Machine learning action parameters in lattice quantum chromodynamics

Phiala E. Shanahan,[1,2] Amalie Trewartha,[2] and William Detmold[3]

[1]*Department of Physics, College of William and Mary, Williamsburg, Virginia 23187-8795, USA*
[2]*Jefferson Laboratory, 12000 Jefferson Avenue, Newport News, Virginia 23606, USA*
[3]*Center for Theoretical Physics, Massachusetts Institute of Technology,*
Cambridge, Massachusetts 02139, USA

(Received 24 January 2018; published 16 May 2018)

Numerical lattice quantum chromodynamics studies of the strong interaction are important in many aspects of particle and nuclear physics. Such studies require significant computing resources to undertake. A number of proposed methods promise improved efficiency of lattice calculations, and access to regions of parameter space that are currently computationally intractable, via multi-scale action-matching approaches that necessitate parametric regression of generated lattice datasets. The applicability of machine learning to this regression task is investigated, with deep neural networks found to provide an efficient solution even in cases where approaches such as principal component analysis fail. The high information content and complex symmetries inherent in lattice QCD datasets require custom neural network layers to be introduced and present opportunities for further development.

DOI: 10.1103/PhysRevD.97.094506

I. INTRODUCTION

Lattice quantum chromodynamics (LQCD) [1] is a well established numerical method [2,3] used to study quantum chromodynamics (QCD), the theory of the strong interaction. A central part of the Standard Model (SM) of nuclear and particle physics, strong interactions bind quarks and gluons into protons and nuclei, and dictate the emergence of complex nuclear structure in nature. High-precision LQCD calculations are important in determining the parameters of the SM and guide searches for evidence of new physics beyond it [4]. Recent LQCD calculations also provide new insights into the quark and gluon structure of protons [5] and the structure and interactions of light nuclei [6,7]. Similarly, LQCD calculations have enabled investigations of QCD matter at extreme temperatures, and efforts to understand QCD matter at high density are underway [8]. These calculations are extremely computationally demanding, consuming significant fractions of the computational resources that are available for scientific research worldwide.

LQCD calculations are performed on a discrete 4-dimensional space-time grid (typically a hypercubic lattice), and use Monte-Carlo importance sampling [9] to determine the dynamics of the quark and gluon fields defined on this space. Achieving physical results requires a series of calculations at different discretization scales (referred to as the lattice spacing), and different lattice volumes, and a subsequent extrapolation to the continuum (where the discretization vanishes) and infinite volume limits. Particularly challenging is the approach to the continuum limit; the computational cost of the hybrid Monte-Carlo (HMC) algorithm [10] typically used scales with a high inverse power of the lattice spacing, a, approximately a^{-z} with $z > 6$ for a fixed physical lattice volume [11]. Known as critical slowing down, this occurs because of the quasilocal nature of the HMC updating procedure, requiring an increasing number of steps to update physics on a fixed physical volume as the lattice spacing decreases. A number of methods attempt to circumvent this issue by acting at multiple physical length scales. Examples include perfect actions [12–15] that aim to achieve almost-continuum physics at finite lattice spacings, and multiscale thermalization techniques [16–21]. Such approaches require careful renormalization group matching [22,23] of the LQCD actions defined at different scales such that they describe the same long-distance physics. An essential challenge is to solve the parametric regression task: Which action parameters best represent the coarse-scale physics of an ensemble of samples generated at a finer resolution, and vice-versa? Similar parameter regression problems of LQCD data sets arise in the context of mixed action LQCD simulations (see for example Refs. [24–26]).

In this work, machine learning (ML) techniques, in particular neural networks, are applied to the regression problem of determining LQCD action parameters from an ensemble of samples. Significant progress in ML over the last few years has led to new scientific applications of ML tools, including to a number of statistical and quantum mechanics problems. In one set of studies, ML has been used to infer the presence of phase transitions and thermodynamic properties in simple condensed matter models [27–30]. In another study, variational methods have been optimized for many-body problems using ML techniques [31,32]. Novel approaches to the Monte-Carlo method that is ubiquitous in numerical simulations of many systems have also been developed using ML ideas [33–39]. Finally, ML regression for matching Hamiltonians in condensed matter contexts has recently been investigated [34,40] and shows promise. Very few studies, however, have applied ML techniques to investigate gauge field theories such as LQCD (LQCD is a particularly important example of a more general class of theories defined with a local invariance known as a gauge symmetry), and new techniques and adaptations are required because of the unique and complex symmetry structures of these theories.[1] Averaged over Monte-Carlo importance sampling, LQCD data is invariant under discrete spacetime translations and hypercubic group transformations, although individual samples do not have these symmetries. In addition, internal symmetries based on the continuous Lie group SU(3) associated with each spacetime location must be respected. Exploiting these symmetries is essential to the success of the approach used here; it is found that suitably customized deep neutral networks can provide an efficient and practical method of determining the action parameters describing the physics of a given set of configurations.

This article is arranged as follows. In Sec. II, the basic aspects of the lattice QCD calculations that are used to train and test parametric regression by neural networks are discussed, and a principal component analysis (PCA) is used to ascertain the difficulty of the regression tasks that are attempted. In Sec. III, a number of different neural network structures are studied. First, in Sec. III A, a fully connected neural network is used. This easily solves the parameter regression problem on training ensembles, but suffers from overfitting due to the inverted hierarchy of the information content of each sample to the number of samples available for training. Despite its failure to generalize, this network finds features that persist in the LQCD data for Monte-Carlo times considerably longer than those seen for typical physics-motivated observables. The overfitting problem is remedied in Sec. III B, where several custom symmetry-enforcing layers are introduced to define neural network structures that efficiently solve the regression problem. The trained networks correctly resolve parameter differences even between ensembles which are essentially indistinguishable under the PCA analysis. Section IV provides a summary. Two appendices provide additional details of aspects of machine learning and of the lattice QCD calculations.

II. LATTICE QCD

Lattice QCD calculations are performed by approximating the QCD path integral by a Monte Carlo sum over gauge field configurations on a discrete four-dimensional space-time. The expectation value of an operator $\mathcal{O}$ that defines some physical quantity is given by:

$$\langle\mathcal{O}\rangle = \frac{1}{\mathcal{Z}}\int \mathcal{D}\psi\mathcal{D}\bar{\psi}\mathcal{D}A\mathcal{O}[\psi,\bar{\psi},A]e^{-S[\psi,\bar{\psi},A]} \tag{1}$$

$$= \frac{1}{\mathcal{Z}}\int \mathcal{D}U\tilde{\mathcal{O}}[U]e^{-\tilde{S}[U]}, \tag{2}$$

where $\mathcal{Z} = \int \mathcal{D}\psi\mathcal{D}\bar{\psi}\mathcal{D}Ae^{-S[\psi,\bar{\psi},A]}$, the (anti-)fermion and gluon fields (gauge fields) are denoted by $\psi(\bar{\psi})$ and A, and $S[\psi,\bar{\psi},A]$ is the discretized QCD action (defined in Appendix B 1). In the second line, the fermion and antifermion fields are integrated out exactly, and the gauge fields are transformed to link fields $U = e^{iA}$, to give an effective action $\tilde{S}[U]$ and operator $\tilde{\mathcal{O}}[U]$ depending only on the gluon link fields. The resulting integral can be approximated as

$$\langle\mathcal{O}\rangle \approxeq \frac{1}{N_{\rm cfg}}\sum_{i=1}^{N_{\rm cfg}}\mathcal{O}[U_i], \tag{3}$$

where the gauge field configurations U_i (i indexes the configurations in a given "ensemble" of fields) are distributed according to the probability measure $e^{-\tilde{S}[U]}$. In practice, this is guaranteed by sampling the fields from a Markov chain Monte-Carlo stream for which this probability measure is a fixed point. These representative gauge fields are the input data for the ML approaches to parametric regression studied here. For additional details of the LQCD approach, see Refs. [2,3] and Appendix B 1.

Lattice QCD gauge fields are represented as links between sites on a 4-dimensional lattice of volume[2] $V = L^3 \times T$, with the lattice sites separated by some physical distance a, typically 0.05–0.15 fm. Each link, labeled by $U_\mu(x)$, where x denotes the spacetime coordinates of the origin site and μ the direction of the link, is encoded by an SU(3) matrix (a 3×3 complex matrix M with $M^{-1} = M^\dagger$ and $\det[M] = 1$).[3] Links in opposing directions are related via $U_{-\mu}(x) = U_\mu^\dagger(x-\hat{\mu})$, and only

[1] Reference [41] investigates the ability for neural networks to learn a simple order parameter in pure SU(2) gauge theory at finite temperature.

[2] The spatial, L, and temporal, T, extents of the lattice geometry are often distinct.

[3] Here, $M^\dagger = (M^*)^T$ is the Hermitian conjugate. An SU(3) matrix can be specified by 8 real numbers, but typically the redundant representation with 18 real numbers is used.

links in the positive direction are stored. In this format, a gauge field used in typical modern lattice QCD calculations, where for example $L = 64$ and $T = 128$, is described by $L^3 \times T \times 4 \times 18 \approx \mathcal{O}(10^9)$ floating point or double precision numbers, where the factor of 4 arises from the number of positive spacetime directions (labelled by μ). In order to recover QCD results, calculations must be performed on a number of ensembles of field configurations with different lattice spacings a and lattice volumes V, and the continuum ($a \to 0$) and large-volume ($V \to \infty$) limits must be taken.

The governing equations of QCD and their lattice counterparts have a variety of symmetries, some that are highly nontrivial. The symmetries satisfied by ensembles of gauge fields are of particular interest in the context of the ML approaches studied here, as they place strong restrictions on numerical operations that can be performed on lattice data to extract physically meaningful results. In particular, lattice QCD is invariant under a local symmetry of the gauge fields known as a gauge symmetry; this is an invariance under local multiplications of link variables by SU(3) matrices

$$U_\mu(x) \to U'_\mu(x) = \Omega(x) U_\mu(x) \Omega^\dagger(x + \hat{\mu}) \quad \text{for all } \Omega(x) \in \mathrm{SU}(3), \tag{4}$$

referred to as a gauge transformation (note that the matrix $\Omega(x)$ differs at every spacetime point). This symmetry is not apparent from the numerical representation of a QCD configuration, but rather constrains physical observables calculated on a given gauge field to be invariant under all gauge transformations of that field. In addition, lattice QCD defined on a discretized finite volume is invariant under discrete translations and under 4-dimensional rotations and reflections (transformations generated by the hypercubic group, H_4 [42]). Unlike gauge symmetry, these latter symmetries do not hold on a configuration-by-configuration basis, but rather emerge after averaging physical quantities over all gauge fields in an ensemble. An additional important property of QCD is that a characteristic length scale, $1/\Lambda_{\mathrm{QCD}} \sim 1$ fm, emerges dynamically from the interactions of the theory, setting a spacetime distance over which values of the link fields are correlated.

A. Lattice QCD ensembles

A number of different ensembles of lattice QCD gauge field configurations were used for this first exploratory study. Each ensemble was generated using a two-color $N_c = 2$ Wilson gauge action with $N_f = 2$ flavors of dynamical Wilson fermions (defined in Appendix B 1). This action depends on two bare couplings/parameters, β and m_0. QCD with $N_c = 2$ exhibits similar rich dynamical structure to the full theory with $N_c = 3$ and is a natural testing ground for the new approaches developed here. Ensembles were generated with a standard HMC algorithm using a leapfrog integrator to take molecular dynamics trajectory steps of length $\tau_{MD} = 0.5$ in 15–40 substeps (tuned to keep the acceptance rate ~70%). In each case, the streams were initialized from a hot start or from a thermalized lattice from a nearby set of couplings, and the initial 500 trajectories were not included in the further analysis. For most ensembles, configurations were saved every 10 trajectories to generate ensembles of $\mathcal{O}(10^3)$ independent configurations, with the separation determined from studies of the autocorrelation times of typical observables (for some ensembles, configurations were saved every trajectory to allow studies of autocorrelation times to be undertaken). Since $N_c = 2$ in these calculations, rather than $N_c = 3$ in full QCD, the lattice data structures used here are somewhat smaller than those used for state-of-the-art calculations, with each configuration represented by $\mathcal{O}(10^6)$ double precision numbers. All ensembles were generated using a modified version of the CHROMA lattice field theory library [43] that was previously [44] found to produce results consistent with an independent code [45].

Ensembles were generated at many points in parameter space:

(i) Grid A: Twenty $12^3 \times 36$ ensembles of 10,000 trajectories with each $\beta \in \{1.785, 1.835, 1.885, 1.935, 1.985\}$ and $m_0 \in \{-0.7, -0.8, -0.9, -1.0\}$, excluding the pair $\{\beta, m_0\} = \{1.985, -1.0\}$ which could not be thermalized efficiently;

(ii) Grid B: Twenty-five $12^3 \times 36$ ensembles of 10,000 trajectories with each $\beta \in \{1.76, 1.81, 1.86, 1.91, 1.96\}$ and $m_0 \in \{-0.65, -0.75, -0.85, -0.95, -1.05\}$, excluding the pair $\{\beta, m_0\} = \{1.91, -1.05\}$ which could not be thermalized efficiently;

(iii) Grid C: Twenty ensembles with the same bare parameters as Grid A, but with a spacetime volume of $16^3 \times 48$, excluding the pairs $\{\beta, m_0\} = \{1.935, -1.0\}$ and $\{1.985, -1.0\}$, which could not be thermalized efficiently;

(iv) Two sequences of ensembles with parameters tuned to produce closely matched plaquette values. The parameters of each set are indicated by the parentheses (β, m_0):
   (a) Set D: $\{D_1(1.815, -0.98), D_2(1.825, -0.93), D_3(1.838, -0.87), D_4(1.85, -0.83) D_5(1.862, -0.79)\}$;
   (b) Set E: $\{E_1(1.826, -1.03), E_2(1.837, -0.99), E_3(1.847, -0.95), E_4(1.858, -0.9) E_5(1.87, -0.85)\}$;

(v) Set F: Ten independent streams of 10,000 trajectories denoted $F_1, \ldots, F_{10}$, saved every trajectory, generated with the same values of $\beta = 1.76$ and $m_0 = -0.75$.

Simple physical observables, including the pion and rho meson masses and scale setting observables w_0 and t_0 [46], have been calculated on Grids A and B; contour plots displaying the variation of these quantities across the ensembles are shown in Fig. 1.

