mu,k} U_{\mathbf{x},k\cdot\mu} W_{\mathbf{x}+k\cdot\mu,j} U^{\dagger}_{\mathbf{x},k\cdot\mu}$$

Bilinear layer (L-Bilin)

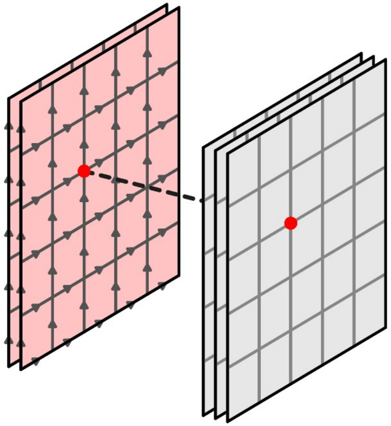


Multiply *W* at each lattice point

$$(\mathcal{U}, \mathcal{W}) \times (\mathcal{U}, \mathcal{W}') \to (\mathcal{U}, \mathcal{W}'')$$

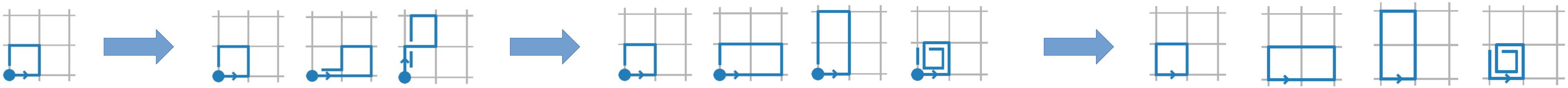
$$W''_{\mathbf{x},i} = \sum_{j,k} \alpha_{ijk} W_{\mathbf{x},j} W'_{\mathbf{x},k}$$

Trace layer

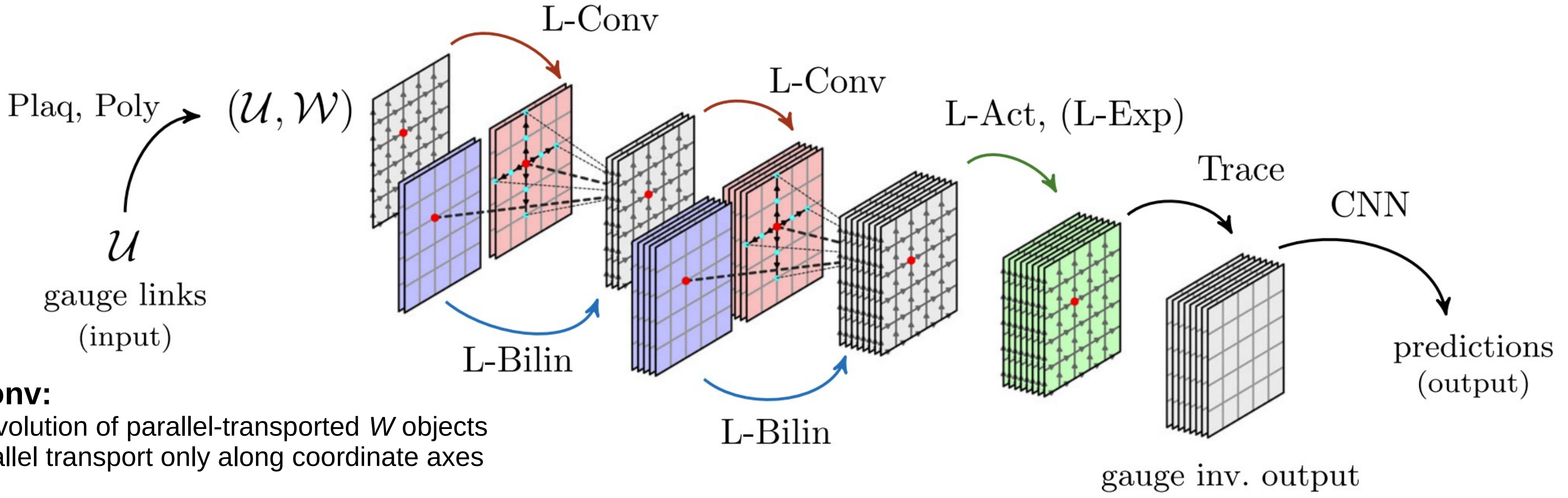


Generate gauge invariant output

$$w_{\mathbf{x},i} = \mathrm{Tr}\, W_{\mathbf{x},i} \in \mathbb{C}$$



Generic L-CNN



L-Conv:
* convolution of parallel-transported *W* objects
* parallel transport only along coordinate axes

L-Bilin:
* bilinear layer, product of locally transforming objects

L-Act:
* activation functions multiply *W* objects by scalar, gauge-invariant functions

L-Exp:
* update link variables using exponential map

Trace:
* calculate gauge invariant trace

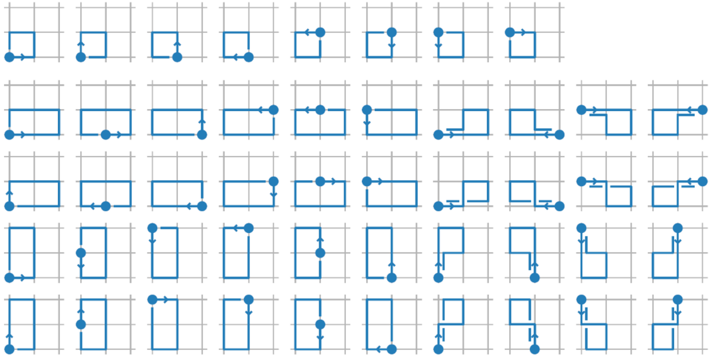
Plaq:
* generate all possible plaquettes

Poly:
* generate all possible Polyakov loops

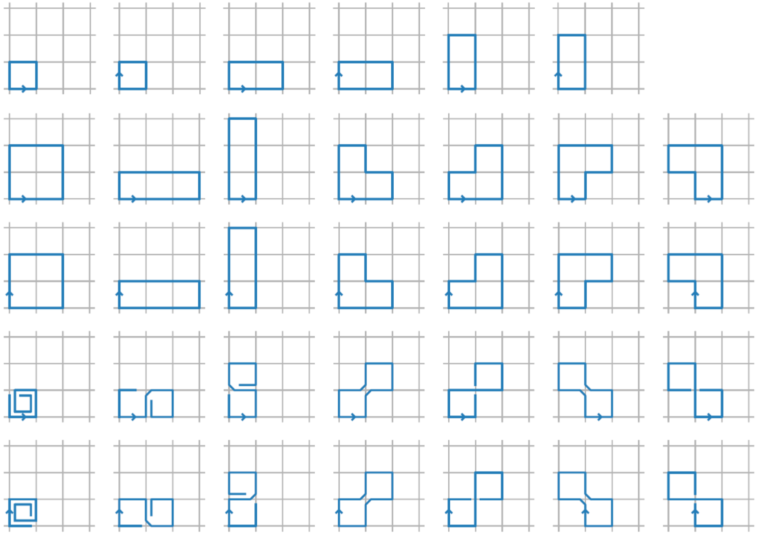
Favoni, AI, Müller, Schuh,
Phys.Rev.Lett. 128 (2022) 032003

