APS physics

CHORUS

This is the accepted manuscript made available via CHORUS. The article has been published as:

Digitizing gauge fields: Lattice Monte Carlo results for future quantum computers

Daniel C. Hackett, Kiel Howe, Ciaran Hughes, William Jay, Ethan T. Neil, and James N. Simone

Phys. Rev. A **99**, 062341 — Published 28 June 2019

DOI: 10.1103/PhysRevA.99.062341

FERMILAB-PUB-18-615-T

Digitizing Gauge Fields: Lattice Monte Carlo Results for Future Quantum Computers

Daniel C. Hackett,[1,*] Kiel Howe,[2,†] Ciaran Hughes,[2,‡]
William Jay,[1,2,§] Ethan T. Neil,[1,3,¶] and James N. Simone[2,**]

[1]*Department of Physics, University of Colorado, Boulder, Colorado 80309, USA*
[2]*Fermi National Accelerator Laboratory, Batavia, Illinois, 60510, USA*
[3]*RIKEN BNL Research Center, Brookhaven National Laboratory, Upton, New York 11973, USA*

Abstract

In the near-future noisy intermediate-scale quantum (NISQ) era of quantum computing technology, applications of quantum computing will be limited to calculations of very modest scales in terms of the number of qubits used. The need to represent numeric quantities using limited resources leads to digitization errors which must be taken into account. As a first step towards quantum simulations of realistic high-energy physics problems, we use ensembles of SU(2) lattice gauge fields generated with standard classical computing to explore the effects of digitizing elements of the gauge group to a finite set. We consider several methods for digitizing the group, finding the best performance from an action-preserving projection onto a mesh. Working in (3+1) dimensions, we find that using ~ 7 (qu)bits to represent each SU(2) gauge link induces a digitization error on the order of 10% in short-distance observables and 2% in long-distance observables. Promisingly, our results indicate that each SU(2) gauge link can be represented by $\mathcal{O}(10)$ (qu)bits, from which we estimate that a 16^3 SU(2) lattice could be simulated with no more than $\mathcal{O}(10^5)$ (qu)bits. Our results on digitization are also of interest as a form of lossy compression that could be used in high-performance classical computing to alleviate communications bottlenecks.

I. INTRODUCTION

Quantum computers offer the promise of solving problems which are presently intractable. In particular, simulating strongly interacting gauge theories on a digital quantum computer is an exciting prospect. Many interesting physics problems in lattice gauge theory remain intractable even for cutting-edge classical computers, including real-time dynamics of hadronization and thermodynamics at large quark-number density. While the lattice community has made impressive progress in studying hadronic physics directly from QCD [1–3], the heavy nuclear physics needed by neutrino experiments, like the quark-current form factors of Argon, remain beyond the limits of current classical computers. More fundamentally, some simulations are intractable on *any* classical computer. For example, storing the full wavefunction of a 500-qubit system would require more classical bits than there are atoms in the observable Universe[1].

Despite the eventual promise of quantum computing, speculation about near-term prospects for these devices in the "noisy intermediate-scale quantum" (NISQ) era suggests that qubit and gate resources will be severely limited [4]. On the other hand, due

* daniel.hackett@colorado.edu
† khowe@fnal.gov
‡ chughes@fnal.gov
§ wjay@fnal.gov
¶ ethan.neil@colorado.edu
** simone@fnal.gov

[1] On an N-bit classical computer, 2^N different numbers are accessible, but the memory needed to store m numbers is mN-bits. On a N-qubit quantum computer, 2^N different numbers are also accessible but these N-qubits can also store the 2^N values.

to their local nature, field theory Hamiltonians require less connectivity to implement than general quantum computing algorithms and may in fact be among the first realistic problems tractable on large-scale quantum computing devices. Nevertheless, efficient use of available qubit resources will be important to simulate a gauge theory on any near-future quantum computer. This requires us to confront "digitization error", which we define as the error due to representing continuously-valued quantum fields in finite numerical precision.

Field theories are physical systems with infinitely many degrees of freedom, and their simulation on any digital computer requires finite approximations. The standard approach restricts the fields to a discrete spacetime lattice in a finite volume, and the control and extrapolation of the resulting finite lattice-spacing error and finite-volume error is well-known in the literature on lattice field theory. However, there is a third finite approximation which is usually not emphasized: the quantum fields themselves must be restricted to take on discrete numerical values, limited by floating-point precision. In principle, this discretization yields a third approximation error, digitization error.

Of course, the use of 64-bit double-precision floating point numbers guarantees that such errors will be utterly negligible in most modern lattice calculations, although there can be concerns about reversibility of the hybrid Monte Carlo algorithm in global sums on large lattices [5]. We refer to the common use of 64-bit floating point numbers to represent gauge fields as *ultrafine digitization*.

Even in the early days of computational lattice field theory, the availability of classical bits was sufficient to avoid large digitization errors compared to other error sources. However, there was early interest in simulating with discrete subgroups of SU(N), which would allow the use of lookup tables to speed up the computations [6]. These studies found that the subgroup digitization created an artifact lattice phase at weak coupling. For the 120-element largest point subgroup of SU(2), this artifact phase was at sufficiently weak coupling that some parameters of interest remained accessible [7, 8]. This was not the case for the largest 1080-element point subgroup of SU(3) [9]. However, an additional study [10] found that this phase could be pushed to weaker coupling in SU(3) by using a finer digitization generated by interpolating between SU(3) elements.

In this work, our goal is different: we study the impact of extreme memory restrictions, rather than to improve computational efficiency. We ask how coarsely the digitization of group elements onto a finite set may be done before inducing large systematic errors. By translating the size of the digitized group into the number of (qu)bits required to represent a gauge link, we can estimate the number of (qu)bits required to avoid digitization error at some level. We study several different group-truncated digitization schemes to gain insight into the features of an optimal scheme.