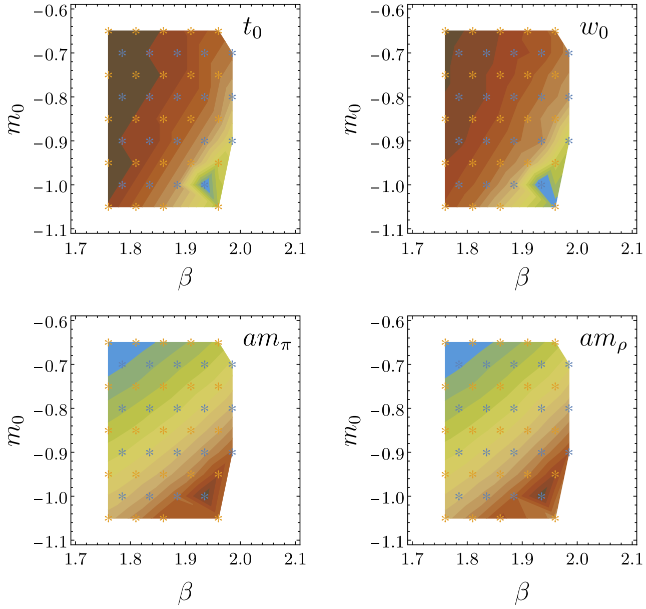


FIG. 1. Contours show the scale setting quantities t_0 and ω_0, as well as the lattice spacing times the pion mass am_π, and rho meson mass am_ρ, determined using calculations on each ensemble in the two $L/a = 12$ grids. The stars show the locations of the ensembles from Grids A (blue) and B (orange).

In order to check the validity of the HMC streams, the evolution of simple quantities along the trajectories has been monitored. The simplest, and computationally cheapest, way to produce a gauge invariant quantity from links is to take the trace of products of links over closed loops ("Wilson loops"). Wilson loops are defined from gauge links as shown schematically in Fig. 2, and detailed in Appendix B 1. Planar Wilson loops $W_{k\times l}(x)$, with indices k and l denoting the dimensions of the loop (with orientation label suppressed), were computed for square loops up to 6×6, as well as rectangular loops of size $1\times n$ for $n = 2,\ldots,12$, and all possible planar orientations. The evolution of representative loop types for the ensembles in Grids A, B, and C, averaged over orientations and spacetime position, is shown in Appendix B 2. For each case, this evolution indicates that the data is well thermalized after approximately 500 trajectories.

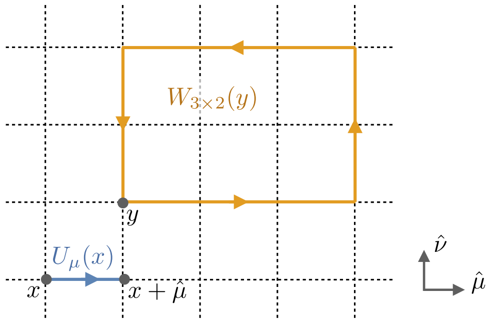


FIG. 2. Diagrammatic representation of the construction of planar Wilson loops $W_{k\times l}(x)$, with indices k and l denoting the dimensions of the loop (with orientation label suppressed), from gauge links $U_\mu(x)$.

To determine the number of HMC steps required for gauge field configurations to be independent, the autocorrelation times of the pion and rho two-point correlation functions, and of the same sets of Wilson loops introduced above, have been calculated. The autocorrelation function for a given operator $\mathcal{O}$ is defined as

$$\rho(\tau) = \sum_{\tau'} \langle (\mathcal{O}(\tau') - \langle\mathcal{O}\rangle)(\mathcal{O}(\tau'+\tau) - \langle\mathcal{O}\rangle)\rangle, \quad (5)$$

where τ is the trajectory difference in the autocorrelation. This function decays exponentially as $\rho(\tau)\sim \exp[-\tau/\tau_{\mathrm{exp}}]$ at large Monte-Carlo times τ. The decay constant τ_{exp} defines an autocorrelation time. Calculations of the autocorrelation time using this definition can suffer from large uncertainties, especially when τ_{exp} is small. Another definition of the autocorrelation time is [3,47]

$$\tau_{\mathrm{int}} = \frac{1}{2} + \lim_{\tau_{\mathrm{max}}\to\infty} \frac{1}{\rho(0)} \sum_{\tau=0}^{\tau_{\mathrm{max}}} \rho(\tau), \quad (6)$$

which approaches a constant as $\tau_{\mathrm{max}}\to\infty$. The autocorrelation functions and integrated autocorrelation times τ_{int} for the Wilson loops, and those for the zero-momentum projected pion and rho two point correlation functions, $C_{\pi(\rho)}$ (defined in Appendix B 1), are shown in Fig. 3. In all cases, the integrated autocorrelation time is $\lessapprox 10$ trajectories, validating the choice to take trajectories spaced by this distance as an uncorrelated set to form an ensemble. Other observables may have different autocorrelation times, but the observables considered here are relatively representative.[4]

B. Ensemble discrimination using principle component analysis

To guide the application of ML methods to parametric regression of gauge fields in the space defined by the sample ensembles, the differentiability of the ensembles was assessed using a principle component analysis (PCA) [48–50]. Since Wilson loops are the simplest gauge-invariant objects, the basis for the PCA was generated by calculating a set of square planar loops of sizes up to $L/2\times L/2$, as well as $1\times n$ for n up to L, averaged over all possible planar orientations and space-time locations. Averaged loops are denoted $W_{j\times l} = \sum_{\mathcal{O}(j\times l)}\sum_x W_{j\times l}(x)$, where the sum over $\mathcal{O}(j\times l)$ is over all hypercubic transformations of the indicated loop. The averaged loop data are sufficiently small

[4] The topological charge of the gauge field typically has a long autocorrelation time, but at the relatively coarse lattice spacings used here, it will be comparable to that of the observables that are investigated.

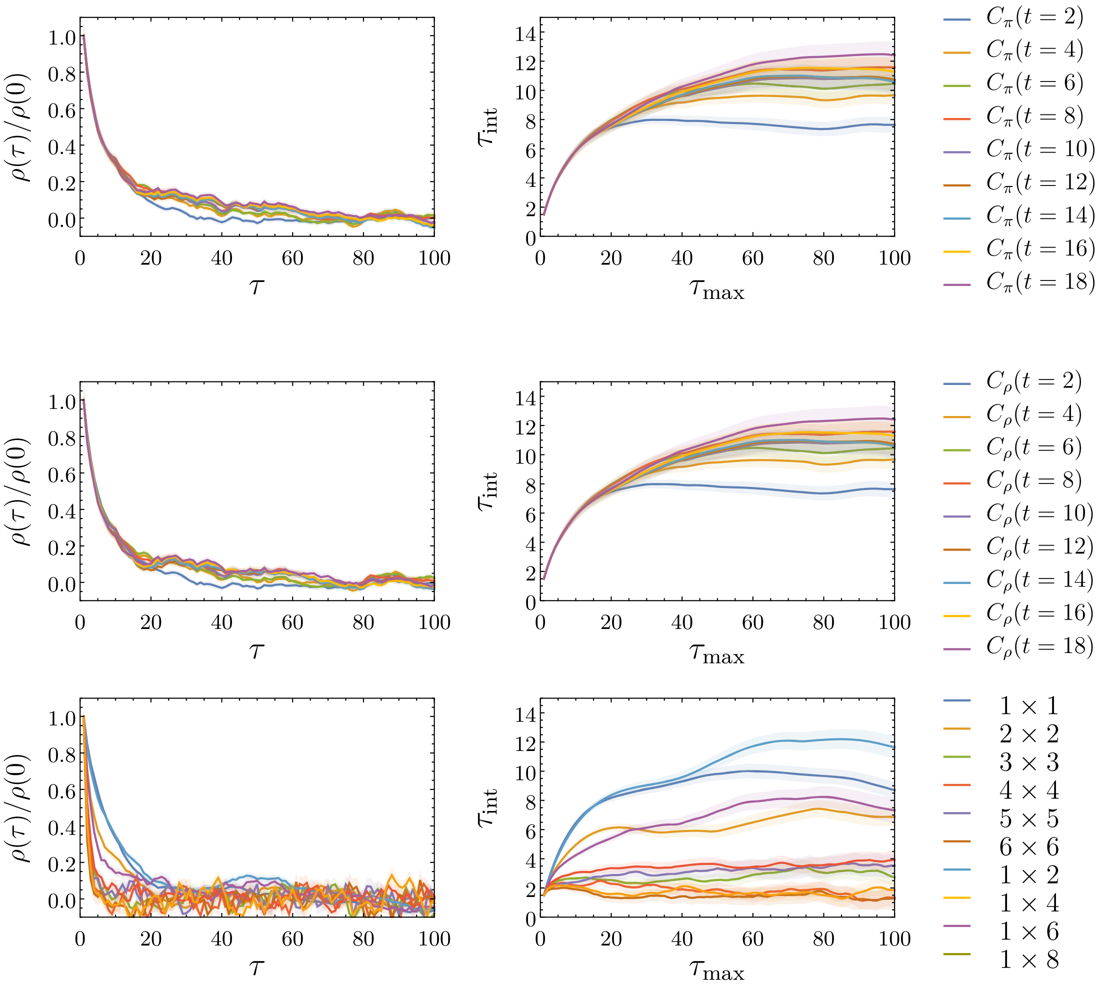


FIG. 3. Autocorrelation functions $\rho(\tau)/\rho(0)$ [left, defined in Eq. (5)] and autocorrelation times τ_{int} [right, defined in Eq. (6)] of the pion (top) and ρ (center) two-point correlation functions at different Euclidean time separations, and of the various space-time averaged $n \times m$ planar Wilson loops (bottom). Measurements are performed on a subset of ensemble F1, for $N_{\text{traj}} = 4000$ sequential trajectories ($N_{\text{traj}} = 7980$ for the loops). The colors identify the type of loop and the shaded bands correspond to the uncertainties on these quantities as determined from a bootstrap procedure using $N_{\text{boot}} = 100$ bootstrap resamplings of size N_{traj}.

in dimension that it is possible to display them for a representative set of ensembles. Figure 4 shows contour plots of $\ln|W_{n\times m}|$ from evaluations on each ensemble in the two $L/a = 12$ grids (Grids A and B). Figures 20, 22, and 24 (in Appendix B 2) show histograms for a subset of the loops for each ensemble in each of Grid A, B, and C, respectively. Clearly, some of the loops are statistically well determined, and subsets of the ensembles can be clearly distinguished. Ensembles in Grid C have loop distributions that are more sharply defined than those in Grids A and B as their larger spacetime volume enables more statistical averaging. For large loop sizes, all ensembles become hard to distinguish.

To perform the PCA on the loop data, a correlation matrix between the various loop observables can be constructed, either for a given ensemble, or, as is done here, across a collection of ensembles. The correlation matrix elements are

$$\mathcal{M}_{\ell_i,\ell_j} = \sum_e \sum_c \frac{[W_{\ell_i}(e,c) - \bar{W}_{\ell_i}(e)][W_{\ell_j}(e,c) - \bar{W}_{\ell_j}(e)]}{\sigma(W_{\ell_i}(e))\sigma(W_{\ell_j}(e))}, \tag{7}$$

where $\ell_i \in \{1\times 1, 2\times 2, \ldots\}$, and e and c label the ensemble and the configuration in that ensemble, respectively. The summation over ensembles is for all ensembles in a given grid, and $\bar{X}$ and $\sigma(X)$ denote the mean and standard deviation of the given quantity over the particular ensemble of configurations. The eigenvalues, e_i, and eigenvectors, v_i, of this correlation matrix for Grid A are shown in Fig. 5. There are three particularly large eigenvalues. Similar pictures emerge from PCAs run on Grid B and Grid C, indicating three dominant degrees of freedom in the calculated Wilson loops. Histograms showing the combinations of loops corresponding to the three dominant, and fourth sub-dominant, eigenvectors are presented for the

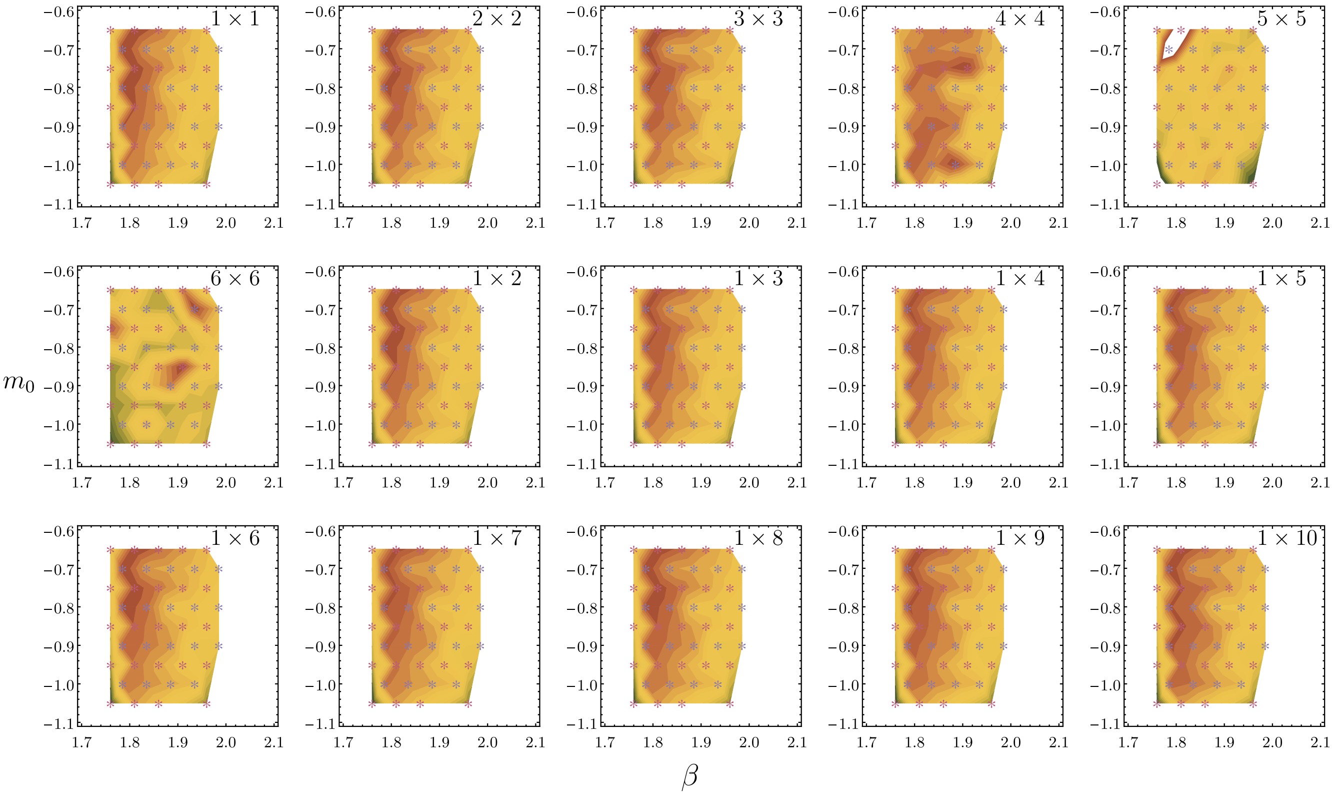


FIG. 4. Contours show ln $|W_{n\times m}|$ from evaluations on each ensemble in the two $L/a = 12$ grids. The stars show the locations of the ensembles from Grids A (blue) and B (orange).

ensembles in Grid A in Fig. 6. Clearly, the information encoded in a collection of the simplest gauge-invariant objects is sufficient to distinguish all but a few of the ensembles in Grid A.

