


Lattice Gauge Equivariant Convolutional Neural Networks

Matteo Favoni¹,[†] Andreas Ipp¹,[‡] David I. Müller¹,^{*} and Daniel Schuh¹,[§]
Institute for Theoretical Physics, TU Wien, A-1040 Wien, Austria

 (Received 23 December 2020; revised 6 August 2021; accepted 1 December 2021; published 20 January 2022)

We propose lattice gauge equivariant convolutional neural networks (L-CNNs) for generic machine learning applications on lattice gauge theoretical problems. At the heart of this network structure is a novel convolutional layer that preserves gauge equivariance while forming arbitrarily shaped Wilson loops in successive bilinear layers. Together with topological information, for example, from Polyakov loops, such a network can, in principle, approximate any gauge covariant function on the lattice. We demonstrate that L-CNNs can learn and generalize gauge invariant quantities that traditional convolutional neural networks are incapable of finding.

DOI: [10.1103/PhysRevLett.128.032003](https://doi.org/10.1103/PhysRevLett.128.032003)

Gauge field theories are an important cornerstone of modern physics and encompass the fundamental forces of nature, including electromagnetism and nuclear forces. The physical information is captured in Wilson loops [1], or holonomies, which describe how a quantity is parallel transported along a given closed path. Local gauge transformations can modify the fundamental fields independently at each space-time point but leave any traced Wilson loop invariant. On the lattice, gauge invariant observables are typically formulated in terms of traced Wilson loops of different shapes. The most basic example is the Wilson action, which is formulated entirely in terms of 1×1 loops, so-called plaquettes. The Wilson action can be systematically improved by including terms involving larger loops [2–6]. Planar rectangular loops are used for characterizing confinement. Most famously, the potential of a static quark pair can be computed from the expectation value of a Wilson loop with large extent in the temporal direction [7]. Improved approximations to the energy momentum tensor or the topological charge density can involve also non-planar loops of growing size [8–10]. As the number of possible loops on a lattice grows exponentially with its path length, a systematic treatment of higher order contributions can become increasingly challenging.

Artificial neural networks provide a way to automatically extract relevant information from large amounts of data. They have become increasingly popular in many Abelian lattice applications, such as for ϕ^4 scalar field, Ising, XY , Potts and Yukawa models, where they can recognize classical [11] and topological [12] phase transitions from

field configurations, determine local and nonlocal features [13,14], or infer action parameters [15]. Neural networks can improve the efficiency of sampling techniques [16], extract optimal renormalization group transformations [17], or reconstruct spectral functions from Green's functions [18]. By the universal approximation theorem, these networks can, in principle, learn any function [19–21]. In order to avoid merely memorizing training samples, imposing additional restrictions on these networks can improve their generalization capabilities [22]. Global translational equivariance induces convolutions [23], which form the basis of convolutional neural networks (CNNs). Additional global symmetry groups, such as global rotations, can be incorporated using group equivariant CNNs (G-CNNs) [24–29]. This approach can be extended to local gauge symmetries. Even though gauge invariant observables can be learned to some extent by nonequivariant networks [30], recently there has been a lot of interest in incorporating gauge symmetries directly into the network structure. For discrete ones, equivariant network structures have been implemented for the icosahedral group [31] or for the \mathbb{Z}_2 gauge group [32]; for continuous ones, a much larger symmetry space is available [33]. A recent seminal work demonstrated that incorporating $U(1)$ or $SU(N_c)$ gauge symmetries into a neural network can render flow-based sampling orders of magnitude faster than traditional approaches [34,35]. This impressive result was obtained using parametrized invertible coupling layers that essentially depend on parallel-transported plaquettes. Up until now, machine learning applications that require larger Wilson loops have relied on manually picking a set of relevant Wilson loops [36] or on simplifications due to the choice of a discrete Abelian gauge group [37]. A comprehensive treatment for continuous non-Abelian gauge groups has been missing so far, and there is an obvious desire to systematically generate all Wilson loops from simple local operations.

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

In this Letter, we introduce lattice gauge equivariant (LGE) CNNs (abbreviated L-CNNs), which we intend as a gauge equivariant replacement for traditional CNNs in machine learning problems for lattice gauge theory. We specify a basic set of network layers that preserve gauge symmetry exactly, while allowing for universal expressivity for physically distinct field configurations. Gauge equivariant layers can be stacked arbitrarily to form gauge equivariant networks. In particular, we provide a new convolutional operation, which, in combination with a gauge equivariant bilinear layer, can grow arbitrarily shaped Wilson loops from local operations. We show that the full set of all contractible Wilson loops can be constructed in this way. Together with topological information from noncontractible loops, in principle, the full gauge connection can be reconstructed [38,39]. Trace layers produce gauge invariant output that can be linked to physical observables. Using simple regression tasks for Wilson loops of different sizes and shapes in pure SU(2) gauge theory, we demonstrate that L-CNNs outperform conventional CNNs by far, especially with growing loop size.