L-CNNs generate Wilson loops

Untraced loops



Traced loops



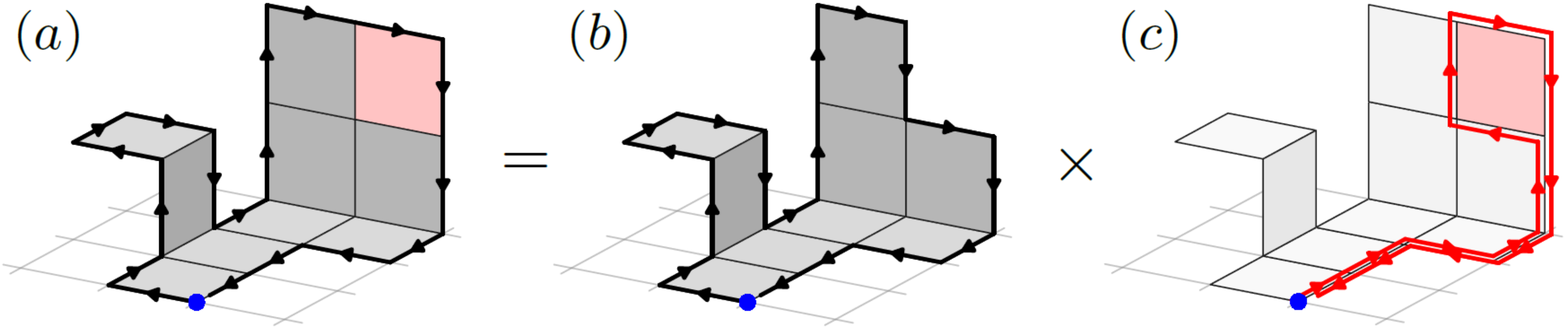
Number of traced Wilson loops covered by L-CNN architectures of various sizes in 1+1 D

Length	Max	$W^{(1\times1)}$	$W^{(1\times2)}$			$W^{(2\times2)}$		
		S	S	M	L	S	M	L
0	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0
4	2	2	2	2	2	2	2	2
6	4		4	4	4	4	4	4
8	28		4	4	4	22	22	22
10	152			8	8	48	76	76
12	1,010				8	92	204	220
14	6,772					120	412	532
16	47,646					100	712	1,080
18	343,168					136	928	1,896
20	2,529,890					32	1,056	2,620
22	18,982,172					64	768	3,152
$\geq$24							800	7,210
Total		3	11	19	27	621	4,985	16,725
Max.Len		4	8	10	12	22	28	34

Architectures differ in number of layers, kernel size, and number of channels.

Favoni, AI, Müller, Schuh, Phys.Rev.Lett. 128 (2022) 032003

Sketch of proof for arbitrary Wilson loops



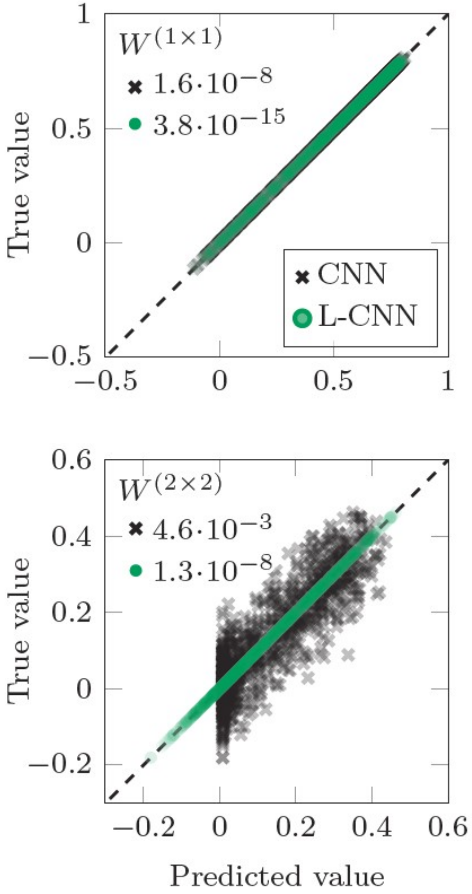
Favoni, AI, Müller, Schuh, Phys.Rev.Lett. 128 (2022) 032003

(a) An arbitrary contractible Wilson loop of *n* tiles ...
(b) … is composed (L-Bilin) of a Wilson loop of (*n*-1) tiles …
(c) … and a parallel-transported (L-Conv) plaquette (Plaq).

Non-contractible loops (like Polyakov loops) have to be added (Poly).

Numerical results

1+1D



Regression task to learn value of rectangular Wilson loops:

$$W_{x,\mu\nu}^{(m\times n)} = \mathrm{Re}\,\mathrm{Tr}\left[U_{x,\mu\nu}^{(m\times n)}\right]$$

Lattice gauge equivariant CNN (L-CNNs, green) can learn the relation, while traditional convolutional neural networks (CNNs, black) struggle to find the solution.

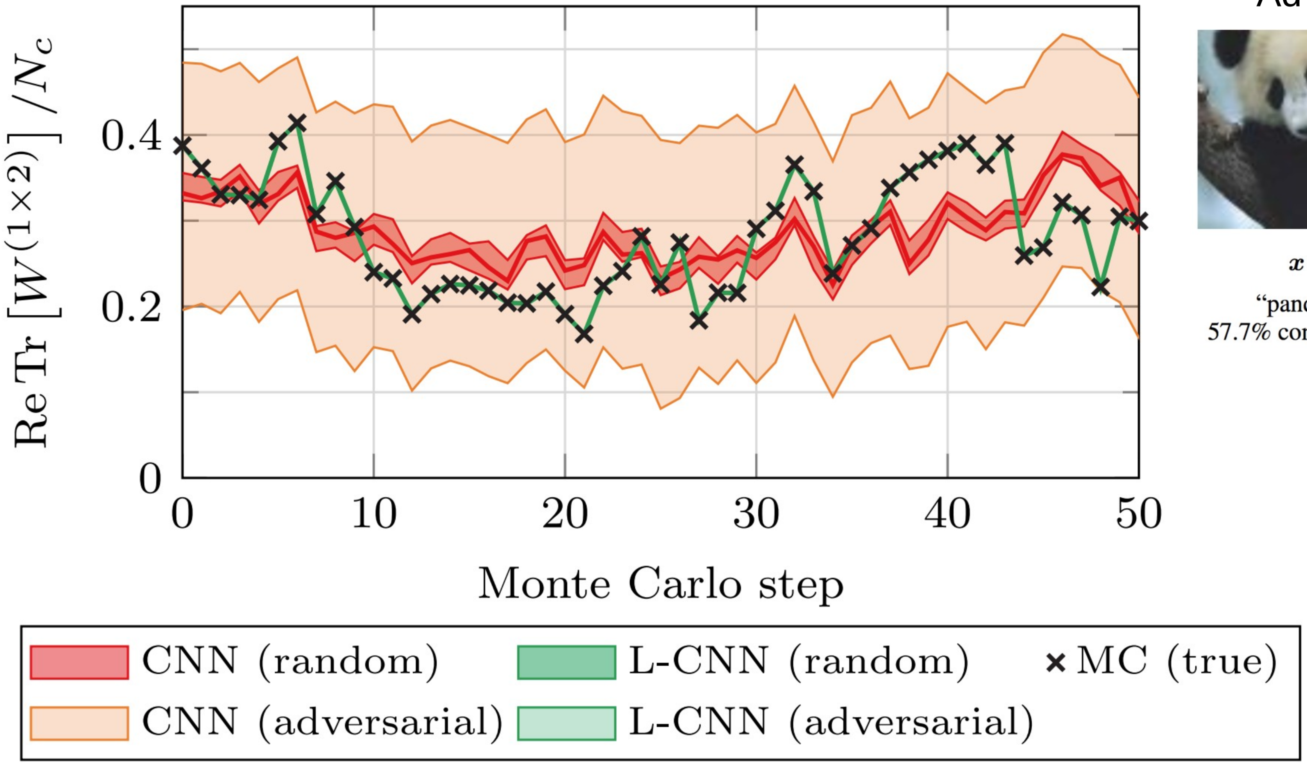
Training on 8 × 8, testing from 8 × 8 up to 64 × 64