The Jensen-Shannon divergence [51,52] provides a measure of the overlap of probability distributions and can be used to quantify the distinguishability of such distributions. Given two probability distributions P and Q, defined over a space X, the Jensen-Shannon divergence is given by

$$D_{\mathrm{JS}}(P\|Q) = \frac{1}{2}D_{KL}(P\|M) + \frac{1}{2}D_{KL}(Q\|M), \qquad (8)$$

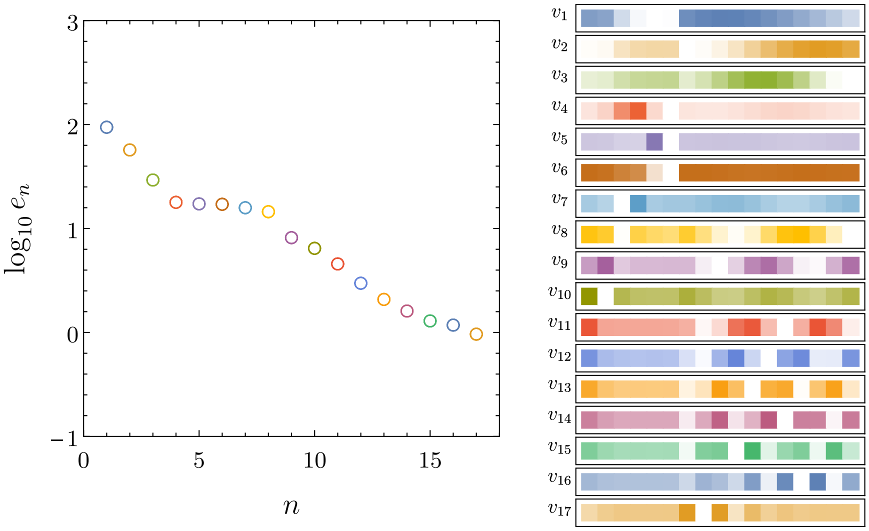


FIG. 5. Eigenvalues e_n (left panel) and eigenvectors v_n (right panel) of the loop correlation matrix for Grid A. The strength of the contribution of each loop to each eigenvector is represented by the tone of the corresponding box in the right panel (i.e., darker = larger contribution).

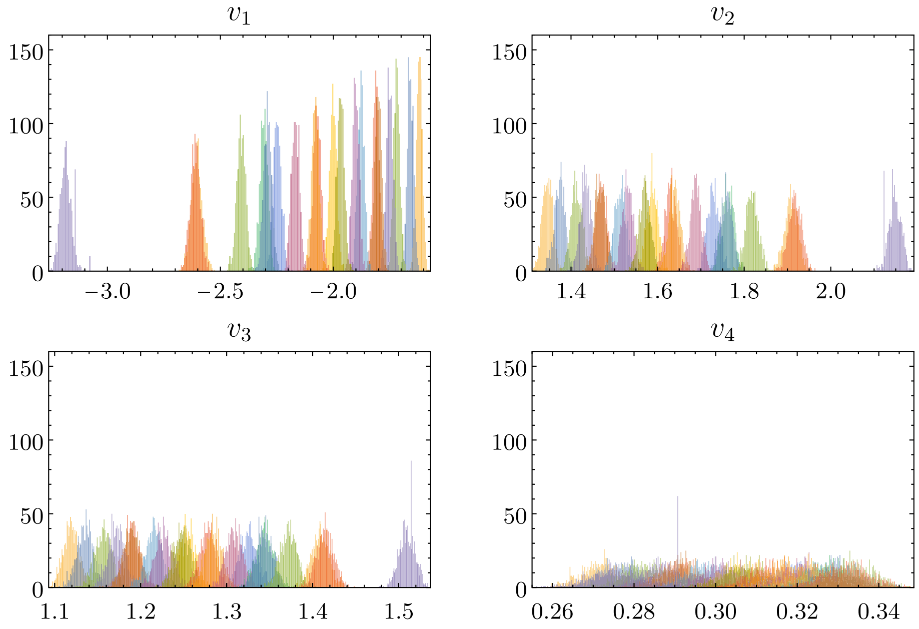


FIG. 6. Combinations of loops corresponding to the four largest eigenvectors of the loop correlation matrix for Grid A. Each color denotes a different ensemble in Grid A.

where $M = \frac{1}{2}(P + Q)$, and $D_{KL}(P\|Q)$ is the Kullback-Leibler divergence [53], defined as

$$D_{KL}(P\|Q) = \int dxP(x)\log_2 \frac{P(x)}{Q(x)}. \tag{9}$$

The Jensen-Shannon divergence is bounded by $0 \leq D_{\mathrm{JS}}(P\|Q) \leq 1$, with $D_{\mathrm{JS}} = 0$ if and only if $Q = P$ almost everywhere, and larger values denoting lower overlap between distributions. The square root of the Jensen-Shannon divergence provides a well-defined metric [54,55].

As a test of differentiability, the Jensen-Shannon divergences were calculated between all pairs of three-dimensional probability distributions defined by the three dominant eigenvectors of the loop correlation matrix for each ensemble in Grid A.[5] To do this, each distribution was first interpolated over the samples from the given ensemble using smooth kernel distributions. The resulting values of D_{JS} are shown pictorially in Fig. 7 for all pairs of the 19 ensembles in Grid A. Clearly, the dominant eigenvectors in loop space allow excellent differentiation between most pairs of ensembles, with approximately 8 out of 171 independent pairs that are only weakly, or not at all, differentiable.

A more challenging test of distribution differentiability is provided by the ensembles in Sets D and E, each designed to have maximal overlap of Wilson loops on each of the ensembles in the set, but different parameters in the $\{\beta, m_0\}$ plane. Figure 8 shows histograms of the combinations of Wilson loops corresponding to the dominant eigenvectors of the loop correlation matrix for ensemble Sets D and E, while Fig. 9 displays the Jensen-Shannon divergence between pairs of ensembles in these sets. As the ensembles in each of Sets D

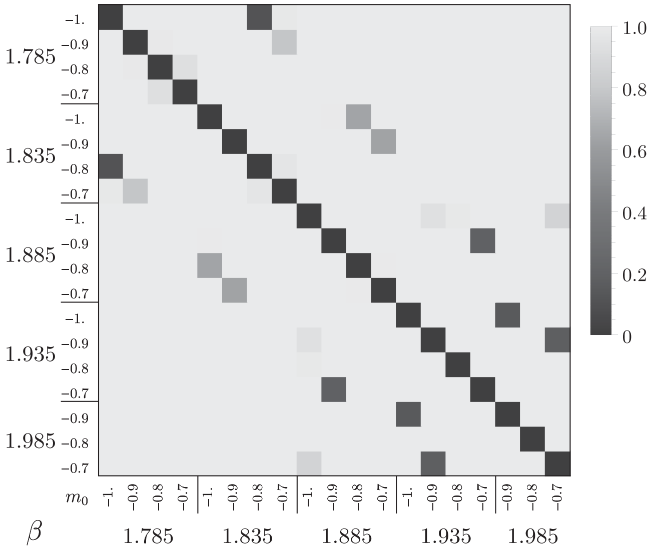


FIG. 7. The Jensen-Shannon divergence, D_{JS}, between pairs of ensembles in Grid A, calculated over the three-dimensional distributions defined by the three dominant eigenvectors of the loop correlation matrix used for the PCA. $D_{\mathrm{JS}} = 1$ implies completely distinguishable distributions.

[5] On a given ensemble e, this three-dimensional probability distribution is given by $P_e(s_i, s_2, s_3)$ where

$$s_i = v_i \cdot (W_{1\times1}(e,c), W_{2\times2}(e,c), \ldots W_{1\times12}(e,c)),$$

and where v_i is the ith eigenvector of the PCA. Additional tests with the largest two or four eigenvectors gave qualitatively similar results.

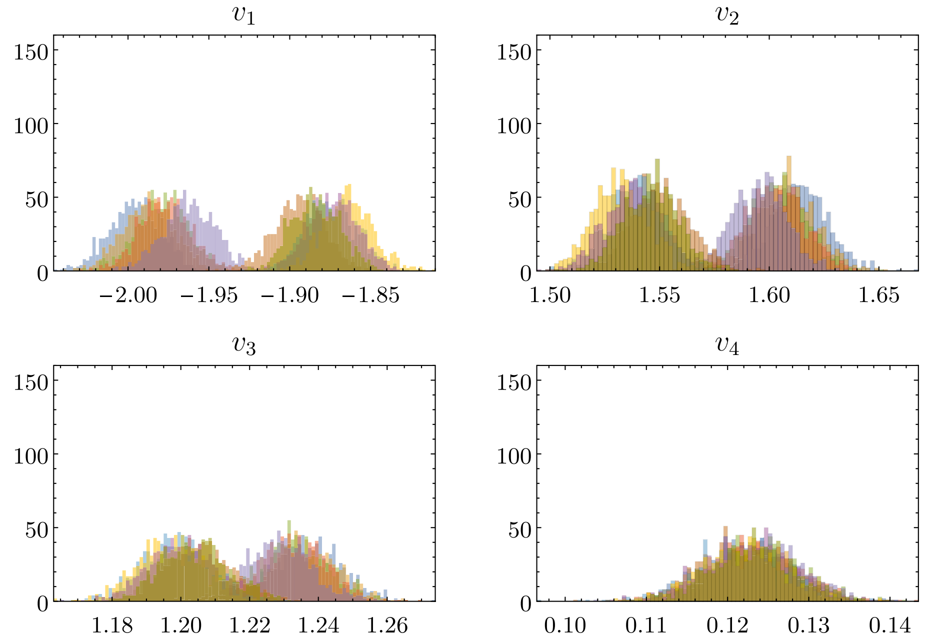


FIG. 8. Combinations of loops corresponding to the dominant eigenvectors of the loop correlation matrix for ensemble Sets D and E. Each color denotes a different ensemble.

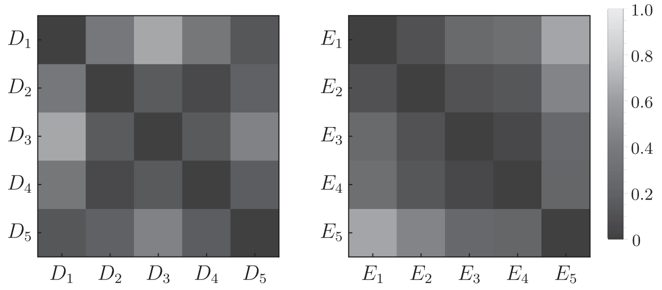


FIG. 9. The Jensen-Shannon divergence, D_{JS}, between pairs of ensembles in Sets D (left) and E (right), calculated over the three-dimensional distributions defined by the three dominant eigenvectors of the loop correlation matrix used for the PCA. The maximum value of D_{JS} in each Set is 0.6. $D_{\mathrm{JS}} = 1$ corresponds to completely distinguishable distributions.

and E are very poorly distinguishable in the space of Wilson loops, accurate differentiation between them presents a key challenge to parametric regression via ML.

III. NEURAL NETWORKS FOR PARAMETRIC REGRESSION OF LATTICE QCD GAUGE FIELDS

Machine learning techniques, and in particular neural networks, offer a promising solution to parameter regression problems. The main focus of this work is to address such a problem in the context of LQCD: given an ensemble of lattice gauge fields, determine the parameters of a given action that are most likely to have generated it. As discussed in the introduction, this challenge arises, for example, in attempts to ameliorate critical slowing down by the matching of coarse and fine lattice actions, and in the context of perfect actions. Its solution will allow for more efficient LQCD calculations, enabling studies in regions of parameter space which are currently computationally unreachable.

To determine the action parameters of a given ensemble (for a particular choice of lattice action), one possible approach is to calculate a sufficiently large set of physics observables both on that ensemble and on a set of ensembles for which the parameters are known, and perform an interpolation and matching task using the calculated observables. The alternative considered here is to train a neural network to perform the regression directly. In principle, this approach is far more general than one based on a set of physics quantities, as the network can use all of the information encoded in a gauge field configuration. On the other hand, this is also challenging. As discussed in Sec. II A, a single gauge field configuration is represented by $\mathcal{O}(10^9)$ real numbers in modern lattice QCD calculations. In comparison, a typical ensemble used for such calculations consists of $\mathcal{O}(10^3)$ configurations. This hierarchy implies that the stochastic learning of features of the relevant degrees of freedom of the gauge field configurations—in particular that extracted physics results must be invariant under spacetime translations, reflections, and hypercubic rotations as well as under gauge transformations—is challenging.

This challenge is approached in two ways, described in the following two sections. First, a multilayer perceptron (a fully connected feed-forward neural network) is trained to learn the action parameters corresponding to lattice gauge field configurations. As anticipated, using gauge fields as input with no symmetry constraints leads to overfitting of the spacetime and gauge features of the data which are not

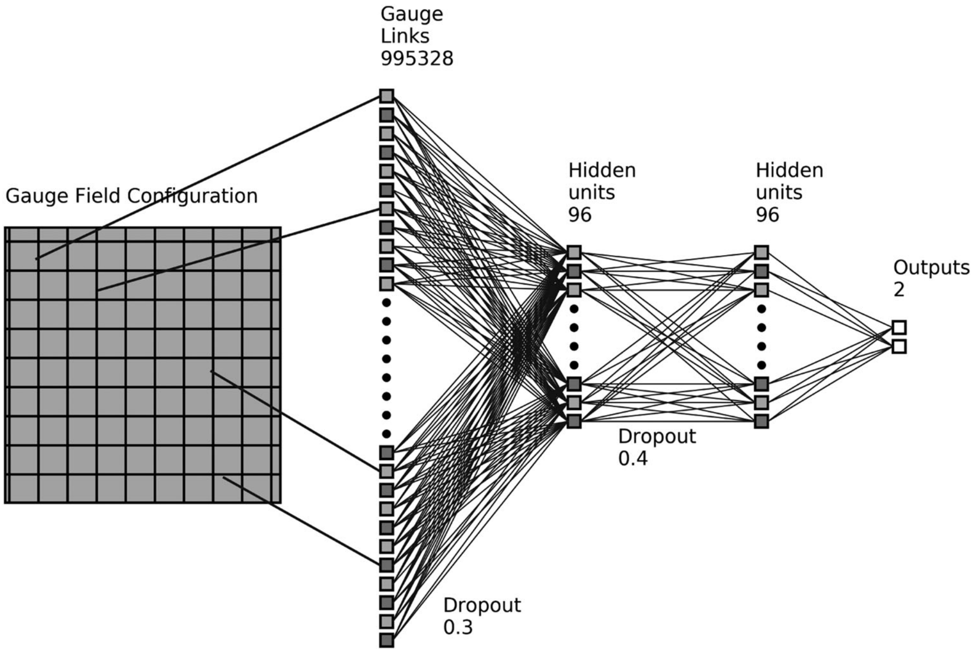


FIG. 10. A schematic representation of the neural network structure used for parametric regression. Gauge links, expressed in an SU (2) basis as 4 real numbers, are used as inputs to the network. There are 4 links in each positive direction from a given site, giving a total of $4 \times 4 \times V = 16 \times 12^3 \times 36 = 995328$ real numbers per gauge field. Two fully connected layers, each with 96 nodes, are used. Each hidden layer features a `tanh` activation function and dropout. A random set of connections between layers are omitted to denote dropout.

related to the physics encoded by a given ensemble. Nevertheless, this exploration reveals a number of interesting features of the problem at hand. Second, a practical solution to the parametric regression problem is provided in the form of a network with a structure that imposes the spacetime and gauge symmetries of LQCD (or, equivalently, involves preprocessing gauge field data into a format that respects these symmetries).