Lattice gauge theory is a discretization of Yang-Mills theory [1,40,41]. We consider a system at finite temperature with gauge group SU(N_c) in $D + 1$ dimensions on a lattice Λ of size $N_t N_s^D$ with N_t (N_s) cells along the imaginary time (spatial) direction(s) with periodic boundary conditions. The link variables $U_{x,\mu}$ specify the parallel transport from a lattice site x to its neighbor $x + \mu \equiv x + a\hat{e}^\mu$ with lattice spacing a . Gauge links transform according to

$$T_\Omega U_{x,\mu} = \Omega_x U_{x,\mu} \Omega_{x+\mu}^\dagger, \quad (1)$$

where the group elements Ω_x are unitary and have unit determinant. The Yang-Mills action can be approximated by the Wilson action [1]

$$S_W[U] = \frac{2}{g^2} \sum_{x \in \Lambda} \sum_{\mu < \nu} \text{Re Tr}[1 - U_{x,\mu\nu}], \quad (2)$$

with the plaquette variables

$$U_{x,\mu\nu} = U_{x,\mu} U_{x+\mu,\nu} U_{x+\nu,\mu}^\dagger U_{x,\nu}^\dagger = \begin{array}{c} \text{---} \text{---} \text{---} \\ \uparrow \quad \downarrow \\ \text{---} \text{---} \text{---} \end{array} \quad (3)$$

which are 1×1 (untraced) Wilson loops on the lattice. Unless specified otherwise, we assume Wilson loops to be untraced, i.e., matrix valued. The plaquette variables transform locally at x as $T_\Omega U_{x,\mu\nu} = \Omega_x U_{x,\mu\nu} \Omega_x^\dagger$.

L-CNNs can express a large class of possible gauge equivariant functions in the lattice gauge theory framework. As customary in feed-forward CNNs, we split L-CNNs into more elementary ‘‘layers,’’ see Fig. 1. As input data for a layer, we use a tuple $(\mathcal{U}, \mathcal{W})$ consisting of nonlocally transforming gauge link variables \mathcal{U} and locally transforming variables \mathcal{W} . The first part of the tuple is the set of variables $\mathcal{U} = \{U_{x,\mu}\}$, which transform according to Eq. (1). For concreteness, we choose the defining (or fundamental) representation of SU(N_c) such that we can treat link variables as complex special unitary $N_c \times N_c$ matrices. Its second part is a set of variables $\mathcal{W} = \{W_{x,i}\}$ with $W_{x,i} \in \mathbb{C}^{N_c \times N_c}$ and index $1 \leq i \leq N_{\text{ch}}$, which we interpret as ‘‘channels.’’ We require these additional input variables to transform locally at x ,

$$T_\Omega W_{x,i} = \Omega_x W_{x,i} \Omega_x^\dagger. \quad (4)$$

A function f that performs some mathematical operation on $(\mathcal{U}, \mathcal{W})$ is called gauge equivariant (or gauge covariant) if $f(T_\Omega \mathcal{U}, T_\Omega \mathcal{W}) = T'_\Omega f(\mathcal{U}, \mathcal{W})$, where $T'_\Omega f$ denotes the gauge transformed expression of the function f . Additionally, a function f is gauge invariant if