Compared best from:
100 L-CNN models (10 – 10^4 trainable parameters, up to 4 L-Conv+L-Bilin)

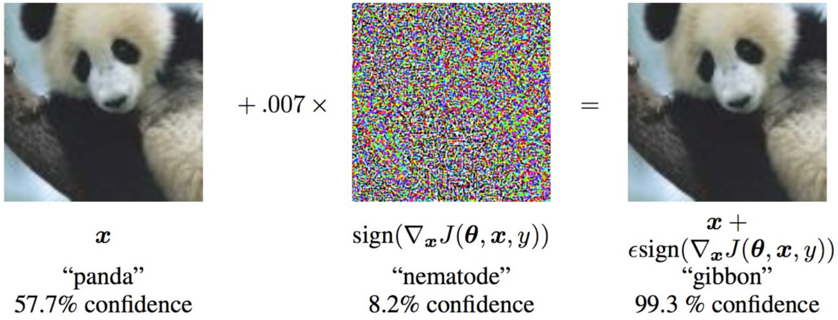
2840 CNN models (100 – 10^5 trainable parameters up to 6 layers, 512 channels, 4 activation functions)

Favoni, AI, Müller, Schuh, Phys.Rev.Lett. 128 (2022) 032003

Adversarial attacks



Adversarial attack:

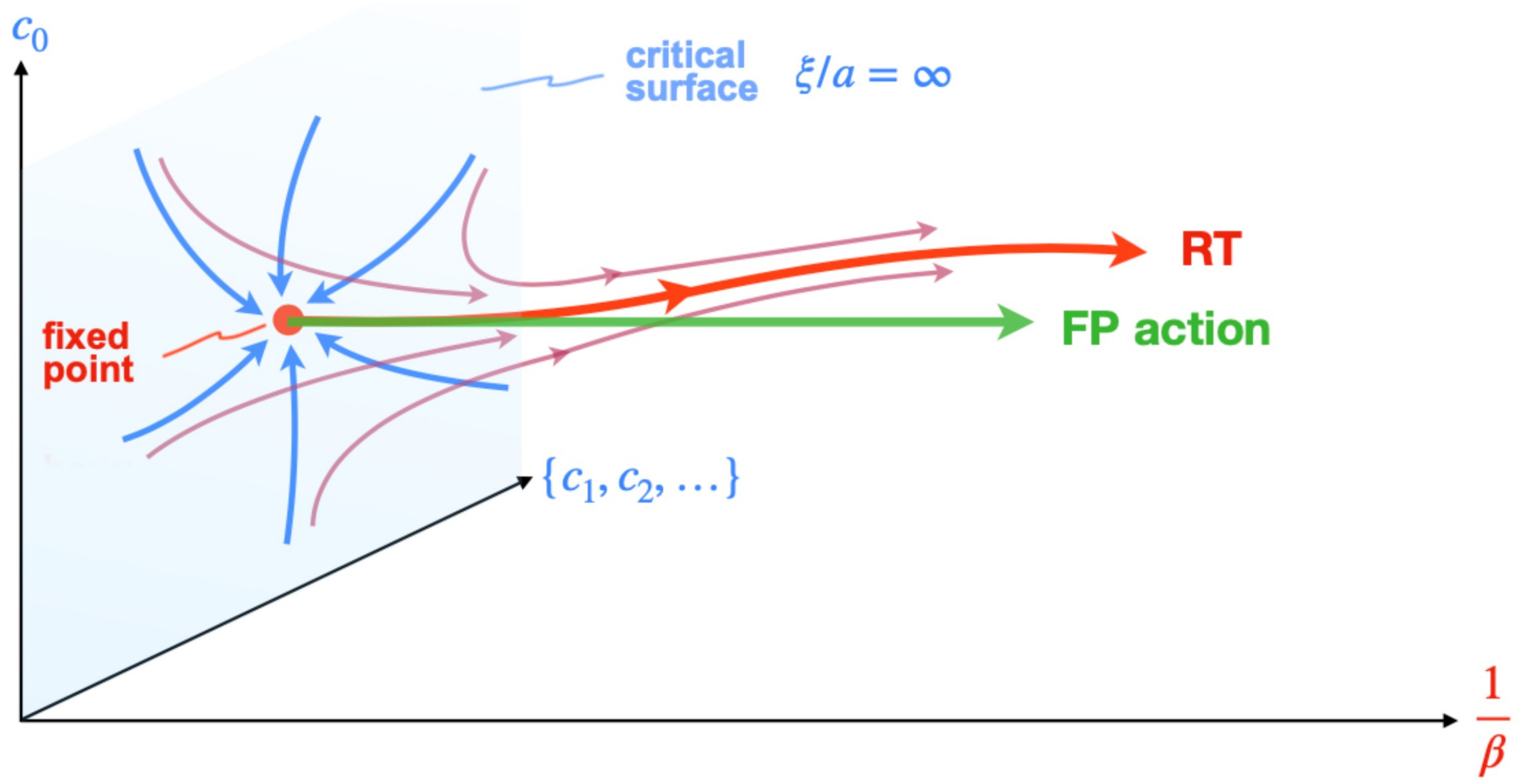


From Goodfellow, Shlens, Szegedy ICLR 2015

L-CNNs are insensitive to random or adversarial gauge transformations

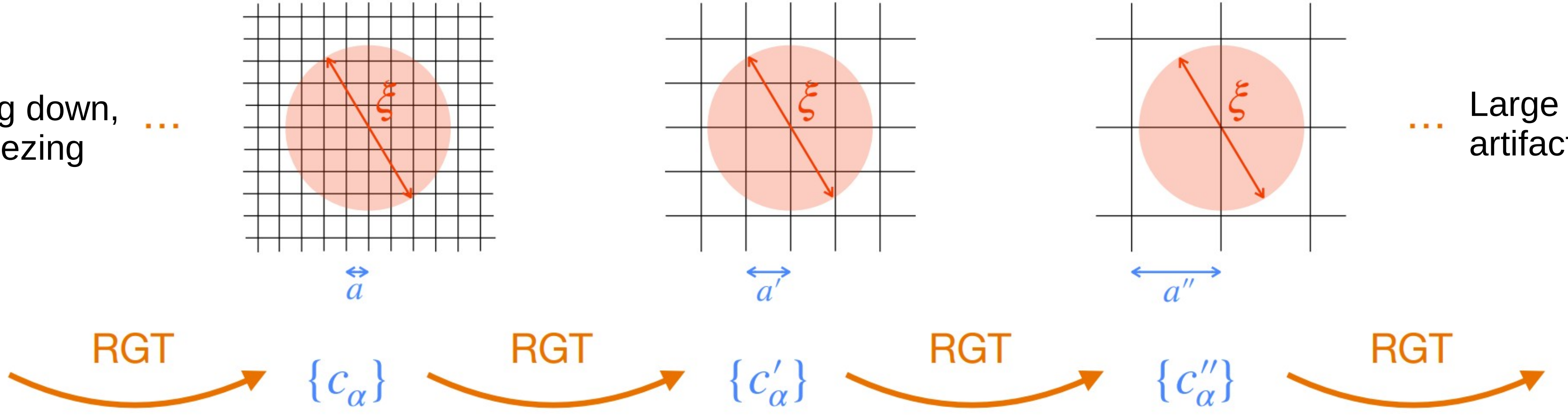
Favoni, AI, Müller, Schuh, Phys.Rev.Lett. 128 (2022) 032003

Fixed point action



Renormalization group transformation

Critical slowing down, topological freezing …



… Large lattice artifacts

Introduce a (real space) renormalization group transformation (RGT)

$$\exp\{-\beta' A'[V]\} = \int \mathcal{D}U \exp\{-\beta(A[U] + T[U,V])\}$$

Blocking kernel

The effective action $\beta' A'[V]$ is described

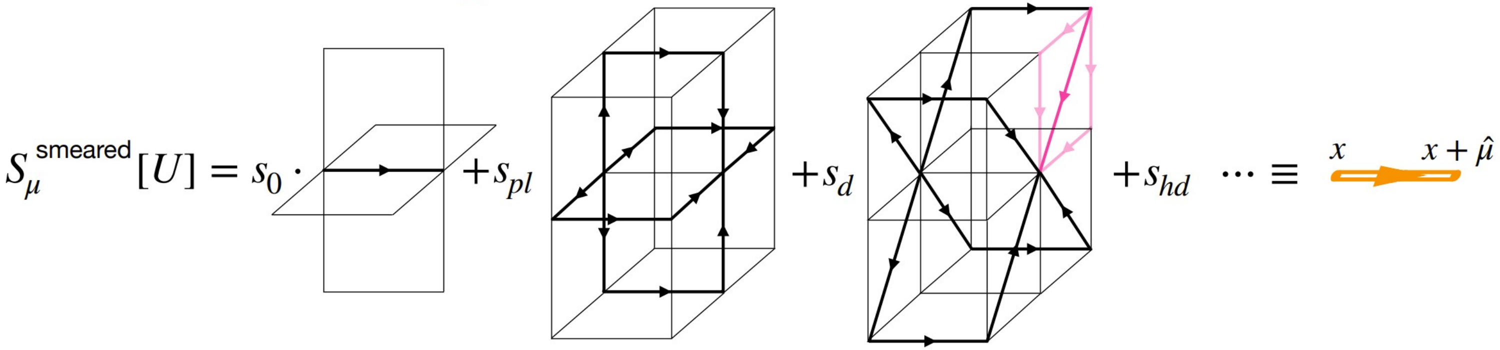
by infinitely many couplings $\{c'_\alpha\}$

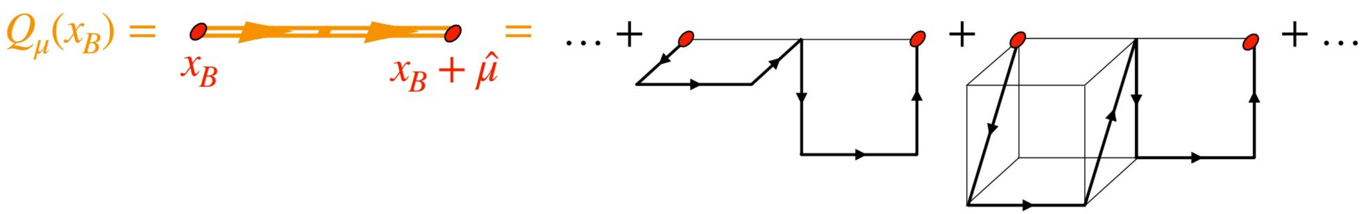
The fixed point is the saddle point in the classical limit $\beta \to \infty$, which can be found by a minimization condition.