A. Fully-connected network

The simplest approach to the parametric regression of lattice QCD gauge fields using neural networks is to use a multilayer perceptron [56–59], i.e., a fully-connected feedforward network structure (a glossary of neural network terminology is provided in Appendix A). For each of the ensembles of gauge-field configurations in Grid B, 850 configurations were randomly selected as training data, while 100 were held out as validation data [60,61]. Each gauge field configuration, consisting of $\mathcal{O}(10^6)$ real numbers, was treated as an individual input. As physical quantities are only defined on ensemble average, regression on these inputs cannot be exact; a given gauge configuration could, with various probabilities, have been generated from an action differing in both form and parameters from the one that it was in fact generated with, so a perfectly functioning network will necessarily have some spread in predictions from a given ensemble. Quantifying this maximum resolution is possible in principle, but computationally prohibitive, and for this reason has not been undertaken. Investigations into new ensemble-based training approaches that would sharpen the maximum regressor predictability are ongoing.

A simple fully-connected neural network structure, represented graphically in Fig. 10, was trained on the regression task.[6] The network was initialized by setting the biases to zero and the weights to a truncated normal distribution centred at zero with a width of 0.02. A `tanh` activation function was applied to the nodes in each layer, as well as an L2 regularizer with weight decay set to 0.001. Dropout [62–64] was also applied to each layer. While many variations of the network structure were investigated, a systematic hyperparameter tuning was not undertaken due to computational limitations. In general, it was found that fewer hidden units and layers than in the illustrated network led to less optimal minima of the loss function, while a greater number did not appreciably change the outcome. Dropouts in the range 0.3–0.6 were required to eliminate over-fitting. A range of regularization prescriptions and hyperparameters, as well as a range of activations including `tanh`, `reLU` [65–67], and `sigmoid` were studied. The Adam optimizer [68] reached the minimum loss with less training than stochastic gradient descent

[6] The open source packages TENSORFLOW and TFLEARN were used to implement all neural networks and are available from `https://www.tensorflow.org` and `http://tflearn.org`, respectively.

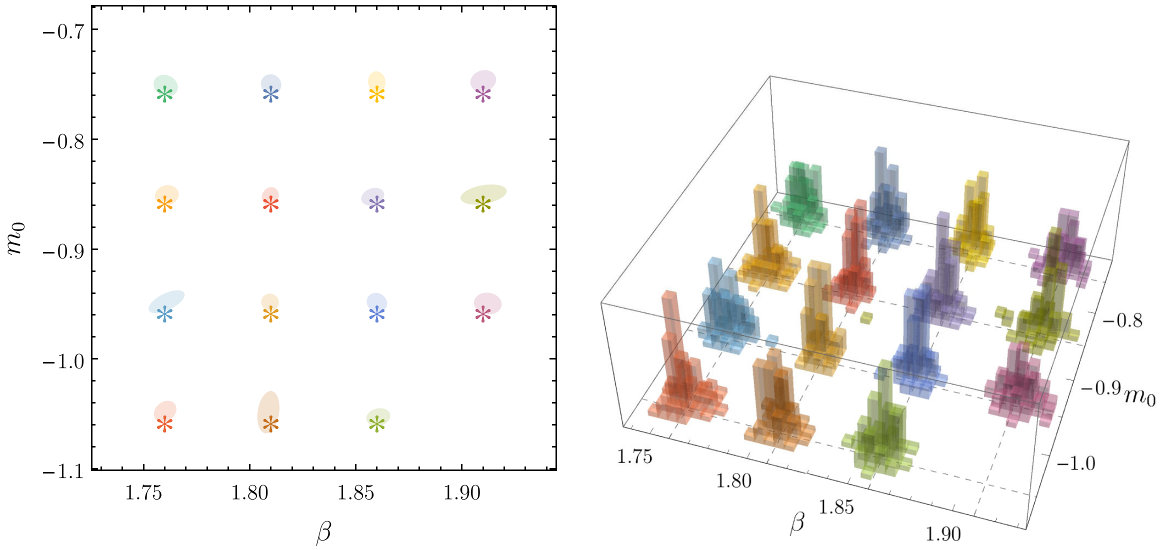


FIG. 11. Predictions of β and m_0 on validation ensembles at the same parameter values as the training ensembles. The stars in the left panel denote the parameters used to generate the ensembles, while the ellipses show the one-standard deviation confidence interval of the predictions for the validation ensembles. The same validation data are shown as histograms in the right figure, with the intersections of the grid lines indicating the parameters used for ensemble generation.

(SGD), and a loss function based on least absolute deviations (L1) rather than least square errors (L2), performed better.

The predictions of the best-performing network for the held-out validation data are shown in Fig. 11. While these results appear to signal the success of this approach, the generalization ability of the network, i.e., its ability to interpolate in parameter space, is poor. In particular:

(i) New ensembles, even those in the 10 ensembles of Set F, generated from separate HMC streams but with the same $\{\beta, m_0\}$ as one of the training ensembles, were predicted to sit at the average β and m_0 values of all ensembles included in training. This indicates that the network did not succeed in learning the gauge-invariance properties of lattice QCD gauge fields, nor in parametrizing the parameter space of the grid of ensembles;

(ii) Configurations from the continuation of the HMC streams used to generate the training and validation configurations were also predicted to have different parameters. Specifically, the next configurations in the HMC streams were predicted to have the correct m_0 and β values, but these predictions drifted towards the average over all training ensembles within a few steps. This indicates that the network is identifying some quantity with a longer autocorrelation time than the physics quantities studied in Sec. II A, i.e., that the configurations separated in MC time such that they are independent by the measure of various physics observables, are not independent by the alternative measure found by the network.

The majority of these features are unsurprising; information content suggests that with $\mathcal{O}(10^3)$ samples containing $\mathcal{O}(10^6)$ real numbers each, it is not feasible to stochastically learn symmetries such as the gauge invariance of the data, and that generalization will be challenging. This could be remedied by using far larger ensembles of gauge field configurations for training, if that were computationally feasible.

The ability of the network to distinguish different streams generated at the same values of β and m_0 is interesting. In the limit of infinite stream lengths, no calculated quantity, corresponding to a physical observable or otherwise, can achieve this distinction. Such distinguishability indicates that the streams are not completely sampling the gauge field configuration space and is tied to the existence of a feature, identified by the network, that has a longer autocorrelation than those of the physics observables studied in Sec. II A. An autocorrelation time of the neural network feature was obtained from the output of classification networks trained on each of the pairs of streams in Set F, generated at the same set of action parameters. Rather than training this network to identify the $\{\beta, m_0\}$ of a given gauge field as for the regression network described previously, the classifier was trained to produce a classification: $\{1, 0\}$ for configurations from one stream, and $\{0, 1\}$ for those from a second. The network structure used was identical to that shown in Fig. 10, with a `softmax` [69] activation function used for the final layer to provide a normalized probability interpretation for the output: an output $\{a, 1-a\}$ for a given configuration indicates that that sample can be identified with the first stream with a probability a. A categorical cross-entropy

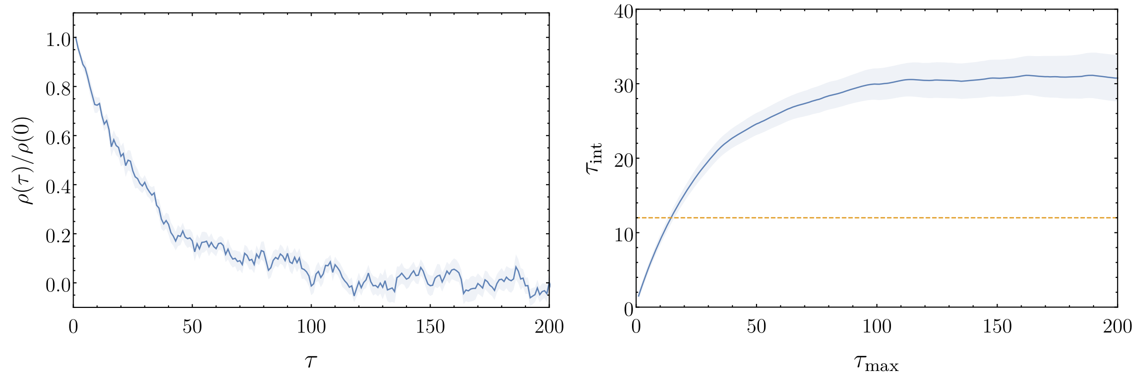


FIG. 12. Autocorrelation function in Monte-Carlo time [left, defined in Eq. (10)] and autocorrelation time [right, defined in Eq. (6)] of the feature distinguishing two streams at the same set of parameters, trained on sequences of gauge field configurations. The autocorrelation function was generated by averaging over many different results (trained using all different pairs of the 10 streams, $F_{1,...,10}$, at the same parameters), and was found to be robust under changes of the network structure used to generate it. The dashed horizontal line on the right figure shows the maximum autocorrelation time of various physics observables (see Fig. 3).

[70,71] loss function was used for this training. For each pair of streams, 600 trajectories from each stream were used to train an instance of the network. The output of that instance for the trajectories sequentially following the training data defines an autocorrelation function:

$$\rho(\tau) = 2[P_\gamma(c^\gamma(\tau))] - 1. \qquad (10)$$

Here, $c^\gamma(\tau)$ labels a trajectory from stream $\gamma = \{\alpha, \beta\}$, τ steps in Monte-Carlo time after the end of the sequence used as training data, and $P_\gamma(c^\gamma)$ denotes the probability, determined from the network output, that trajectory c is in stream γ. The autocorrelation function, and an autocorrelation time determined from this function by Eq. (6), are shown in Fig. 12. Comparing to Fig. 3, it is clear that the autocorrelation time of the feature used by the network to distinguish streams is approximately three times longer than the longest autocorrelation time of the physics observables that were calculated in Sec. II.

It is natural to speculate that the strong autocorrelation observed in the neural network output is based on some local features of the data, rather than features encoding the physics of interest.[7] Further investigation did not find evidence for this interpretation; neither Moran's I [72] nor Geary's C [73] tests supported the existence of correlated spatial regions in the derivatives of the loss function with respect to inputs. There is also no correlation of these derivatives with known spatially-varying physical quantities such as topological charge density and action density. While the long–correlation-time feature could not be identified in this study, it provides an interesting topic for further study. In particular, it will be informative to investigate how this scale changes with parameter range, particularly in regions of parameter space where topological charge freezing becomes a difficult problem for simulations.

B. Custom symmetry enforcing network structure

As described in the previous section, experiments with simple fully-connected neural networks were not successful at parametric regression of lattice QCD gauge fields for the training data sets used in this study. This is not unexpected; learning the symmetries of gauge field configurations stochastically is certain to be a challenging task. Symmetries of lattice QCD, however, act to reduce the effective degrees of freedom of the problem, and can be incorporated into the structure and training of neural networks in several ways. First, the stochastic learning of symmetries can be accelerated through data augmentation (i.e., randomly performing a gauge transformation and/or translation/lattice rotation on a configuration). This is analogous to typical uses of data augmentation [74] in, for example, image recognition [75,76], to introduce symmetries such as rotational symmetry.[8] In practice, this was found to be untenable for the case studied here as a result of the large number of symmetries that must be learned, their complex nature, and the requirement that they be strictly observed. Second, custom network layers can be designed (or equivalently, data can be pre-processed) to only allow gauge invariant and lattice-symmetry invariant outputs of the network. This approach is found to be successful.

To incorporate the symmetries of lattice QCD gauge fields into neural network structures, several custom networks were designed, featuring an initial preprocessing layer that forms only quantities that respect the invariances of the problem, followed by fully-connected layers operating on these quantities. The possible gauge and translation-invariant degrees of freedom that are allowed by the first

[7] This is supported by the observation that features with similar autocorrelation times were identified using network structures that respect gauge-invariance, but retain full spatial information.