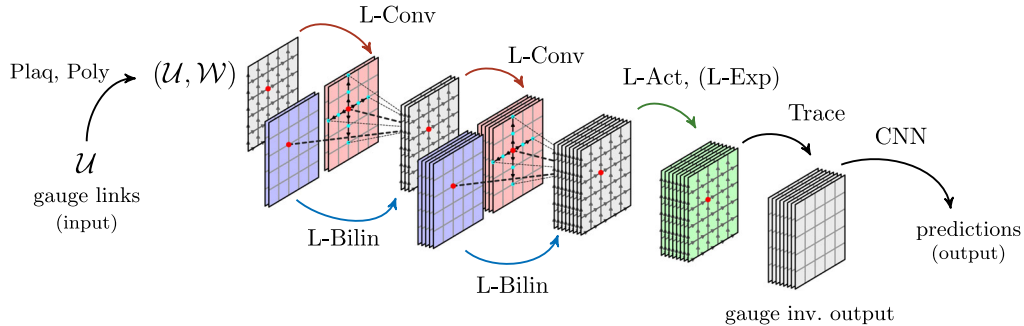


FIG. 1. A possible realization of a L-CNN. Lattice data in the form of \mathcal{U} links are first preprocessed by Plaq. and Poly. in order to generate elementary locally transforming \mathcal{W} objects. A L-Conv is used to parallel transport nearby \mathcal{W} objects (green dots) along the coordinate axes to a particular lattice site (red dot). A L-Bilin combines two layers by forming products of locally transforming objects, which are stored in an increasing number of channels (indicated by stacked lattices). The second input layer (blue) for this operation can be a duplicate of the original layer (red). An additional L-Act (L-Exp) can modify \mathcal{W} (\mathcal{U}) in a gauge equivariant way (green layer). A trace layer generates gauge invariant output that can be further processed by a traditional CNN. The example depicts a 1 + 1D lattice but applies to higher dimensions as well. The basic layers presented can be combined to form other deeper network architectures.

$f(T_\Omega \mathcal{U}, T_\Omega \mathcal{W}) = f(\mathcal{U}, \mathcal{W})$. All possible functions that can be expressed as L-CNNs should either be equivariant or invariant.

LGE convolutions (L-Conv).—Parallel transport of \mathcal{W} objects at neighboring sites to the current location is performed by L-Conv. They can be written as

$$W_{x,i} \rightarrow \sum_{j,\mu,k} \omega_{i,j,\mu,k} U_{x,k\mu} W_{x+k\mu,j} U_{x,k\mu}^\dagger, \quad (5)$$

where $\omega_{i,j,\mu,k} \in \mathbb{C}$ are the weights of the convolution with $1 \leq i \leq N_{\text{ch,out}}$, $1 \leq j \leq N_{\text{ch,in}}$, $0 \leq \mu \leq D$, and $-K \leq k \leq K$, where K is the kernel size. Unlike traditional convolutional layers, the gauge equivariant kernels connect to other lattice sites only along the coordinate axes. The reason is path dependence. In the continuum case, a natural choice would be the shortest path (or geodesic) connecting x and y , which is also used for gauge equivariant neural networks that are formulated on manifolds [31]. However, in our lattice approach, the shortest path is not unique, unless one restricts oneself to the coordinate axes. Possible variations of this layer are to include an additional bias term or to restrict to even sparser dilated convolutions [42].

LGE bilinear layers (L-Bilin).—Two tuples $(\mathcal{U}, \mathcal{W})$ and $(\mathcal{U}, \mathcal{W}')$ are combined by L-Bilin to form products of locally transforming quantities as

$$W_{x,i} \rightarrow \sum_{j,k} \alpha_{i,j,k} W_{x,j} W'_{x,k}, \quad (6)$$

where $\alpha_{i,j,k} \in \mathbb{C}$ are parameters with $1 \leq i \leq N_{\text{out}}$, $1 \leq j \leq N_{\text{in},1}$, and $1 \leq k \leq N_{\text{in},2}$. Since only locally transforming terms are multiplied in Eq. (6), gauge equivariance holds. For more flexibility, the bilinear operation can be further generalized by enlarging \mathcal{W} and \mathcal{W}' to also include the unit element 1 and all Hermitian conjugates of \mathcal{W} and \mathcal{W}' . A L-Bilin can then also act as a residual module [43] and includes a bias term.

LGE activation functions (L-Act).—These functions can be applied at each lattice site via

$$W_{x,i} \rightarrow g_{x,i}(\mathcal{U}, \mathcal{W}) W_{x,i}, \quad (7)$$

using any scalar-valued, gauge invariant function g . A gauge equivariant generalization of the commonly used rectified linear unit (ReLU) could be realized by choosing $g_{x,i}(\mathcal{U}, \mathcal{W}) = \text{ReLU}(\text{ReTr}[W_{x,i}])$ where g only depends on local variables. In general, g can depend on values of variables at any lattice site and, in principle, could also depend on trainable parameters.

LGE exponentiation layers (L-Exp).—L-Exp can be used to update the link variables through

$$U_{x,\mu} \rightarrow U'_{x,\mu} = \mathcal{E}_{x,\mu} U_{x,\mu}, \quad (8)$$

where $\mathcal{E}_{x,\mu} \in \text{SU}(N_c)$ is a group element that transforms locally $T_\Omega \mathcal{E}_{x,\mu} = \Omega_x \mathcal{E}_{x,\mu} \Omega_x^\dagger$. By this update, the unitarity ($U_{x,\mu}^\dagger U_{x,\mu} = \mathbb{1}$) and determinant ($\det U_{x,\mu} = 1$) constraints remain satisfied. A particular realization of $\mathcal{E}_{x,\mu}$ in terms of \mathcal{W} variables is given by the exponential map