P. Hasenfratz, F. Niedermayer, Nucl.Phys.B 414 (1994) 785

Blocking kernel

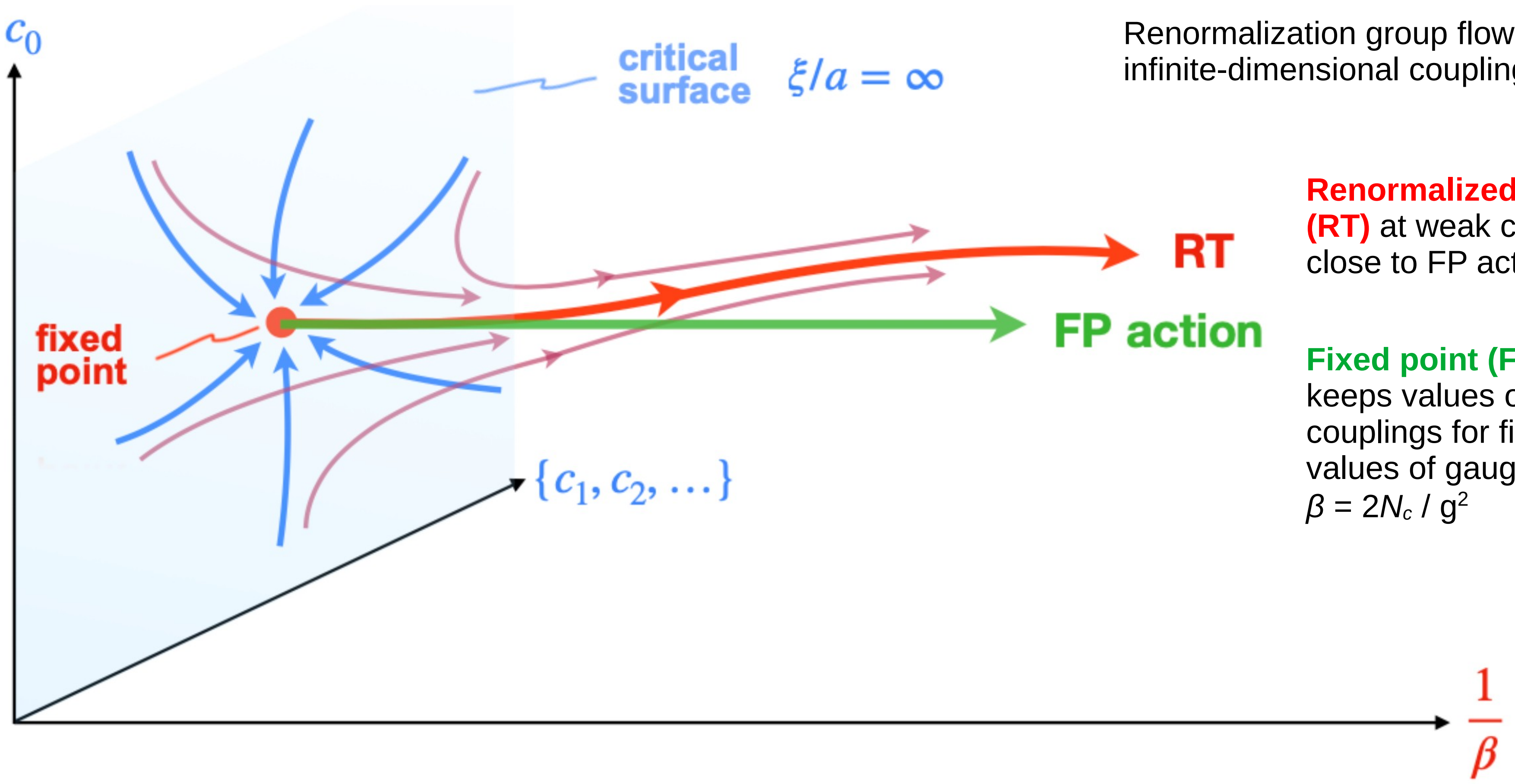
$$T[U,V] = -\frac{\kappa}{N_c} \sum_{x_B,\mu} \left\{ \mathrm{ReTr}\left(V_\mu(x_B) \cdot Q_\mu^\dagger(x_B) \right) - \mathcal{N}_\mu^\beta \right\}$$





Choice of blocking kernel determines how couplings are modified across scales.

Renormalization group transformation and Fixed point action



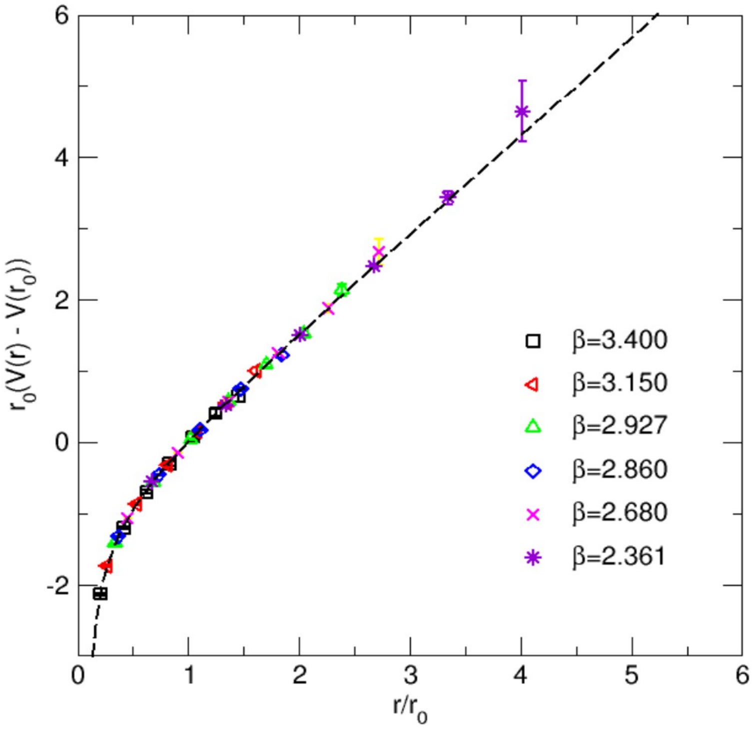
Renormalization group flow in infinite-dimensional coupling space.

Renormalized trajectory (RT) at weak coupling close to FP action.

Fixed point (FP) action keeps values of critical couplings for finite values of gauge coupling $\beta = 2N_c / g^2$

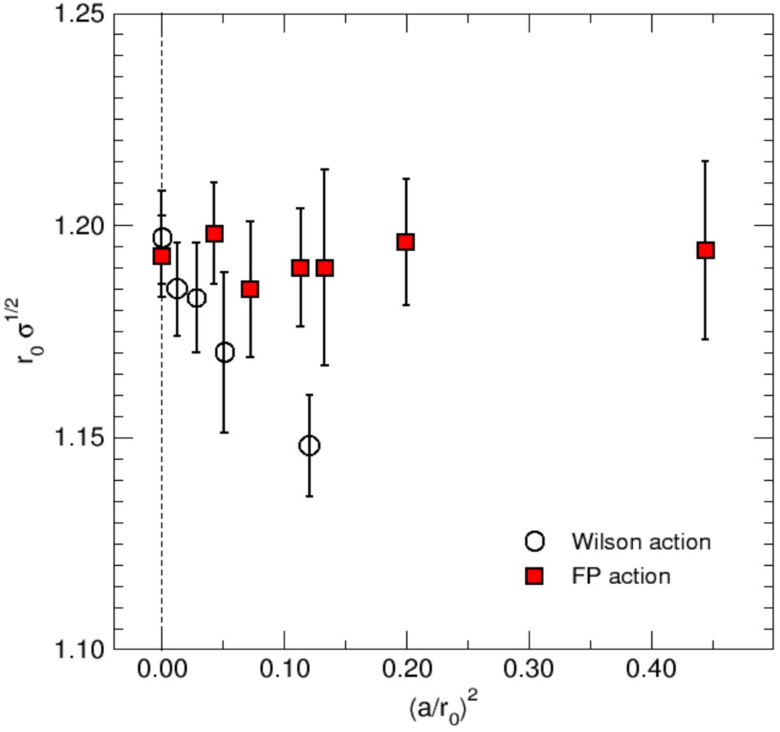
Fixed point action using older parametrizations