[8] The incorporation of symmetries into various neural network structures has been studied in Refs. [77–80].

layer are specified by hand; in principle this choice could be part of the learning process, although naïve implementations are prohibitively expensive. Wilson loops of all shapes and sizes, along with their correlated products, suitably averaged over spacetime, provide a natural choice of gauge-invariant, translation-invariant quantities that can be formed from a gauge field configuration[9] The number of such loops is exponentially large in the spacetime volume and it is computationally intractable to allow all to be generated, so a suitable subset must be chosen. As used in the PCA analysis in Sec. II B, one such subset is the set of square planar loops of sizes up to $L/2 \times L/2$, as well as $1 \times n$ rectangular loops for n up to L, averaged over all possible planar orientations and space-time locations. Another natural choice is the set of all correlated products of two Wilson loops, similarly averaged:

$$\mathcal{W}_{j\times k,l\times m}(R) = \sum_{|r|=R} \sum_{\ell\in\mathcal{O}(j\times k)} \sum_{\ell'\in\mathcal{O}(l\times m)} \sum_{x} \mathcal{W}_{\ell}(x)\mathcal{W}_{\ell'}(x+r), \tag{11}$$

where the sum over $\ell \in \mathcal{O}(j \times k)$ is over all lattice rotations of loops of size $j \times k$, and these loops are chosen from the same list as the single loops described above. Histograms of these correlated loop products for each ensemble in Grids A, B, and C are shown in Figs. 21, 23, and 25 in Appendix B 2. A third choice of simple, gauge-invariant quantities is the set of subtracted correlated products of loops,

$$\mathcal{W}^{(\mathrm{sub})}_{j\times k,l\times m}(R) = \sum_{|r|=R} \sum_{\ell\in\mathcal{O}(j\times k)} \sum_{\ell'\in\mathcal{O}(l\times m)} \left[\sum_{x} \mathcal{W}_{\ell}(x)\mathcal{W}_{\ell'}(x+r) - \sum_{x} \mathcal{W}_{\ell}(x) \sum_{x} \mathcal{W}_{\ell'}(x) \right]. \tag{12}$$

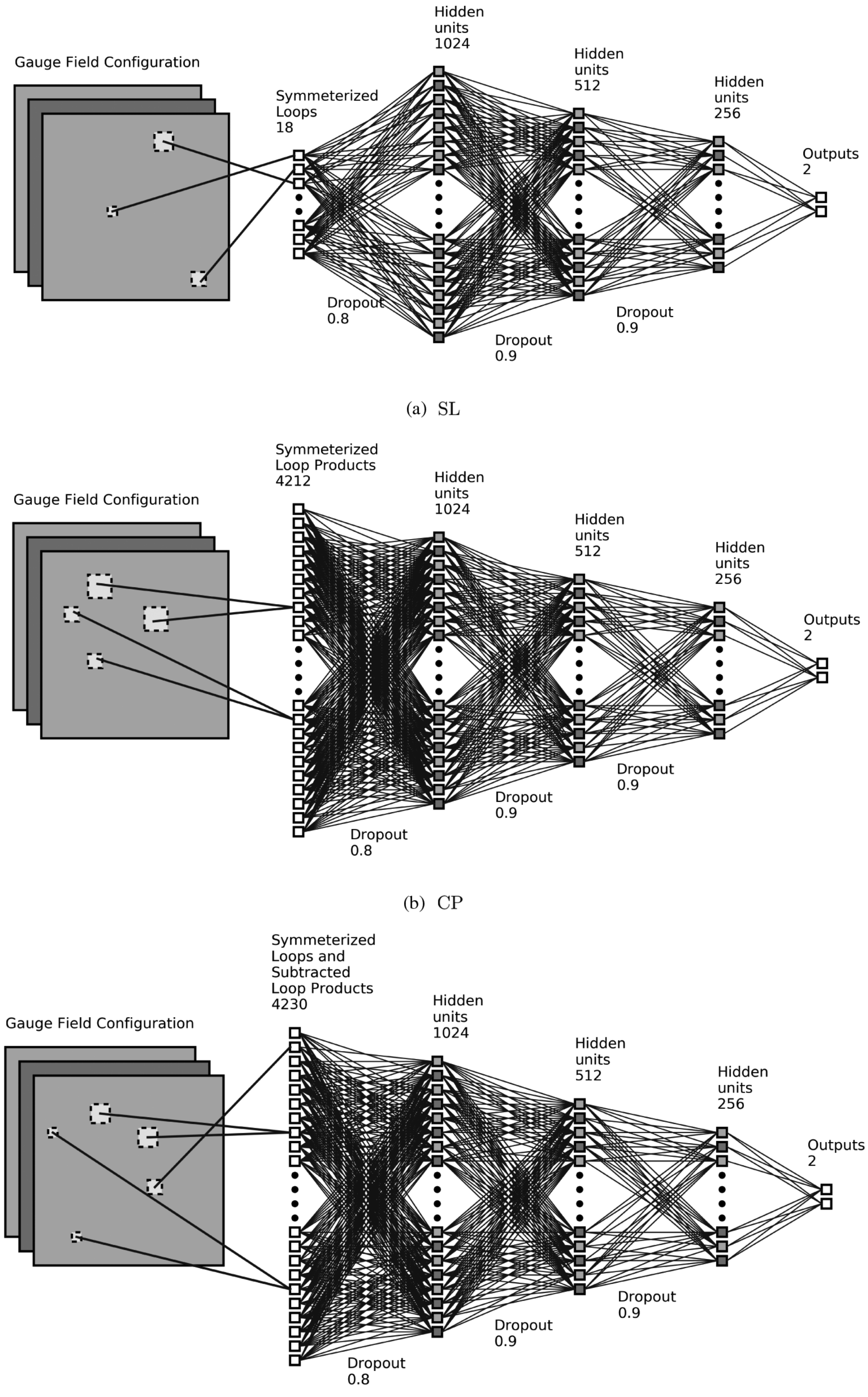
Network structures that allow each of these sets—labeled as single loops (SL), unsubtracted products of two loops (*CP*), and single loops plus the subtracted correlated products of two loops (SLCP)—to be formed in the first layer, are studied.

The complete network structures used for regression are illustrated in Fig. 13 for each of the SL, *CP*, and SLCP cases. Each network was trained using 850 independent configurations from each ensemble in a given grid, with a further 100 held out as validation data. As for the fully-connected network described in the previous section, the networks were initialized by setting the biases to zero and the weights to a truncated normal distribution centred at zero with a width of 0.02. Although no rigorous tuning of the hyperparameters of the networks was undertaken for the various structures, a large number of variations were investigated. In general, networks with fewer hidden units, or fewer layers, than those illustrated in Fig. 13 were found to produce less optimal solutions, while larger networks did not significantly improve on the results that are presented. As for the fully-connected networks, an L1 distance in the two-dimensional parameter space was used as the loss function, and this was found to perform considerably better than the L2 distance. For a given network structure and loss function, the same minimum loss was achieved using different choices of optimizer, including SGD, Adam [68], and Nesterov [81], with various parameters, although the number of epochs required to convergence varied.

The outputs of neural networks allowing each of the SL, SLCP, or *CP* loop sets to be formed in the first layer, trained on the ensembles in Grid A, are shown in Fig. 14. In each case, the results display accurate regression and clear differentiation between the ensembles, with the shapes of the confidence ellipses of network predictions elongated in the direction of constant 1×1 plaquette, the simplest and most precise gauge-invariant object. The mild distortion of the regression results towards the centre of the grid is natural, as this will always lead to a smaller loss in the case of misidentifications than any alternative. With additional tuning and larger or denser parameter grids for training, one might expect that this distortion can be removed. The training and validation losses of each network are shown against training epoch in Fig. 15. The *CP* network performs slightly better than the SL network, as one may anticipate, given that it allows a larger number of degrees of freedom to be utilized. The SLCP network, while also having more degrees of freedom than the SL network, displays overfitting: while the training loss is as good as that of the *CP* network, the validation loss remains higher. It is likely that tuning the network hyperparameters individually for each network structure would improve these results. For the purpose of the present proof-of-principle study, the *CP* network is taken as the best example for further study.

Unlike the fully-connected network described in the previous section, the symmetry-respecting networks generalize successfully, both correctly identifying the parameters of other streams generated with the same action as the training data, which are indistinguishable from the validation distributions, and interpolating to intermediate ensembles. This interpolation is illustrated in Fig. 16, which shows the predictions of the *CP* network on both the evenly-spaced intermediate ensembles of Grid B, and on ensembles in Sets D and E, generated to lie along lines of constant plaquette (isoplaquette lines). While the latter ensembles are essentially indistinguishable along each isoplaquette by various Wilson loops, even using a principal component analysis (see Sec. II B), the parameter predictions from the trained network are distinguishable,

[9] It may also be interesting to explore using the PCA basis of Wilson loops as features for network training.



(a) SL

(b) CP

(c) SLCP

FIG. 13. Diagrams of the neural network structure used. In the first layer, SL, *CP*, or SLCP structures are formed, e.g., in the *CP* case, products of the 18 different types of loops separated by lattice distance $R < 13$ (averaged in integer space bins of R) are allowed, for a total of $18 \times 18 \times 13 = 4212$ loop products. The first layer is followed by 3 fully connected hidden layers with 1024, 512, and 256 nodes. Each hidden layer uses a `tanh` activation function, with dropouts between layers.

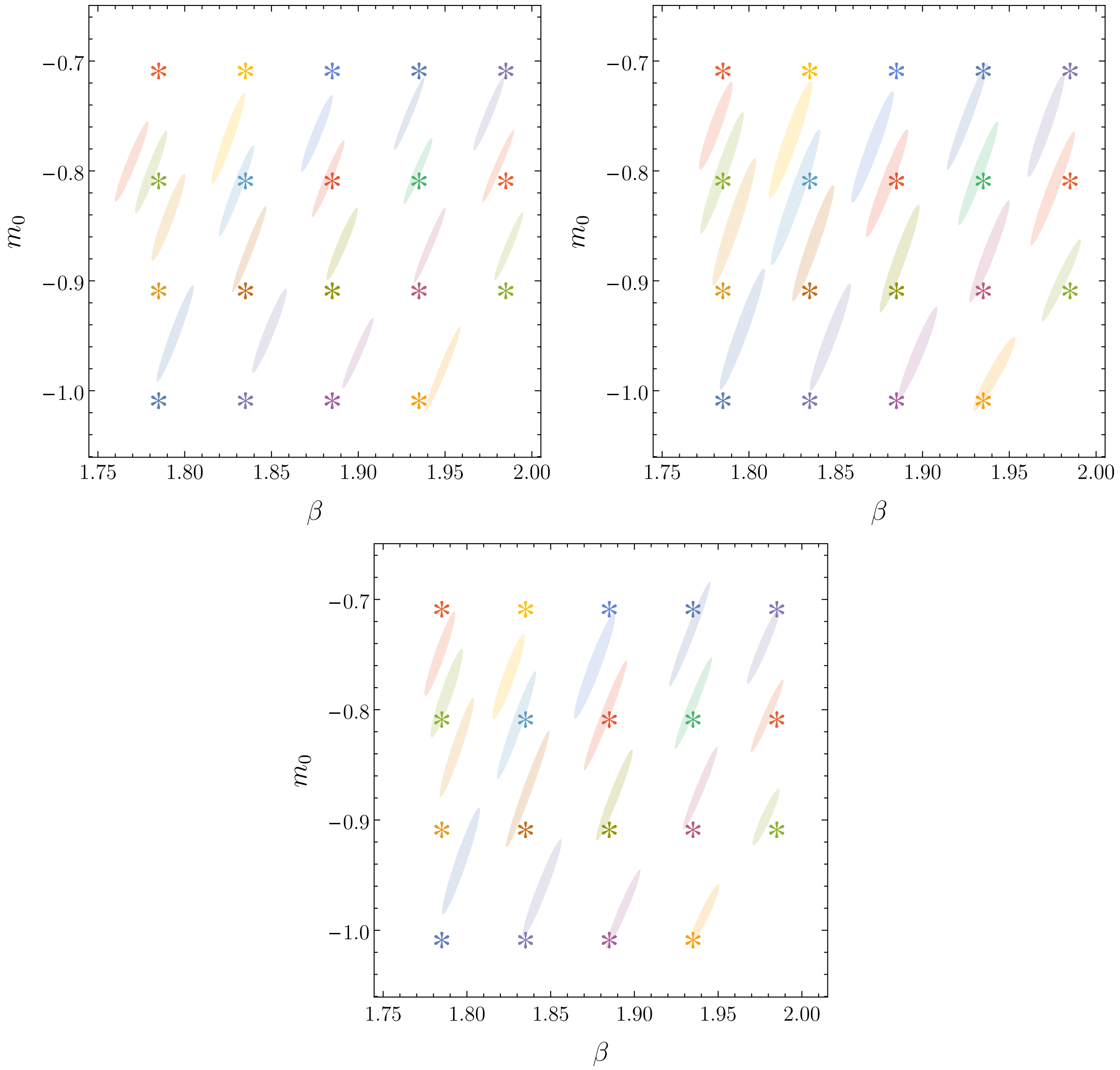


FIG. 14. Predictions of β and m_0 for the validation ensembles in Grid A at the same parameter values of the training ensembles, using SL (left panel), SLCP (right panel) and *CP* (bottom panel) network structures. The stars show the location of each ensemble in parameter space, while the ellipses show the 1σ confidence regions generated from the variation of the predictions for the 100 validation samples from each ensemble.

and, most importantly, have the correct relative positions in parameter space. The overlap between the network predictions for the very closely-spaced ensembles from Set E is anticipated; as described in earlier sections, there is a maximum resolution inherent in this regression problem. Nevertheless, the ordering of the central values of the distributions remains robust. This shows accurate regression of dense points in a region of parameter space significantly smaller than the space between adjacent training ensembles, confirming that the network has successfully parametrized the relevant features of lattice QCD gauge fields.

The accurate regression achieved with the *CP* network relies on having a sufficient density of points in the $\{\beta, m_0\}$ plane in the training data set to enable interpolation. Reducing this density by half, for example, and training the same network structure in the same manner, yields a network instance that generalizes poorly to intermediate ensembles. Figure 17 shows the results of such a test, using the Grid A ensembles. Despite the poor generalization performance, both training and validation loss converge to the same values as for the *CP* network trained on the entirety of Grid A; that is, the training does not indicate over-fitting.

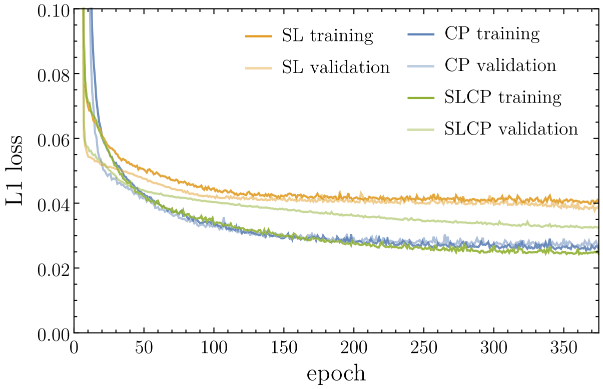


FIG. 15. Loss for networks trained on the ensembles in Grid A with SL (orange), *CP* (blue), and SLCP (green) structures in the first layer, optimized with the Adam optimizer. The dark lines indicate the training loss and the pale lines show loss on the validation data.

The successful parametric regression of lattice QCD gauge fields presented here must be extended to larger-volume lattices more typical of modern lattice QCD calculations for the method to be applied in practice. As lattice volume increases, Wilson loop distributions become more sharply peaked, and as a result become more distinct, as can be seen by comparing Figs. 21 and 25 which display these loops for data sets with spacetime volumes $V = L^3 \times T = 12^3 \times 36$ and $16^3 \times 48$, respectively. It can thus be anticipated that regression performance with the network structures developed here will improve on larger lattice volumes. Figure 18 shows the results of a *CP* network structure trained on Grid C. As expected, the regression performance is better than for the smaller-volume ensembles. Extending these results to even larger volumes, and to $N_c = 3$ QCD, is essential.

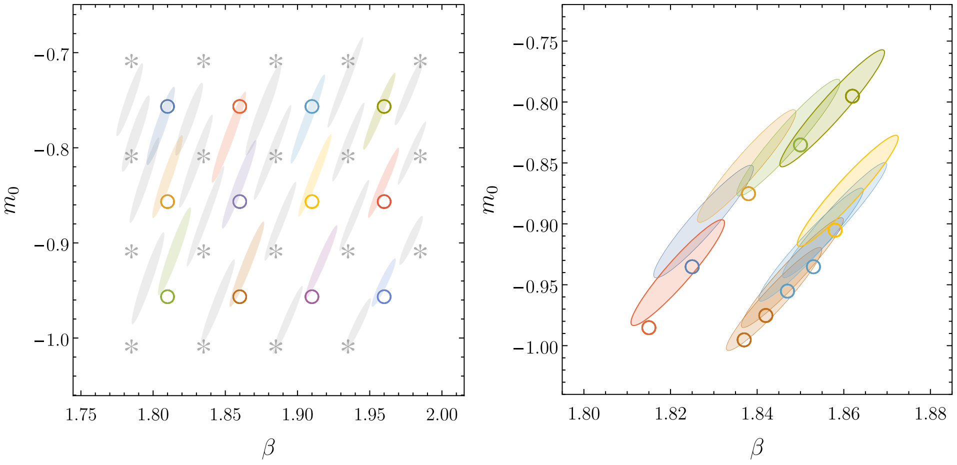


FIG. 16. Predictions of β and m_0 from the *CP* network trained on Grid A, for the ensembles in Grid B (left panel) and Sets D and E (right panel). The open circles show the location of each ensemble in parameter space, while the ellipses show the 1σ confidence regions generated from the variation of the predictions for the 100 validation samples from each ensemble. The greyed-out stars and ellipses show the validation data and training ensemble locations.