$$\mathcal{E}_{x,\mu}(\mathcal{W}) = \exp\left(i \sum_i \beta_{\mu,i} [W_{x,i}]_{\text{aH}}\right), \quad (9)$$

where $[W_{x,i}]_{\text{aH}}$ denotes the anti-Hermitian traceless part of $W_{x,i}$, and $\beta_{\mu,i} \in \mathbb{R}$ are real-valued weight parameters with $0 \leq \mu \leq D$ and $1 \leq i \leq N_{\text{ch}}$. The above method projects $W_{x,i}$ onto the Lie algebra, and therefore $\mathcal{E}_{x,\mu}$ is guaranteed to be an element of the Lie group.

Additional layers.—Trace layers generate gauge invariant output

$$\mathcal{T}_{x,i}(\mathcal{U}, \mathcal{W}) = \text{Tr}[W_{x,i}]. \quad (10)$$

Plaquette layers (Plaq.) generate all possible plaquettes $U_{x,\mu\nu}$ from Eq. (3) at location x and add them to \mathcal{W} as a preprocessing step. To reduce redundancy, we can choose to only compute plaquettes with positive orientation, i.e., $U_{x,\mu\nu}$ with $\mu < \nu$. Polyakov layers (Poly.) compute all possible Polyakov loops [44] at every lattice site according to

$$\mathcal{L}_{x,\mu}(\mathcal{U}) = \prod_k U_{x+k\mu,\mu} = U_{x,\mu} U_{x+\mu,\mu}, \dots, U_{x-\mu,\mu} \quad (11)$$

and add them to the set of locally transforming objects in \mathcal{W} as a preprocessing step. These loops wrap around the periodic boundary of the (toruslike) space-time lattice and cannot be contracted to a single point.

Figure 2 contains a sketch of the proof by induction that L-CNNs can generate arbitrary Wilson loops [Fig. 2(a)]. This is achieved by concatenating loops as shown using a L-Bilin such that intermediate path segments to the origin (indicated by a blue dot in Fig. 2) cancel. Arbitrary paths to a plaquette and back along the same path as shown in Fig. 2(c) can be generated by an initial Plaq. with repeated

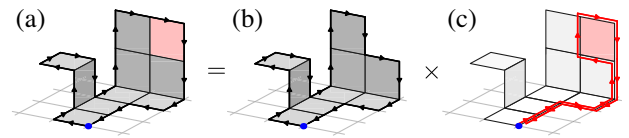


FIG. 2. Sketch of the proof that L-CNNs can generate arbitrary Wilson loops. (a) An arbitrary contractible Wilson loop (depicted here in three dimensions) surrounds a surface that can be tessellated into n tiles of 1×1 unit lattice area. The blue dot indicates the starting point of the untraced Wilson loop. (b) A Wilson loop with n tiles can be composed of an untraced Wilson loop with $n - 1$ tiles and a path along the boundary to the missing tile using a L-Bilin. (c) An arbitrary return path to and from a 1×1 plaquette is obtained by successive applications of L-Convs after an initial Plaq.

application of L-Conv. On topologies that are not simply connected, loops that cannot be contracted to a point can be added by Poly. The possibility of forming a complete Wilson loop basis [38,39] together with the universality of deep convolutional neural networks [21] makes L-CNNs capable of universal approximation within an equivalence class of gauge connections.

These layers can be assembled and applied to specific problems in lattice gauge theory. A possible architecture is depicted in Fig. 1. The alternated application of L-Conv and L-Bilin can double the area of loops. Repeating this block can grow Wilson loops to arbitrary size. L-Bilins are already nonlinear, but even more general relations can be expressed through L-Acts. Building blocks in the form of L-Conv + L-Bilin + L-Act cover a wide range of possible gauge equivariant nonlinear functions. The trace layer renders the output gauge invariant so that it can be further processed by a conventional CNN or a multilayer perceptron without spoiling gauge symmetry. Some applications, such as classical time evolution [45] or gradient flow [46], require operations that can change the set of gauge links \mathcal{U} . This can be achieved using a L-Exp. After a L-Exp, one can use Pla. and Poly. to update \mathcal{W} accordingly.