Static quark-antiquark potential



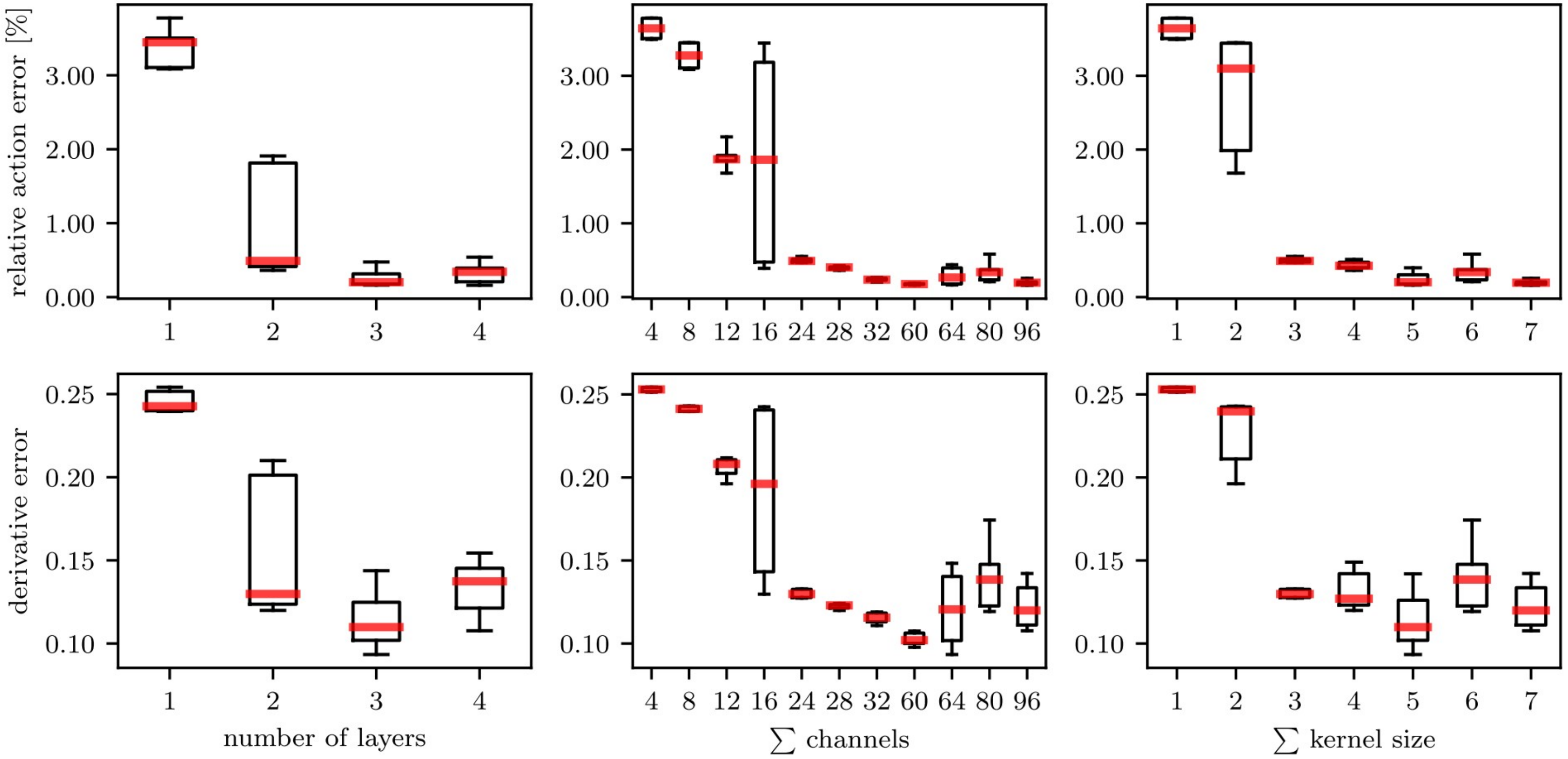
fit to $V(r) = V_0 - \frac{\alpha}{r} + \sigma r$

Extraction of string tension σ



Niedermayer, Rüfenacht, Wenger, Nucl.Phys.B 597 (2001) 413, hep-lat/0007007

Scan through various architectures



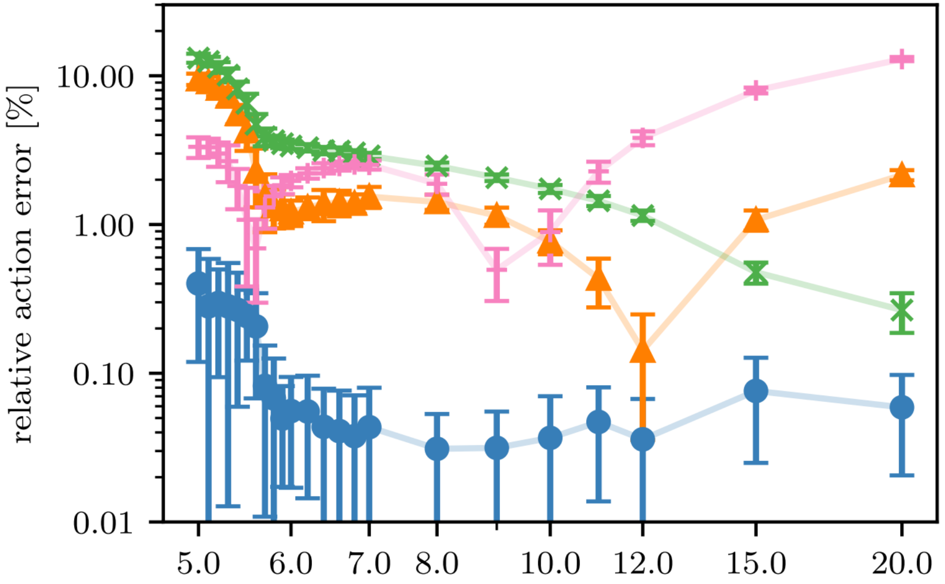
Supervised learning from coarse configurations and corresponding minimized action values on fine configurations.

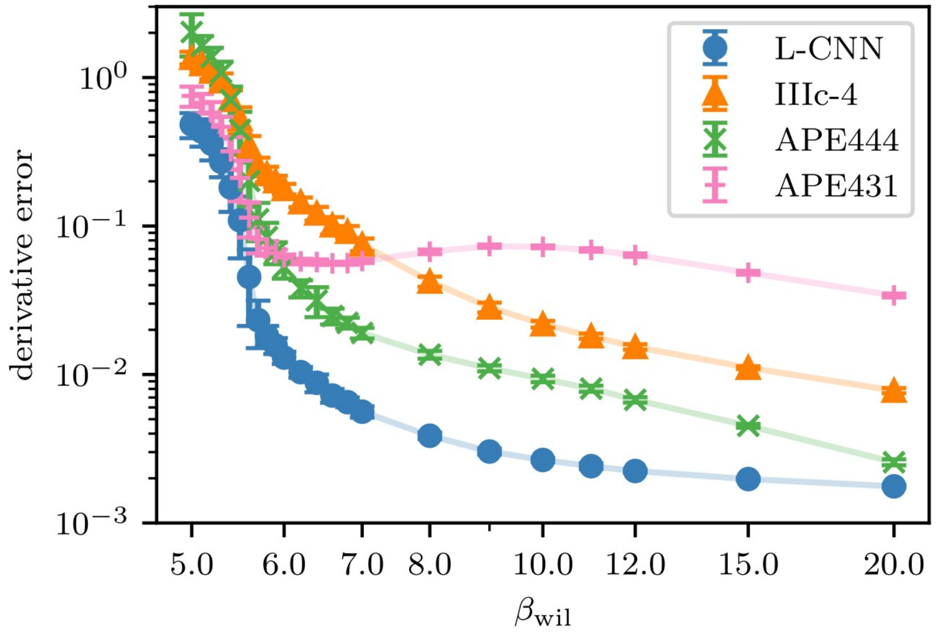
Also use derivatives of fixed point action for learning.