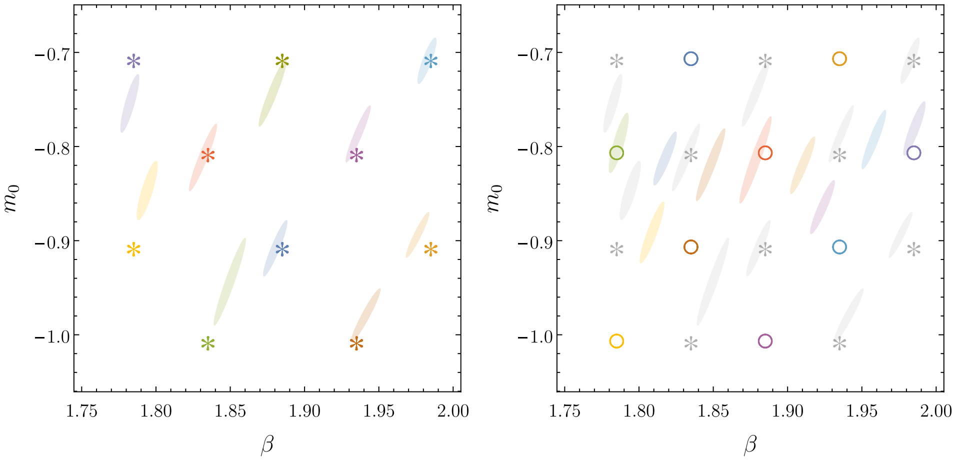


FIG. 17. Predictions of β and m_0 from a *CP* network trained on a subset of the ensembles in Grid A. The stars show the location of each ensemble in parameter space, while the ellipses show the 1σ confidence regions generated from the variation of the predictions for the 100 validation samples from each ensemble. In the right panel, the open circles show the location of testing ensembles, that were not included in training, in the parameter space, while the matched-color ellipses show the 1σ confidence regions of the network predictions.

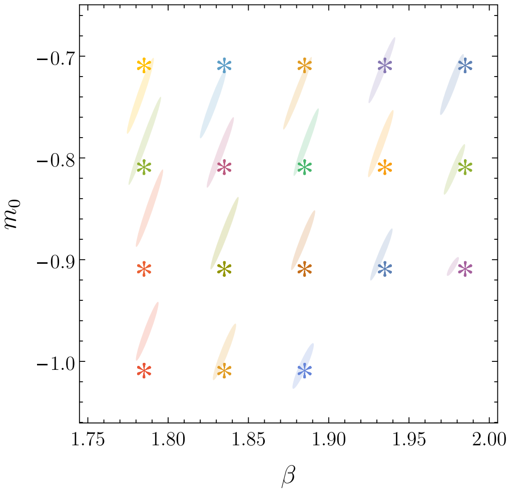


FIG. 18. Predictions of β and m_0 for the validation ensembles in Grid C at the same parameter values of the training ensembles, using a *CP* network structure. The stars show the location of each ensemble in parameter space, while the ellipses show the 1σ confidence regions generated from the variation of the predictions for the 100 validation samples from each ensemble.

IV. SUMMARY

Deep neural networks with custom symmetry-preserving layers provide a solution to the parameter regression problem in lattice QCD, for the proof-of-principle case considered here. Specifically, neural networks regressors trained on grids of ensembles in action parameter space were able to accurately identify the parameters used to generate streams of ensembles, generalizing successfully and accurately to ensembles densely spaced and between grid points in the training space. Non-symmetry preserving networks were also studied. While these were unsuccessful at the regression task, they revealed an unknown feature of the lattice ensembles with a longer correlation length than any of the physics observables that were studied.

Extending this work to SU(3) gauge groups and to larger lattice volumes will be essential for the practical application of the methods developed. In addition to the symmetries exploited here, a typical length scale, $1/\Lambda_{\mathrm{QCD}} \sim 10^{-15}$ m, emerges dynamically in LQCD calculations. Consequently, there are potential advantages for a convolutional approach [82–84] at larger lattice volumes. Convolutional layers would again have to be customized, respecting the gauge symmetry of the problem. Particular use-cases of LQCD parameter regression may also impose additional constraints. For example, regression for the matching of coarse and fine lattice actions requires the identification of ensembles generated in a coarse space with ensembles describing the same physics, but generated via a coarsening prescription [20,21]. The latter ensembles, by renormalization-group evolution, are described by lattice actions with more parameters than those generated in the coarse space. Preliminary investigation suggests that regression under these conditions will require network structures invariant under irrelevant short-distance degrees of freedom, or the marginalization over such degrees of freedom in the learning procedure. Regression of the larger number of parameters in such actions (and used in the construction of perfect actions [12–15]), must also be investigated further.

Clearly, having demonstrated the feasibility of neural network approaches to LQCD in the present work, significant further study is warranted. In particular, the use of lattice symmetries to overcome the dramatic inverted data hierarchy of LQCD—the feature that there are typically far fewer samples than degrees of freedom per sample available—opens the door to many novel applications of machine learning in LQCD.

ACKNOWLEDGMENTS

We are grateful to Kyle Cranmer, Michael Endres, Brendan Fong, Andrew Pochinsky, and Mike Williams for numerous discussions. The calculations in this project were performed using the Hyak High Performance Computing and Data Ecosystem at the University of Washington, supported, in part, by the U.S. National Science Foundation Major Research Instrumentation Award, Grant Number 0922770, and on clusters at MIT with support from the NEC Corporation Fund. W. D. was partly supported by U.S. Department of Energy Early Career Research Award No. DE-SC0010495 and Grant No. DE-SC0011090 and by the SciDAC4 Award No. DE-SC0018121: Computing the Properties of Matter with Leadership Computing Resources. P. E. S. and A. T. were partially supported through contract number DE-AC0506OR23177 under which JSA operates the Thomas Jefferson National Accelerator Facility. A. T. was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

APPENDIX A: NEURAL NETWORK GLOSSARY

Multi-layer perceptron: A multilayer perceptron is the simplest form of a multilayer neural network, having a feed-forward network structure, i.e., triggering the activation of each layer of the network successively, without circulating, and consisting of multiple fully-connected layers that use nonlinear activation functions.

Loss function: A loss, or objective, function, is a measure of the difference between the output of a neural network for a given training sample, and the ground truth. This function defines success for network training. Training procedures, such as stochastic gradient descent, or adaptive learning rate algorithms such as Adam or

Nesterov, update the weights and biases of neural networks to minimize the loss.

Training and validation data sets: It is typical to hold out some data from training a neural network to form a validation data set to provide a generalization test for the network. A larger loss calculated on the validation data than on that used for training is an indication of over-fitting.

Over-fitting: The production of a model that is fit to irrelevant features or fluctuations of the training data and therefore fails to generalize reliably.

Dropout: Dropout is a regularization procedure in neural networks whose purpose is to prevent over-fitting. Dropout prevents neutrons from coadapting by randomly setting a fraction, governed by the dropout hyperparameter, to zero at each training iteration. This results in a model that can be interpreted as randomly sampling from an exponential number of similar networks [64], and creates more generalizable representations of data.

Activation: A neural network layer typically consists of a linear transformation followed by a non-linear transformation at each node, known as the activation function. This non-linearity is what allows neural networks to learn complex decision boundaries. Typical choices of activation functions include `sigmoid`, `tanh`, and `reLU` (defined as x for $x > 0$, 0 otherwise).

Epoch vs. iteration: In the training of a neural network, an iteration is one update of the neural net model parameters. Typically, networks are batch-trained, with a hyperparameter governing the batch size of training data considered per update. An epoch is a complete pass through a given training data set, which may take one (if the batch size is equal to the size of the dataset) or more iterations.

APPENDIX B: LATTICE QCD DETAILS

1. Details of lattice actions, correlation functions, and Wilson loops

The discretized lattice QCD action is expressed in terms of the gauge links between lattice sites, $U_\mu(x)$ (which are $\mathrm{SU}(N_c)$ matrices for a theory with N_c colors), and the fermion fields $\psi(x)$, with the Euclidean space-time positions $x \in \Lambda = \{a(n_1, n_2, n_3, n_4) | n_i \in \mathbb{Z}\}$. The simplest action with the appropriate symmetries for a theory with N_f flavors is given by:

$$S(\beta, m_0) = \frac{\beta}{N_c} \sum_{x\in\Lambda} \sum_{\mu<\nu} \mathrm{Re}\ \mathrm{Tr}[1 - P_{\mu\nu}(x)] + \sum_{f=1}^{N_f} a^4 \sum_{x,y\in\Lambda} \bar{\psi}_f(x) D(m_0) \psi_f(y), \tag{B1}$$

where $P_{\mu\nu}$ is the plaquette: the shortest, nontrivial, closed loop on the lattice, defined in terms of gauge links as

$$P_{\mu\nu}(x) = U_\mu(x) U_\nu(x+\hat{\mu}) U_{-\mu}(x+\hat{\mu}+\hat{\nu}) U_{-\nu}(x+\hat{\nu}), \tag{B2}$$

where $\hat{\mu}$ denotes the vector of length a in the μ direction. The Wilson Dirac operator is

$$D(m_0) = \left(\frac{4}{a} + m_0\right)\mathbb{I} - \frac{1}{a} \sum_{\mu=0}^{3} (P_\mu^- \Omega_\mu^+ + P_\mu^+ \Omega_\mu^-), \tag{B3}$$

with

$$P_\mu^\pm = \frac{1}{2}(1 \pm \gamma_\mu), \quad \langle x|\Omega_\mu^+|y\rangle = \delta_{x+\mu,y} U(x,\mu), \quad \Omega_\mu^- = (\Omega_\mu^+)^\dagger. \tag{B4}$$

The action is parametrized by two values: the coupling constant β and the bare quark mass m_0.

The plaquette can be generalized to Wilson loops of arbitrary shapes and dimensions. Planar Wilson loops $W_{k\times l}(x)$, with indices k and l denoting the dimensions of the loop (with orientation label suppressed), as illustrated in Fig. 2, are expressed in terms of gauge links as

$$\begin{aligned} W_{k\times l}(x) = {} & U_\mu(x) U_\mu(x+\hat{\mu}) \dots U_\mu(x+(k-1)\hat{\mu}) \\ & \times U_\nu(x+k\hat{\mu}) U_\nu(x+k\hat{\mu}+\hat{\nu}) \dots \\ & \times U_\nu(x+k\hat{\mu}+(l-1)\hat{\nu}) U_{-\mu}(x+k\hat{\mu}+l\hat{\nu}) \\ & \times U_{-\mu}(x+(k-1)\hat{\mu}+l\hat{\nu}) \dots U_{-\mu}(x+\hat{\mu}+l\hat{\nu}) \\ & \times U_{-\nu}(x+l\hat{\nu}) U_{-\nu}(x+(l-1)\hat{\nu}) \dots U_{-\nu}(x+\hat{\nu}) \end{aligned} \tag{B5}$$

Two-point correlation functions are defined as the matrix elements corresponding to the creation of some state at a time 0, and annihilation at some later time t. For the pion and rho mesons considered in this work, with suitable choices of creation and annihilation operators, the zero-momentum projected correlation functions can be defined as

$$C_{\pi(\rho)}(t) = \sum_{\mathbf{x}} \langle 0|\bar{u}\gamma_{5(3)} d(\mathbf{x}, t) \bar{d}\gamma_{5(3)} u(\mathbf{0}, 0)|0\rangle, \tag{B6}$$

where u and d denote quark creation (and $\bar{u}$ and $\bar{d}$ annihilation) operators. For further details, see Refs. [2,3].

2. Further details of ensemble properties

In this appendix, the properties of the various LQCD data sets used in this work are presented. Figure 19 shows the evolution of various Wilson loops with HMC trajectory for the ensembles in Grids A, B, and C, while Figs. 20–25 present histograms of the Wilson loops and correlated products of Wilson loops on each ensemble in these grids.

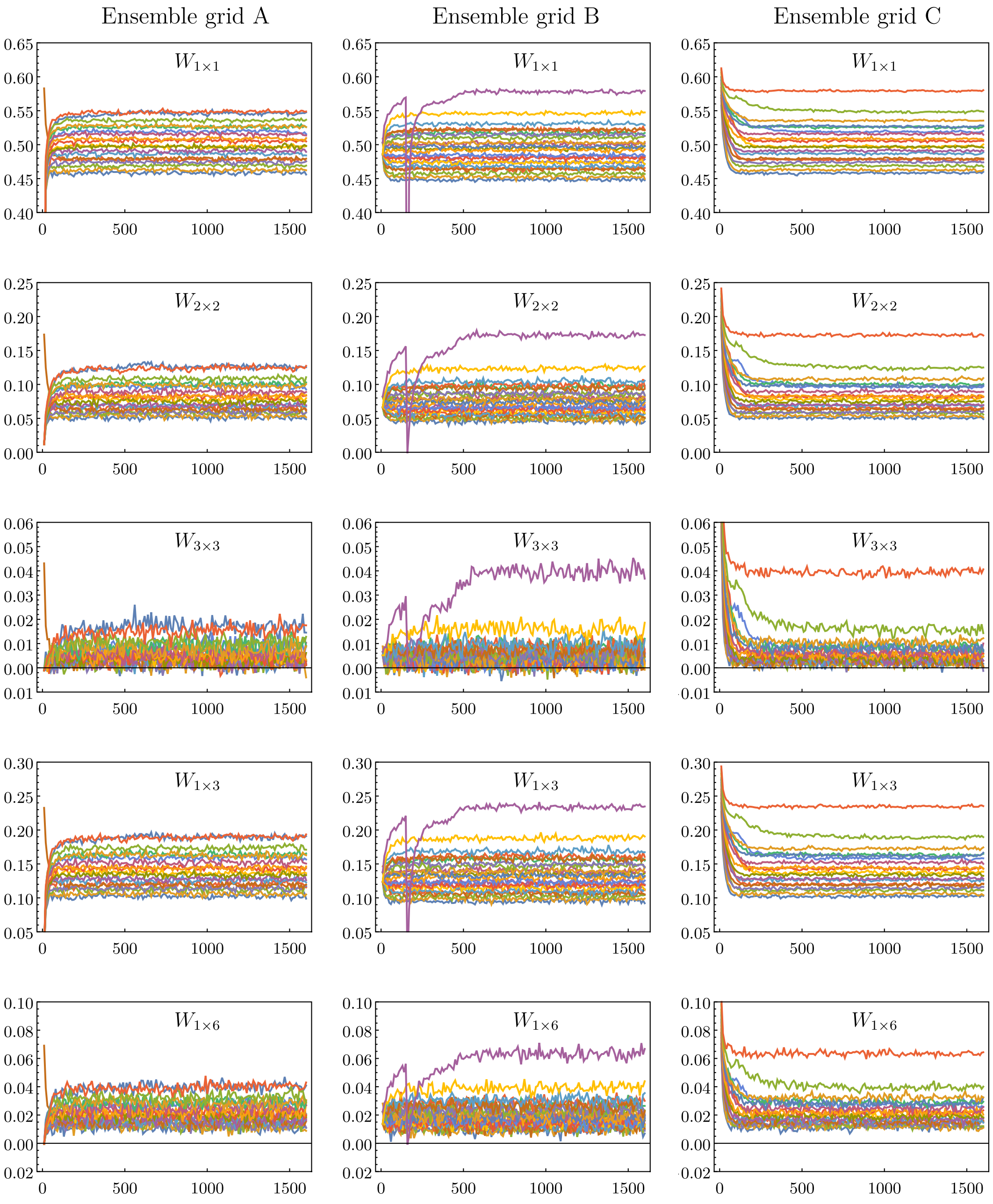


FIG. 19. The various Wilson loops, $W_{m\times n}$, on the first 1600 (of 10000) trajectories of each of the ensembles in Grid A (left column), Grid B (middle column) and Grid C (right column).