We demonstrate the performance of L-CNNs by applying them to a number of seemingly simple regression problems. Specifically, we train L-CNN models using supervised learning to predict local, gauge invariant observables and make comparisons to traditional CNN models as a baseline test. We perform our experiments on data from 1 + 1D and 3 + 1D lattices with various sizes and coupling constants g , which we have generated using our own SU(2) Monte Carlo (MC) code based on the Metropolis algorithm [47]. One type of observable that we focus on is the real value of traced Wilson loops, i.e.,

$$W_{x,\mu\nu}^{(m \times n)} = \frac{1}{N_c} \text{ReTr}[U_{x,\mu\nu}^{(m \times n)}], \quad (12)$$

where $U_{x,\mu\nu}^{(m \times n)}$ is an $m \times n$ Wilson loop in the $\mu\nu$ plane. A second observable that we study, which is of more immediate physical relevance, is the topological charge density q_x , which is only available in 3 + 1D. In particular, we focus on the plaquette discretization given by

$$q_x^{\text{plaq}} = \frac{\epsilon_{\mu\nu\rho\sigma}}{32\pi^2} \text{Tr} \left[\frac{U_{x,\mu\nu} - U_{x,\nu\mu}^\dagger}{2i} \frac{U_{x,\rho\sigma} - U_{x,\sigma\rho}^\dagger}{2i} \right]. \quad (13)$$

Our frameworks of choice are PyTorch and PyTorch Lightning. We have implemented the necessary layers discussed previously as modules in PyTorch, which can be used to assemble complete L-CNN models. Our code is open source and hosted on GitLab [48]. In addition to gauge equivariance, we formulate our models to be translationally equivariant, which makes them applicable to arbitrary lattices. The task of the training procedure is to minimize a mean-squared error (MSE) loss function, which compares

the prediction of the model to the ground truth from the dataset. For technical details, see our Supplemental Material [49].

Our L-CNN architectures consist of stacks of L-Conv + L-Bilin blocks, followed by a trace operation, as shown in Fig. 1. The gauge invariant output at each lattice site is mapped by linear layers to the final output nodes. We have experimented with architectures of various sizes, with the smallest models only consisting of a single L-Conv + L-Bilin layer and ≈ 100 parameters to very large architectures with a stack of up to four layers of L-Conv + L-Bilin and $\approx 40\,000$ trainable parameters.

For comparison, we implement gauge symmetry breaking baseline models using a typical CNN architecture. We use stacks of two-dimensional convolutions followed by nonlinear activation functions (such as ReLU, LeakyReLU, tanh, and sigmoid) and global average pooling [50] before mapping to the output nodes using linear layers. Baseline architectures vary from just one or two convolutions with ≈ 300 parameters to large models with up to six convolutions and $\approx 100\,000$ trainable weight parameters. These models are trained and validated on small lattices (8×8 for 1 + 1D and 4×8^3 for 3 + 1D, 10^4 training and 10^3 validation examples) but tested on data from larger lattices (up to 64×64 and 8×16^3 , 10^3 test examples). In total, we have trained 2680 individual baseline models.

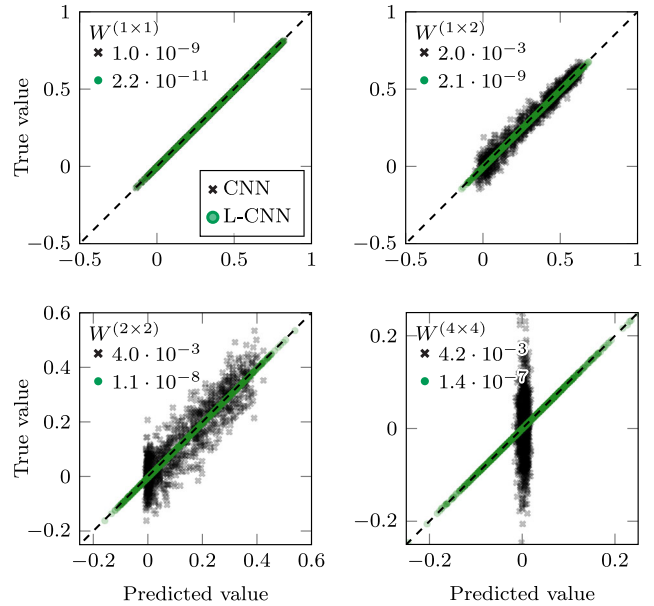


FIG. 3. Scatter plots comparing best L-CNN models to baseline CNN models for Wilson loops of various sizes for 1 + 1D. For each example in the $N_s N_t = 8 \times 8$ test dataset, we plot the true value vs the model prediction. Perfect agreement is indicated by the dashed 45° line. As the size of the traced Wilson loops grows, the performance of the baseline CNN models worsens quickly. On the other hand, L-CNN models achieve high agreement in all cases. The values in the upper left corner denote the MSEs of each plot.