Train 130 models of various sizes for 4^4 lattice, SU(3) gauge group, and $\beta_{\mathrm{wil}} \in [5,10]$.

Holland, AI, Müller, Wenger, arXiv:2401.06481

Learning the fixed point action with L-CNNs





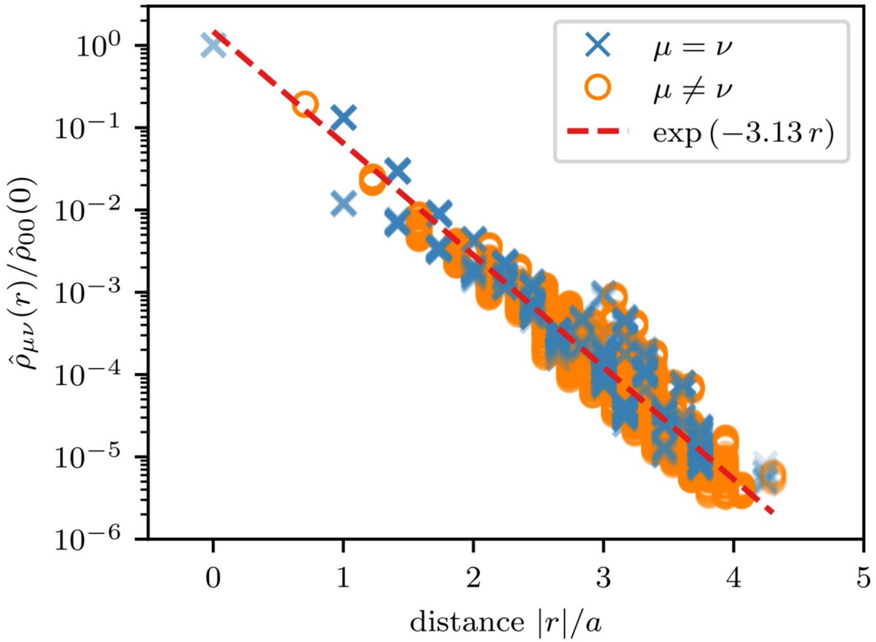
Best model: L-CNN with
3 layers with 12, 24, 24 channels
and kernel size 2, 2, 1.

L-CNN superior to older parametrizations of FP action.

Holland, AI, Müller, Wenger
arXiv:2401.06481

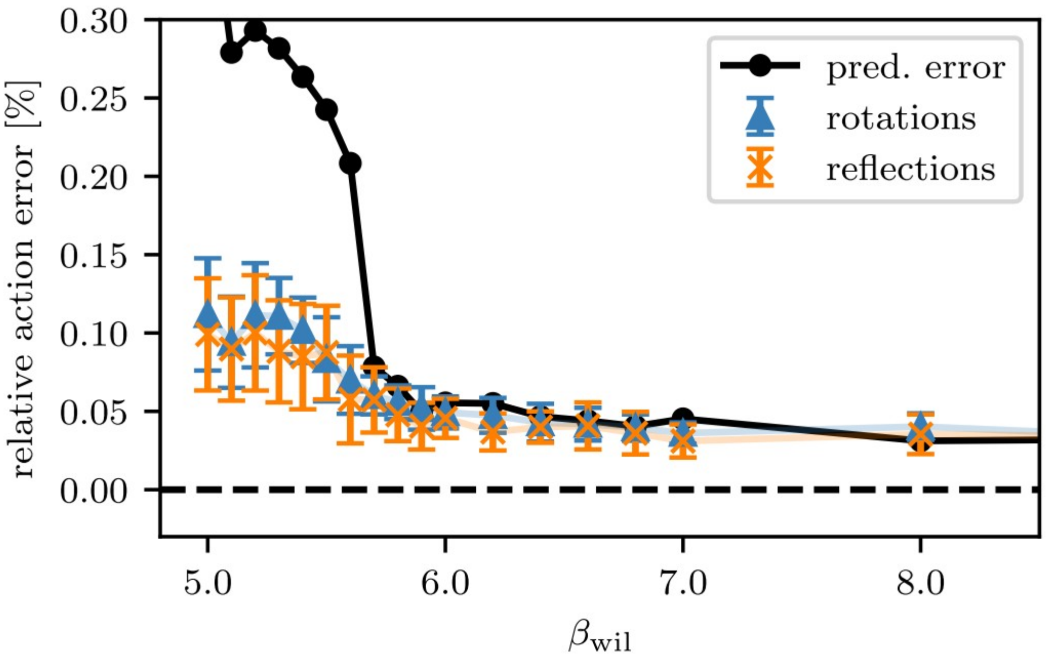
Properties of the learned FP action

Locality measurement



Couplings of learned fixed point actions decrease exponentially with separation.

Rotation and reflection error

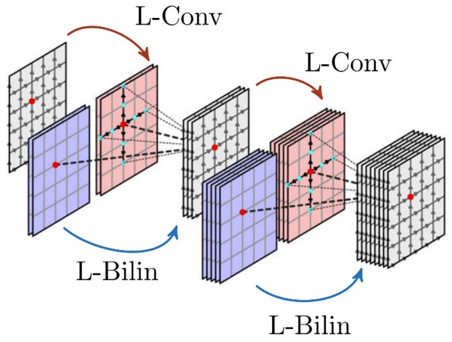


Learned symmetry errors are smaller than or comparable to prediction errors.

Holland, AI, Müller, Wenger
arXiv:2401.06481

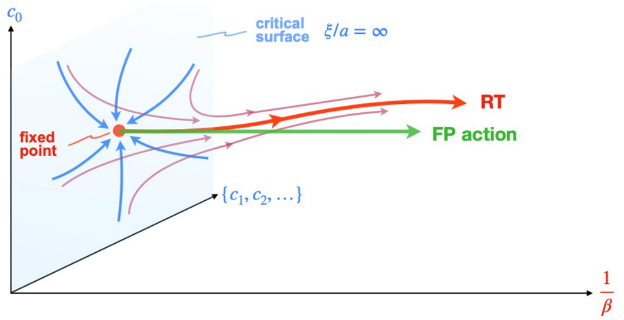
Summary

Lattice gauge equivariant convolutional neural networks (L-CNNs)



Favoni, AI, Müller, Schuh,
Phys.Rev.Lett. 128 (2022) 032003

Learning fixed-point actions



Holland, AI, Müller, Wenger
arXiv:2401.06481

- Fixed-point actions may enable calculations at coarser lattice spacing
- L-CNNs achieve higher accuracy than previous hand-crafted parametrizations
- Outlook: apply learned FP action to Monte Carlo simulations

Open source: https://gitlab.com/openpixi/lge-cnn

Backup

Machine learning the fixed point action

To obtain the training data for supervised machine learning, first generate ensembles of coarse configurations *V.*

For a given coarse configuration *V*, the fixed point action values are determined by minimizing configurations *U*, *U'*, ...

$$A^{\text{FP}}[V] = \min_{\{U\}} \{A^{\text{FP}}[U] + T[U,V]\} = \min_{\{U',U\}} \{A^{\text{FP}}[U'] + T[U',U] + T[U,V]\}$$

Use additional information for training obtained from derivatives of the fixed point action:

$$\frac{\delta A^{\text{FP}}[V]}{\delta V^a_{x,\mu}} = \frac{\delta T[U,V]}{\delta V^a_{x,\mu}} = -\kappa \, \text{Re Tr}(it^a V_{x,\mu} Q^\dagger_{x,\mu}) \qquad Q^\dagger_{x,\mu} = Q^\dagger_{x,\mu}[U]$$

yields *D* [link directions] x (N^2 - 1) [colors] x L^D [lattice sites] data per configuration.