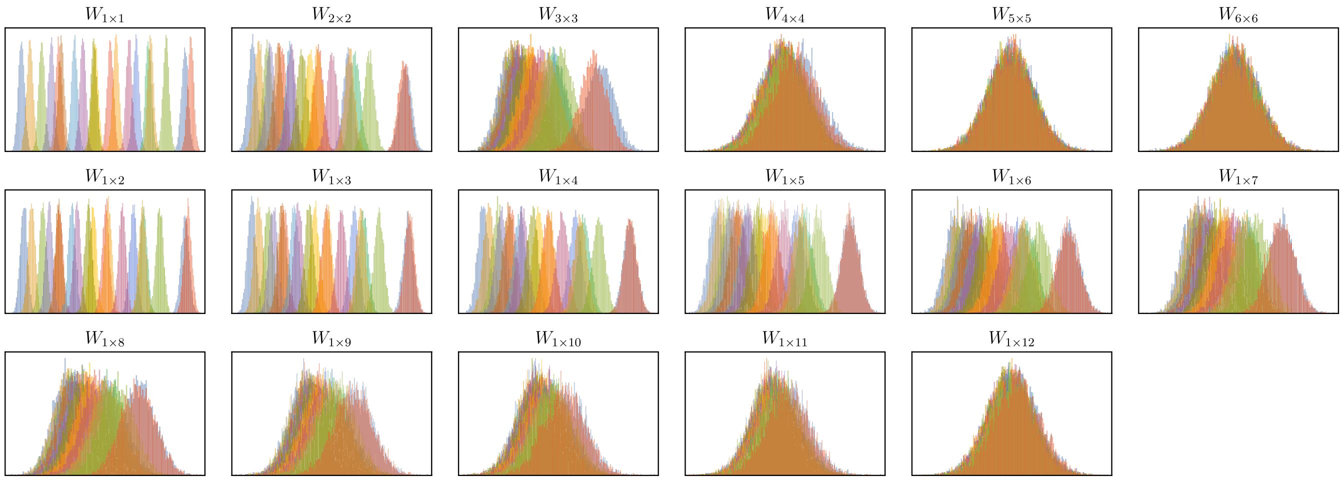


FIG. 20. The various Wilson loops, $W_{m\times n}$, on each of the ensembles in Grid A for all m, n combinations used in this work.

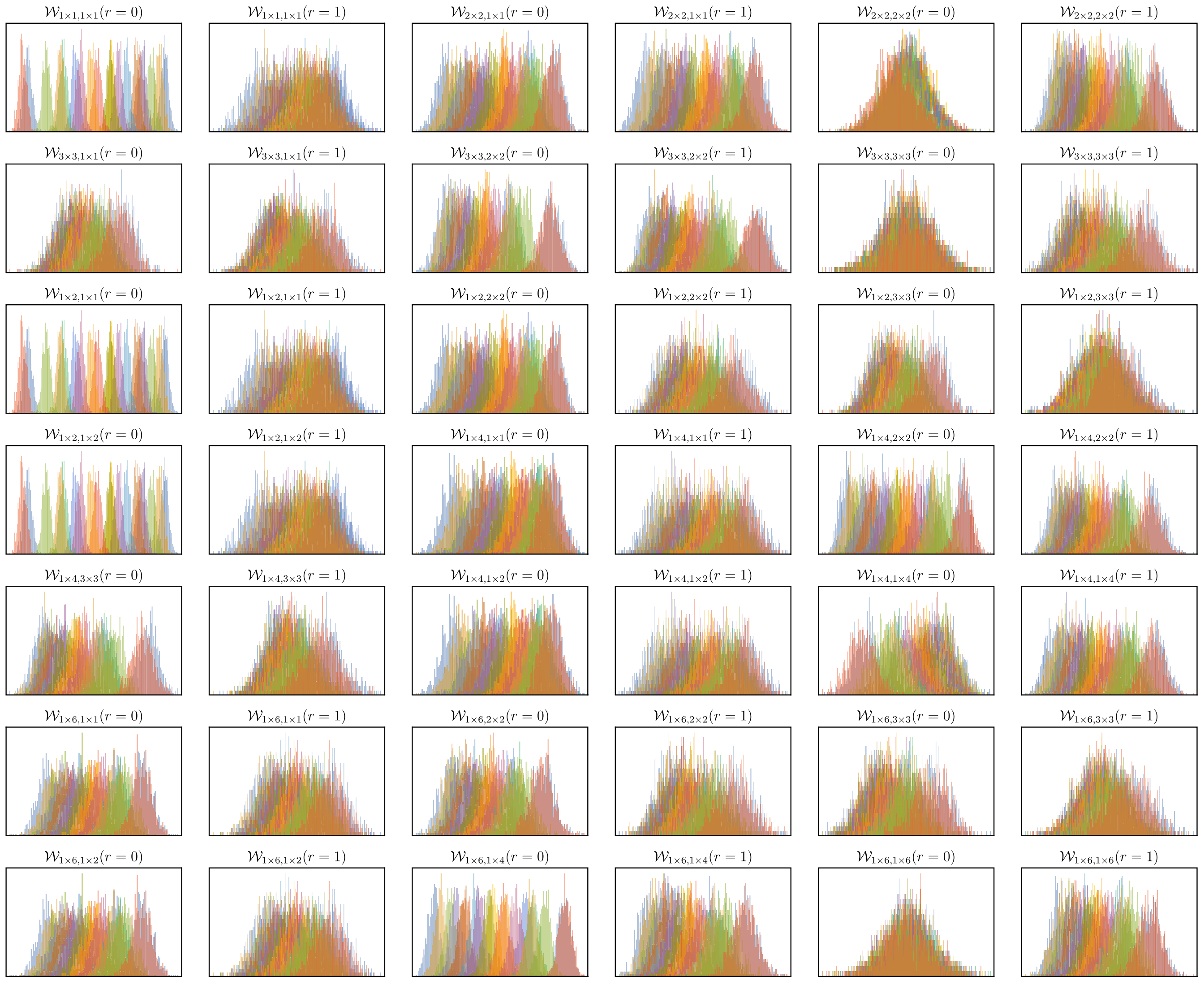


FIG. 21. The various Wilson loop correlators, $\mathcal{W}_{m\times n,p\times q}(r)$, on each of the ensembles in Grid A for a selection of choices of loop shapes and separations $r = 0$, 1.

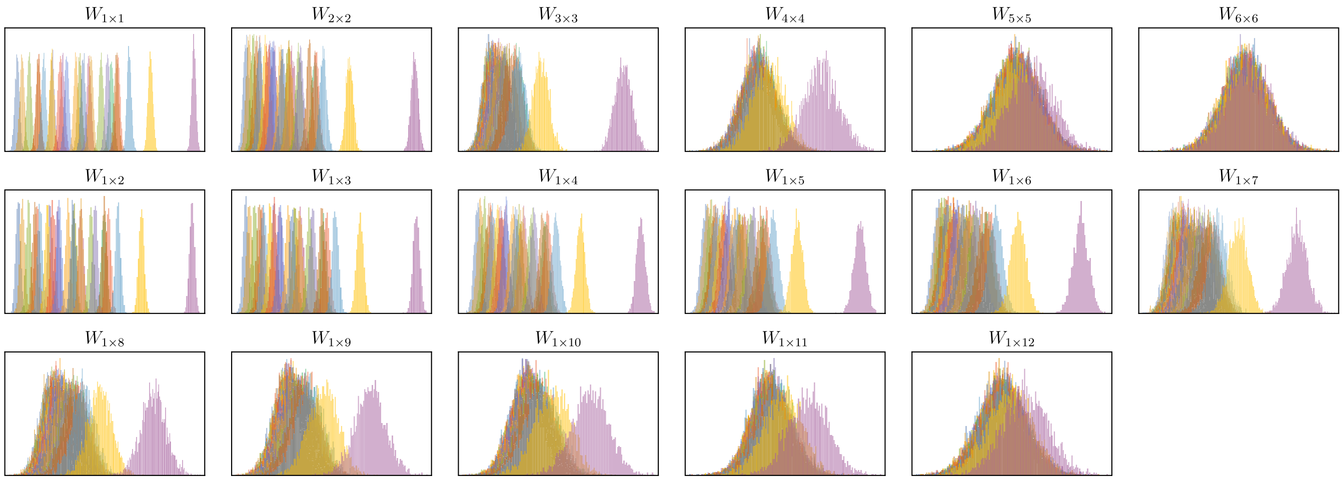


FIG. 22. The various Wilson loops, $W_{m\times n}$, on each of the ensembles in Grid B for all m, n combinations used in this work.

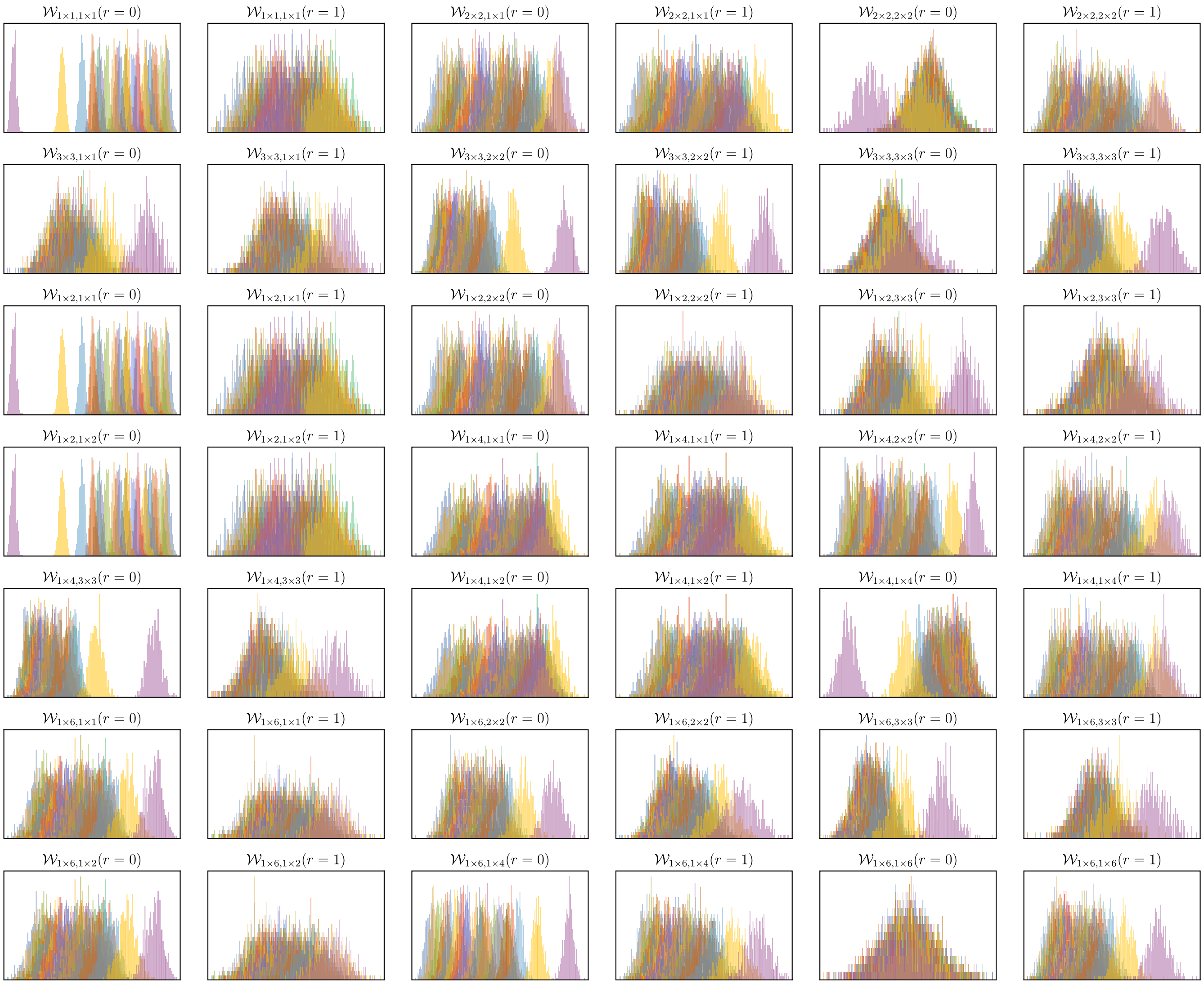


FIG. 23. The various Wilson loop correlators, $\mathcal{W}_{m\times n,p\times q}(r)$, on each of the ensembles in Grid B for a selection of choices of loop shapes and separations $r=0$, 1.