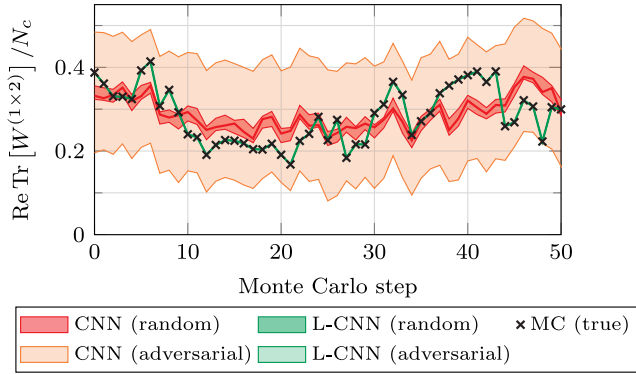


FIG. 4. Prediction uncertainty in $W^{(1 \times 2)}$ due to breaking of gauge symmetry for our best baseline CNN and L-CNN models on 8×8 test data. Black crosses (MC) denote the calculated true value of the Wilson loop. The red bands show the effects of random gauge transformations and transformations obtained from adversarial attacks. Predictions by the L-CNN models are invariant by construction.

A selection of results are presented in Figs. 3 and 4 for $1 + 1$ D and Fig. 5 for $3 + 1$ D lattices. Figure 3 shows scatter plots of our best performing models (L-CNN and baseline) evaluated on test data. We demonstrate that the performance of the baseline models quickly deteriorates with the growing size of the Wilson loop. In the case of 4×4 loops, the baseline model collapses and only predicts the average value of the training data. This signals that the baseline models are unable to learn any meaningful relationship between input and output data. Except for the case of 1×1 loops, the baseline CNN models are not able to adequately learn even moderately sized Wilson loops in $1 + 1$ D and have particular difficulty with predicting negative values, which are associated with large gauge rotations.

We have experimented with different baseline CNN architectures of various widths and depths and a variety of activation functions. In all of our experiments, we have obtained similar behavior as shown in Fig. 3. In contrast, L-CNN architectures are able to converge to solutions that can predict the observables to a high degree of accuracy in all tasks, and they are gauge covariant by construction. Furthermore, our models perform well across all

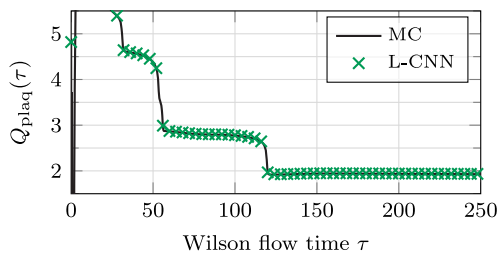


FIG. 5. Predictions of our best L-CNN for the topological charge on a Wilson flowed 8×24^3 lattice configuration at $2/g^2 = 0.2$.

considered lattice sizes because of translational equivariance. In Fig. 4, we show how predictions of baseline CNNs and L-CNNs change under gauge transformations. We have tried two strategies: random gauge transformations and adversarial attacks (see our Supplemental Material, Sec. VI A [49]). We observe that trained CNN models learn to approximate gauge invariance (see [30] for a similar result) but are vulnerable to certain gauge transformations that can drastically change their predictions.

L-CNNs can also be applied to $3 + 1$ D lattices. Figure 5 demonstrates the predictions of our best L-CNN model for topological charge (trained on 4×8^3) during Wilson (gradient) flow [46] on an 8×24^3 lattice. The values from our simulation (MC) agree with the model predictions to high accuracy and assume integer values, as expected.

To summarize, we introduced a neural network structure for processing lattice gauge theory data that is capable of universal approximation within physically relevant degrees of freedom. The network achieves this by growing Wilson loops of arbitrary shapes in successive trainable gauge covariant network layers. We demonstrated that our method surpasses ordinary convolutional networks in simple regression tasks on the lattice and that it manages to predict and generalize results for larger Wilson loops where a baseline network completely fails. Furthermore, our models can also be applied to lattices of any size without requiring retraining or transfer learning. From a broader perspective, we introduced a generalization of traditional CNNs that could replace them in a large range of machine learning applications where CNNs are applied to lattice gauge data.

Our approach opens up exciting possibilities for future research. So far, we implemented the network layers for the $SU(2)$ gauge group only, but our method works for any $SU(N_c)$. Also, we have introduced the general concepts of Polyakov-loop generating layers and of exponentiation layers, but we have not exploited them in numerical experiments. It would be interesting to study these layers and their possible applications. Finally, the compositional nature of successive gauge covariant network layers is reminiscent of the renormalization group picture [51–53]. Trainable networks could provide a viable implementation of the renormalization group approaches by Wilson [54] and Symanzik [55]. Improved lattice actions and operators could be obtained by training on coarse lattices, while providing ground-truth data from finer grained simulations. Automatically learning improved lattice actions could make accessible previously unreachable system sizes for zero and finite temperature applications [2,3,5] as well as for real-time lattice simulations [4,6,56,57].