Fixed point action using L-CNNs

Parametrize **action model** in particular way:

$$\mathcal{A}^{\text{L-CNN}}[V] = \sum_x \mathcal{A}_x^{\text{pre}}[V] \sum_{n=0}^{\infty} b^{(n)} (N_x[V] - N_x[\mathbb{1}])^n$$

Prefactor controls continuum behavior

L-CNN

Loss function combines action values and its derivatives $\mathcal{L} = w_1 \mathcal{L}_1 + w_2 \mathcal{L}_2$

$$\mathcal{L}_1 = \frac{1}{L^4 N_{\text{cfg}}} \sum_{i=1}^{N_{\text{cfg}}} |\mathcal{A}^{\text{FP}}[V_i] - \mathcal{A}^{\text{L-CNN}}[V_i]|,$$

$$\mathcal{L}_2 = \frac{1}{32 L^4 N_{\text{cfg}}} \sum_{i=1}^{N_{\text{cfg}}} \sum_{x,\mu} \text{Tr}\left[(D_{x,\mu}^{\text{FP}}[V_i] - D_{x,\mu}^{\text{L-CNN}}[V_i])^2 \right]$$

Technical remark: derivatives of L-CNNs are obtained through backpropagation

Continuous formulation of L-CNNs

Define a continuous version of a gauge equivariant convolution:

$$[\psi * \mathcal{W}]^a(\mathbf{x}) = \sum_b \int_{\mathbb{R}^D} \mathrm{d}\mathbf{y}^D \, \psi^{ab}(\mathbf{y}-\mathbf{x}) U_{\mathbf{x}\to\mathbf{y}} W^b(\mathbf{y}) U^\dagger_{\mathbf{x}\to\mathbf{y}}$$

with kernel components $\psi^{ab} : \mathbb{R}^D \to \mathbb{R}$

and parallel transporter $U_{\mathbf{x}\to\mathbf{y}} = \mathcal{P} \exp \left\{ i \int_0^1 \mathrm{d}s \frac{\mathrm{d}x^\nu(s)}{\mathrm{d}s} A_\nu(x(s)) \right\}$

that map $\mathcal{W} = (W^1, \ldots, W^m)$ objects to new objects

in a gauge equivariant manner:

$$[\psi * T_\Omega \mathcal{W}]^a(\mathbf{x}) = T_\Omega [\psi * \mathcal{W}]^a(\mathbf{x})$$

Similarly define continuous bilinear layer, trace layer, …

Discretize this to obtain previous formulation.

Compatible with G-CNNs.

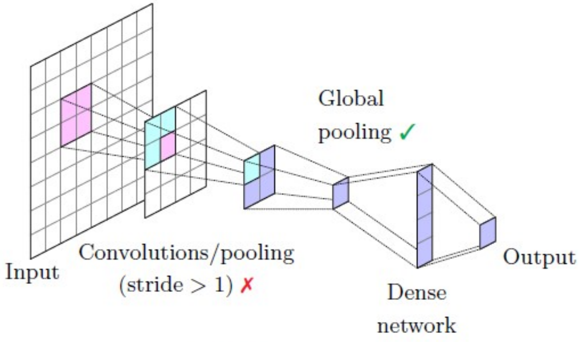
Generalizable to vectors and tensors.

Aronsson, Müller, Schuh, arxiv:2303.11448

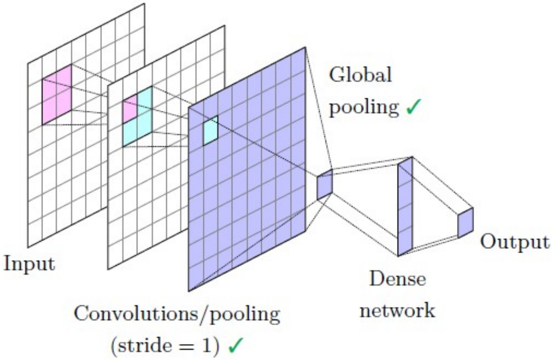
Comparison of architecture types

For fair comparison, best architectures for each type have been obtained by an Optuna optimization (scanning through various kernel sizes, number of layers, number of channels, …)

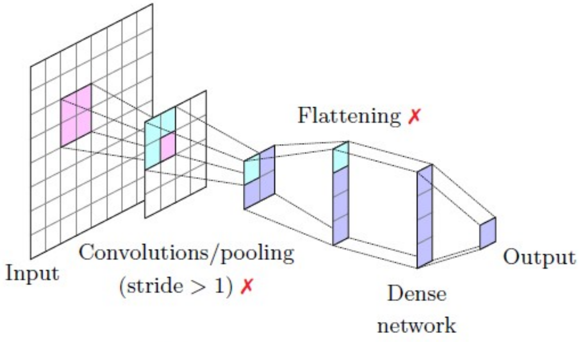
Best architectures are retrained 10 times and evaluated on the validation set.



(b)Strided architecture (ST)

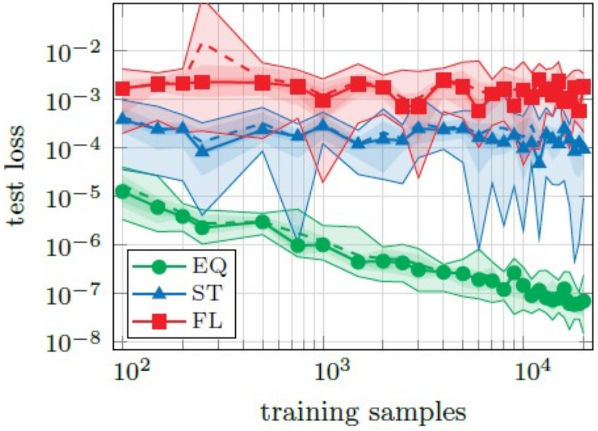


(a)Equivariant architecture (EQ)



(c)Flattening architecture (FL)

Test regression tasks on observables of a scalar field model in 2 dimensions:



Bulusu, Favoni, AI, Müller, Schuh,
Phys. Rev. D 104 (2021) 074504