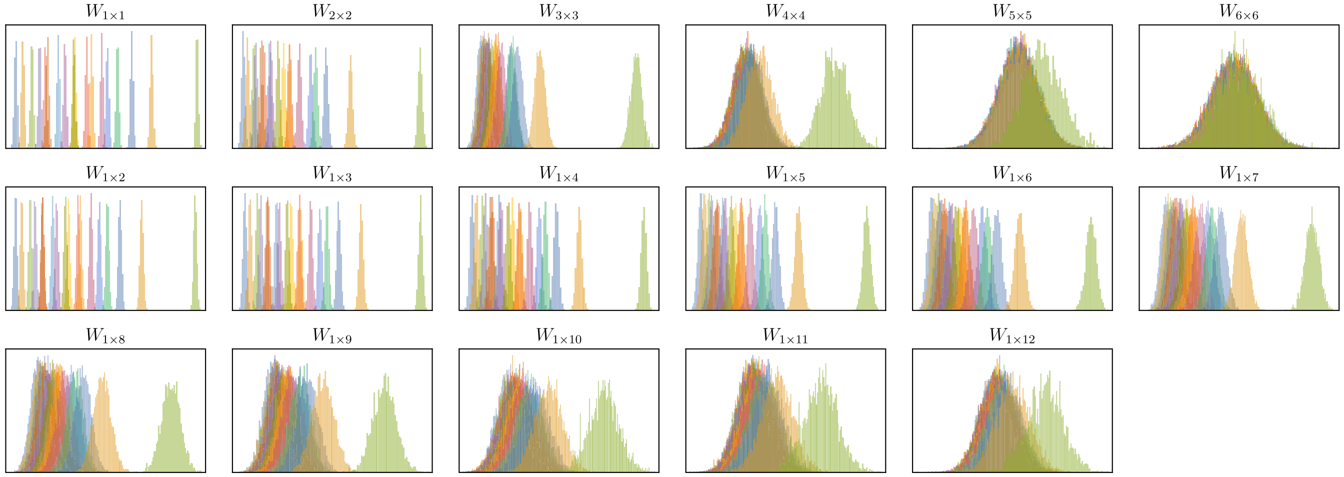


FIG. 24. The various Wilson loops, $W_{m\times n}$, on each of the ensembles in Grid C for all m, n combinations used in this work.

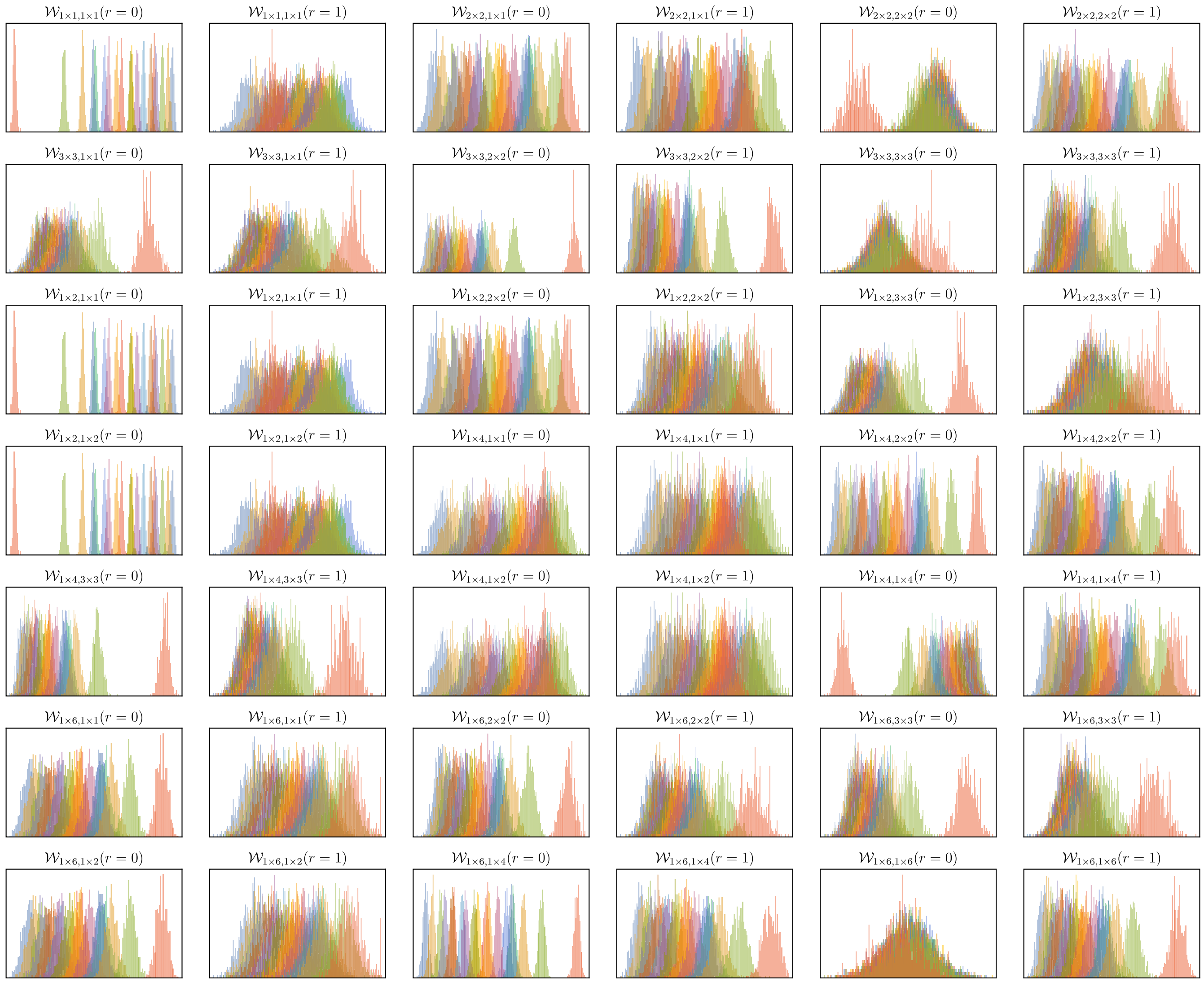


FIG. 25. The various Wilson loop correlators, $\mathcal{W}_{m\times n,p\times q}(r)$, on each of the ensembles in Grid C for a selection of choices of loop shapes and separations $r = 0$, 1.

[1] K. G. Wilson, Phys. Rev. **D10**, 2445 (1974); **10**, 45 (1974).
[2] H. J. Rothe, World Sci. Lect. Notes Phys. **43**, 1 (1992); **82**, 1 (2012).
[3] C. Gattringer and C. B. Lang, Lect. Notes Phys. **788**, 1 (2010).
[4] S. Aoki *et al.*, Eur. Phys. J. C **77**, 112 (2017).
[5] M. Constantinou, *Proc. Sci.*, CD15 (2015) 009, [arXiv:1511.00214].
[6] S. R. Beane, W. Detmold, K. Orginos, and M. J. Savage, Prog. Part. Nucl. Phys. **66**, 1 (2011).
[7] Z. Davoudi, EPJ Web Conf. **175**, 24(2018).
[8] H.-T. Ding, F. Karsch, and S. Mukherjee, Int. J. Mod. Phys. E **24**, 1530007 (2015).
[9] M. Lüscher, in *Modern perspectives in lattice QCD: Quantum field theory and high performance computing. Proceedings, International School, 93rd Session, Les Houches, France, August 3-28, 2009* (2010), pp. 331–399, arXiv:1002.4232.
[10] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, Phys. Lett. B **195**, 216 (1987).
[11] S. Schaefer, R. Sommer, and F. Virotta (ALPHA), Nucl. Phys. **B845**, 93 (2011).
[12] T. L. Bell and K. G. Wilson, Phys. Rev. B **11**, 3431 (1975).
[13] P. Hasenfratz and F. Niedermayer, Nucl. Phys. **B414**, 785 (1994).
[14] U. J. Wiese, Phys. Lett. B **315**, 417 (1993).
[15] P. Hasenfratz, Prog. Theor. Phys. Suppl. **131**, 189 (1998).
[16] J. Goodman and A. D. Sokal, Phys. Rev. Lett. **56**, 1015 (1986).
[17] R. G. Edwards, J. Goodman, and A. D. Sokal, Nucl. Phys. **B354**, 289 (1991).
[18] R. G. Edwards, S. JosFerreira, J. Goodman, and A. D. Sokal, Nucl. Phys. **B380**, 621 (1992).
[19] M. Grabenstein and K. Pinn, Phys. Rev. D **50**, 6998 (1994).
[20] M. G. Endres, R. C. Brower, W. Detmold, K. Orginos, and A. V. Pochinsky, Phys. Rev. D **92**, 114516 (2015).
[21] W. Detmold and M. G. Endres, Phys. Rev. D **94**, 114502 (2016).
[22] K. G. Wilson and J. B. Kogut, Phys. Rep. **12**, 75 (1974).
[23] T. Balaban, M. O'Carroll, and R. Schor, Commun. Math. Phys. **122**, 233 (1989).
[24] W. Schroers *et al.* (LHPC, SESAM), Nucl. Phys. B, Proc. Suppl. **129–130**, 907 (2004).
[25] K. C. Bowler, B. Joo, R. D. Kenway, C. M. Maynard, and R. J. Tweedie (UKQCD), J. High Energy Phys. 08 (2005) 003.
[26] S. R. Beane, P. F. Bedaque, K. Orginos, and M. J. Savage (NPLQCD), Phys. Rev. D **73**, 054503 (2006).
[27] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, arXiv:1608.07848.
[28] C.-D. Li, D.-R. Tan, and F.-J. Jiang, Ann. Phys. (Amsterdam) **391**, 312 (2018).
[29] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Phys. Rev. X **7**, 031038 (2017).
[30] G. Torlai and R. G. Melko, Phys. Rev. B **94**, 165134 (2016).
[31] G. Carleo and M. Troyer, Science **355**, 602 (2017).
[32] L. Wang, Phys. Rev. B **94**, 195105 (2016).
[33] L. Huang and L. Wang, Phys. Rev. B **95**, 035105 (2017).
[34] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B **95**, 041101 (2017).
[35] J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017).
[36] S. J. Wetzel and M. Scherzer, Phys. Rev. B **96**, 184410 (2017).
[37] P. Zhang, H. Shen, and H. Zhai, Phys. Rev. Lett. **120**, 066401 (2018).
[38] L. Wang, Phys. Rev. E **96**, 051301 (2017).
[39] A. Tanaka and A. Tomiya, arXiv:1712.03893.
[40] H. Shen, J. Liu, and L. Fu, arXiv:1801.01127.
[41] S. J. Wetzel and M. Scherzer, Phys. Rev. B **96**, 184410 (2017).
[42] J. E. Mandula, G. Zweig, and J. Govaerts, Nucl. Phys. **B228**, 91 (1983).
[43] R. G. Edwards and B. Joo (SciDAC, LHPC, UKQCD), Nucl. Phys. B, Proc. Suppl. **140**, 832 (2005).
[44] W. Detmold, M. McCullough, and A. Pochinsky, Phys. Rev. D **90**, 114506 (2014).
[45] R. Lewis, C. Pica, and F. Sannino, Phys. Rev. D **85**, 014504 (2012).
[46] M. Lüscher, J. High Energy Phys. 08 (2010) 071; 03 (2014) 092.
[47] N. Madras and A. D. Sokal, J. Stat. Phys. **50**, 109 (1988).
[48] K. Pearson, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**, 559 (1901), .
[49] H. Hotelling, J. Educ. Psychol. **24**, 417 (1933).
[50] H. Hotelling, Biometrika **28**, 321 (1936).
[51] J. Lin and S. Wong, Int. J. Gen. Syst. **17**, 73 (1990).
[52] J. Lin, IEEE Trans. Inf. Theory **37**, 145 (1991).
[53] S. Kullback and R. A. Leibler, Ann. Math. Stat. **22**, 79 (1951).
[54] D. M. Endres and J. E. Schindelin, IEEE Trans. Inf. Theory **49**, 1858 (2003).
[55] F. Österreicher and I. Vajda, Ann. Inst. Stat. Math. **55**, 639 (2003).
[56] F. X. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms* (Spartan Books, Washington, D.C., 1961).
[57] P. Werbos, Ph.D. thesis, Harvard University, 1974.
[58] S. Linnainmaa, Master's thesis, University of Helsinki, 1970.
[59] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, in *Parallel distributed processing: Explorations in the Microstructure of Cognition, Volume 1: Foundation*, edited by D. E. Rumelhart, J. L. McClelland, and the PDP research group (MIT Press, Cambridge, MA, 1986).
[60] F. Mosteller and J. W. Tukey, *Handbook of Social Psychology*, Vol. 2 (Addison-Wesley, Reading, MA, 1968).
[61] M. Stone, J. R. Stat. Soc. **36**, 111 (1974).
[62] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, arXiv:1207.0580.
[63] J. Ba and B. Frey, in *Advances in Neural Information Processing Systems 26*, edited by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Curran Associates, Inc., Burlington, Massachusetts, 2013), p. 3084.
[64] P. Baldi and P. Sadowski, Artif. Intell. **210C**, 78 (2014).
[65] R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. Seung, Nature (London) **405**, 947 (2000).

[66] R. Hahnloser and H. Seung, in *Advances in Neural Information Processing Systems 13*, edited by T. K. Leen, T. G. Dietterich, and V. Tresp (MIT Press, Cambridge, MA, 2001), p. 217.

[67] X. Glorot, H. Bordes, and Y. Bengio, in AISTATS (2001).

[68] D. P. Kingma and J. Ba, arXiv:1412.6980.

[69] C. Bishop, *Pattern Recognition and Machine Learning* (Springer-Verlag, New York, 2006).

[70] R. Linsker, Computer **21**, 105 (1988).

[71] H. Barlow, T. Kaushal, and G. Mitchison, Neural Comput. **1**, 412 (1989).

[72] P. A. P. Moran, Biometrika **37**, 17 (1950).

[73] R. C. Geary, The Incorporated Statistician **5**, 115 (1954).

[74] L. S. Yaeger, R. F. Lyon, and B. J. Webb, in *Advances in Neural Information Processing Systems 9*, edited by M. C. Mozer, M. I. Jordan, and T. Petsche (MIT Press, Cambridge, MA, 1997), p. 807.

[75] H. Baird, in *Document Image Defect Models* (IEEE Computer Society Press, Los Alamitos, CA, 1995), p. 315.

[76] L. Perez and J. Wang, arXiv:1712.04621.

[77] P. Simard, B. Victorri, Y. LeCun, and J. Denker, in *Advances in Neural Information Processing Systems 4*, edited by J. E. Moody, S. J. Hanson, and R. P. Lippmann (Morgan-Kaufmann, Red Hook, NY, 1992), p. 895.

[78] F. J. Király, A. Ziehe, and K.-R. Müller, arXiv:1411.7817.

[79] B. Schölkopf, C. Burges, and V. Vapnik, *Incorporating Invariances in Support Vector Learning Machines* (Springer Berlin Heidelberg, Berlin, Heidelberg, 1996), p. 47, ISBN 978.

[80] C. J. C. Burges and B. Schölkopf, in *Advances in Neural Information Processing Systems 9*, edited by M. C. Mozer, M. I. Jordan, and T. Petsche (MIT Press, Cambridge, MA, 1997), p. 375.

[81] Y. Nesterov, Sov. Math. Dokl. **27**, 372 (1983).

[82] K. Fukushima, Trans. IECE **J62-A(10)**, 658 (1979).

[83] K. Fukushima, Biol. Cybern. **36**, 193 (1980).

[84] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, and G. Wang, arXiv:1512.07108.