D. M. thanks Jimmy Aronsson for valuable discussions regarding group equivariant and gauge equivariant neural networks. This work has been supported by the Austrian Science Fund FWF No. P32446-N27, No. P28352 and Doctoral program No. W1252-N27. The Titan V GPU used for this research was donated by the NVIDIA Corporation.

*Corresponding author.

dmueller@hep.itp.tuwien.ac.at

†favoni@hep.itp.tuwien.ac.at

‡ipp@hep.itp.tuwien.ac.at

§schuh@hep.itp.tuwien.ac.at

- [1] K. G. Wilson, *Phys. Rev. D* **10**, 2445 (1974).
- [2] F. Niedermayer, *Nucl. Phys. B, Proc. Suppl.* **53**, 56 (1997).
- [3] Y. Iwasaki, *Nucl. Phys.* **B258**, 141 (1985).
- [4] G. D. Moore, *Nucl. Phys.* **B480**, 689 (1996).
- [5] J. F. Lagaë and D. K. Sinclair, *Phys. Rev. D* **59**, 014511 (1998).
- [6] A. Ipp and D. Müller, *Eur. Phys. J. C* **78**, 884 (2018).
- [7] G. S. Bali, *Phys. Rep.* **343**, 1 (2001).
- [8] S. Caracciolo, G. Curci, P. Menotti, and A. Pelissetto, *Ann. Phys. (N.Y.)* **197**, 119 (1990).
- [9] C. Alexandrou, A. Athenodorou, K. Cichy, A. Dromard, E. Garcia-Ramos, K. Jansen, U. Wenger, and F. Zimmermann, *Eur. Phys. J. C* **80**, 424 (2020).
- [10] S. O. Bilson-Thompson, D. B. Leinweber, and A. G. Williams, *Ann. Phys. (Amsterdam)* **304**, 1 (2003).
- [11] K. Zhou, G. Endrődi, L.-G. Pang, and H. Stöcker, *Phys. Rev. D* **100**, 011501(R) (2019).
- [12] L. Wang, Y. Jiang, L. He, and K. Zhou, [arXiv:2005.04857](https://arxiv.org/abs/2005.04857).
- [13] D. Grimmer, I. Melgarejo-Lermas, and E. Martín-Martínez, [arXiv:1910.03637](https://arxiv.org/abs/1910.03637).
- [14] D. Bachtis, G. Aarts, and B. Lucini, *Phys. Rev. E* **102**, 053306 (2020).
- [15] S. Blücher, L. Kades, J. M. Pawłowski, N. Strodthoff, and J. M. Urban, *Phys. Rev. D* **101**, 094507 (2020).
- [16] J. M. Pawłowski and J. M. Urban, *Mach. Learn. Sci. Tech.* **1**, 045011 (2020).
- [17] H.-Y. Hu, S.-H. Li, L. Wang, and Y.-Z. You, *Phys. Rev. Research* **2**, 023369 (2020).
- [18] L. Kades, J. M. Pawłowski, A. Rothkopf, M. Scherzer, J. M. Urban, S. J. Wetzel, N. Wink, and F. P. G. Ziegler, *Phys. Rev. D* **102**, 096001 (2020).
- [19] G. Cybenko, *Math. Control, Signals Syst.* **2**, 303 (1989).
- [20] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, *Adv. Neural Inf. Process. Syst.* **30** (2017), <https://papers.nips.cc/paper/2017/hash/32cbf687880eb1674a07bf717761dd3a-Abstract.html>.
- [21] D.-X. Zhou, *Appl. Comput. Harmon. Anal.* **48**, 787 (2020).
- [22] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, [arXiv:1710.05468](https://arxiv.org/abs/1710.05468).
- [23] B. Jähne, *Digital Image Processing, 5th Revised and Extended Edition* (Springer-Verlag, Berlin, 2002).
- [24] T. S. Cohen and M. Welling, in *Proceedings of the 33rd International Conference on Machine Learning* (PMLR, 2016), Vol. 48, pp. 2990–2999, <https://arxiv.org/abs/1602.07576>.
- [25] R. Kondor and S. Trivedi, in *Proceedings of the 35th International Conference on Machine Learning* (PMLR, 2018), Vol. 80, pp. 2747–2755, <https://arxiv.org/abs/1802.03690>.
- [26] M. C. Cheng, V. Anagiannis, M. Weiler, P. de Haan, T. S. Cohen, and M. Welling, [arXiv:1906.02481](https://arxiv.org/abs/1906.02481).
- [27] C. Esteves, [arXiv:2004.05154](https://arxiv.org/abs/2004.05154).
- [28] M. Rath and A. P. Condurache, [arXiv:2006.16867](https://arxiv.org/abs/2006.16867).
- [29] J. E. Gerken, J. Aronsson, O. Carlsson, H. Linander, F. Ohlsson, C. Petersson, and D. Persson, [arXiv:2105.13926](https://arxiv.org/abs/2105.13926).
- [30] D. L. Boyda, M. N. Chernodub, N. V. Gerasimeniuk, V. A. Goy, S. D. Liubimov, and A. V. Molochkov, *Phys. Rev. D* **103**, 014509 (2021).
- [31] T. S. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling, in *Proceedings of the 36th International Conference on Machine Learning* (PMLR, 2019), Vol. 97, pp. 1321–1330, <https://arxiv.org/abs/1902.04615>.
- [32] D. Luo, G. Carleo, B. K. Clark, and J. Stokes, *Phys. Rev. Lett.* **127**, 276402 (2021).
- [33] M. Finzi, S. Stanton, P. Izmailov, and A. G. Wilson, [arXiv:2002.12880](https://arxiv.org/abs/2002.12880).
- [34] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, *Phys. Rev. Lett.* **125**, 121601 (2020).
- [35] D. Boyda, G. Kanwar, S. Racanière, D. J. Rezende, M. S. Albergo, K. Cranmer, D. C. Hackett, and P. E. Shanahan, *Phys. Rev. D* **103**, 074504 (2021).
- [36] P. E. Shanahan, D. Trewartha, and W. Detmold, *Phys. Rev. D* **97**, 094506 (2018).
- [37] Y. Zhang, P. Ginsparg, and E.-A. Kim, *Phys. Rev. Research* **2**, 023283 (2020).
- [38] R. Giles, *Phys. Rev. D* **24**, 2160 (1981).
- [39] R. Loll, *Nucl. Phys.* **B400**, 126 (1993).
- [40] C. Gattringer and C. B. Lang, *Quantum Chromodynamics on the Lattice* (Springer, Berlin, Heidelberg, 2010).
- [41] J. Smit, *Introduction to Quantum Fields on a Lattice* (Cambridge University Press, Cambridge, England, 2002).
- [42] F. Yu and V. Koltun, in *4th International Conference on Learning Representations (ICLR 2016) San Juan, Puerto Rico, 2016, Conference Track Proceedings* (2016) [[arXiv:1511.07122](https://arxiv.org/abs/1511.07122)].
- [43] K. He, X. Zhang, S. Ren, and J. Sun, in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New York, 2016), pp. 770–778, [10.1109/CVPR.2016.90](https://arxiv.org/abs/10.1109/CVPR.2016.90).
- [44] A. M. Polyakov, *Phys. Lett.* **72B**, 477 (1978).
- [45] J. Ambjørn, T. Askgaard, H. Porter, and M. Shaposhnikov, *Nucl. Phys.* **B353**, 346 (1991).
- [46] M. Lüscher, *J. High Energy Phys.* **08** (2010) 071; **03** (2014) 092(E).
- [47] M. Creutz, *Phys. Rev. D* **21**, 2308 (1980).
- [48] Our repository is hosted at <https://gitlab.com/openpixi/lge-cnn>.
- [49] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevLett.128.032003> for details regarding our architectures and training procedure.
- [50] M. Lin, Q. Chen, and S. Yan, [arXiv:1312.4400](https://arxiv.org/abs/1312.4400).
- [51] C. Bény, [arXiv:1301.3124](https://arxiv.org/abs/1301.3124).
- [52] P. Mehta and D. J. Schwab, [arXiv:1410.3831](https://arxiv.org/abs/1410.3831).
- [53] S.-H. Li and L. Wang, *Phys. Rev. Lett.* **121**, 260601 (2018).
- [54] K. G. Wilson, in *Recent Developments in Gauge Theories*, edited by G. 't Hooft, C. Itzykson, A. Jaffe, H. Lehmann, P. Mitter, I. Singer, and R. Stora, NATO Advanced Study Institutes Series Vol. 59 (Springer, Boston, 1980), pp. 363–402, [10.1007/978-1-4684-7571-5_20](https://arxiv.org/abs/10.1007/978-1-4684-7571-5_20).
- [55] K. Symanzik, *Nucl. Phys.* **B226**, 187 (1983).
- [56] A. Ipp and D. Müller, *Phys. Lett. B* **771**, 74 (2017).
- [57] A. Ipp and D. I. Müller, *Eur. Phys. J. A* **56**, 243 (2020).