More specifically, we consider the effect of digitizing pure SU(2) gauge theory. Although not directly relevant to QCD, this theory is computationally less expensive than SU(3), allowing us to generate large data sets and thereby avoid statistical errors that might introduce ambiguity in our results otherwise. Furthermore, SU(2) is isomorphic to the four-dimensional unit sphere S_3, providing an intuitive picture for digitization that would be lacking for larger gauge groups. Our work shares some characteristics with a recent paper by Urbach [5], although his focus is on the hybrid Monte Carlo algorithm while we focus on the observables, and his study is concerned with a much finer digitization than our results will explore.

Other work on gauge group digitization in quantum computing for high-energy physics has focused on finding few qubit problems that can be implemented on NISQ technol-

ogy. For example, [11] examines the $(1+1)$-D Schwinger model with one and two spatial sites using a 2-qubit digital quantum computer and shows that the quantum computer produces observables which agree with analytic results for a large time range before fidelity becomes appreciable. Also, using an analogue quantum computer, [12] computes two-body and three-body forces between heavy-mesons in the Schwinger model. The authors of [13] formulate the SU(N) Hamiltonian with matter fields for an analogue quantum simulation and study the digitization error in the $(2+1)$-D compact QED ground state energy for a range of couplings. All these works digitize the gauge group by imposing a cutoff on the allowed energy eigenstates [13]. This approach has the advantage of conceptual clarity: keeping only the low-energy eigenstates should leave low-energy physical observables unaffected. Moreover, the eigenstate digitization always enforces gauge invariance but breaks unitarity at finite truncation [13]. [13] show the mathematical mapping between the eigenstate- and group-representations of the infinite dimensional gauge Hilbert space, yet it is also unclear how practically the resource requirements of the eigenstate-truncation method compares with the group-truncation schemes that we study here, especially for short-range observables. Notably, the group truncation always enforces unitarity but can break gauge invariance at finite truncation and is more natural for lattice gauge theory simulations on classical computers.

Previous works have examined the eigenstate truncation methodology in models of $SU(2)$ within the Hamiltonian approach. Of these, [14] classically performs exact diagonalisation of an $(1+1)$D $U(2)$ Hamiltonian in a quantum link model with different spatial lattice extents ranging from $L_s = 2-18$. There, they use one rishon-per-link and study the energy splitting between the two lowest energy eigenstates and the real-time evolution of an order parameter. In [15], the authors classically examine a $(1+1)$D $SU(2)$ tensor network approximation in a quantum link model with two rishons-per-link and study the quantum phase diagram[2]. In [16], the authors extensively study (2+1)D $SU(2)$ with projected entangled pair states (PEPS) in the tensor network framework. For this specific construction of PEPS observables, the physical states are restricted to the $j \leq 1/2$ sector. In this approximation the observables with a truncated gauge Hilbert space are exactly identical to the model results if they were calculated in the full infinite-dimensional gauge Hilbert space. Also, the authors of [17] use the physical subspace to find an efficient eigenstate truncation in a $(1+1)$D $SU(2)$ model. Allowing up to $j = 2$ eigenstates, and using the matrix product state approximation in the tensor network framework, the authors are able to compute a large range of physical quantities in different limits.

The outline of this work is as follows: in Section II we discuss several different schemes for digitizing SU(2) as well as the projections to coarser digitizations. In Section III we describe how we generate SU(2) lattice ensembles and our lattice gauge theory methodology used to compute observables. In Section IV, we present the main results of our work: the error induced by our group-truncated digitization and projection schemes. Finally, in Section V we discuss our conclusions.

[2] In quantum link models, the continuous gauge operators of $SU(N)$ are embedded into a $SU(2N)$ algebra which are fermionic, and thus have finite-dimensional Hilbert spaces. However, here one has to show how to get the correct continuum physics out of the embedded simulated $SU(2N)$ model. This can be done by adding an extra compact dimension to the model and then separately taking this extra dimension correlation length to infinity in a numerical study.

II. DIGITIZING SU(2)

In this work we digitize the gauge group by reducing the infinite set of SU(2) matrices to a finite set $\mathcal{A}$ of size $|\mathcal{A}|$. Each such digitization scheme defines a different surjective mapping $f : \mathrm{SU}(2) \to \mathcal{A}$. The elements of $\mathcal{A}$ may or may not be elements of SU(2). We write $\mathcal{A}_f$ to denote the set $\mathcal{A}$ associated with a given digitization f.

There are many different choices for the digitization scheme f. Our objective is to compare several reasonable and well-motivated schemes which reproduce a set of physical observables to a predetermined level of accuracy with the smallest set $\mathcal{A}$, thereby requiring the least resources. In the following we discuss the different digitization schemes we will study in this work. We view our list as a base for future improvement rather than an exhaustive study of all possible digitizations.

In what follows, we parametrize any SU(2) element in the fundamental representation as

$$g = \begin{pmatrix} a + ib & -c + id \\ c + id & a - ib \end{pmatrix} \tag{1}$$

where a, b, c, and d are real numbers in $[-1, 1]$. As the determinant of any SU(2) matrix is unity, $a^2 + b^2 + c^2 + d^2 = 1$ and there are only three independent real degrees of freedom.

Since SU(2) and the sphere S^3 are diffeomorphic, any SU(2) matrix may be written as a (Euclidean) four-dimensional unit vector (a, b, c, d). In this sense, a digitization of SU(2) is a finite set of points on or near S^3.

A. Fixed-point Digitization

In typical lattice gauge theory simulations, an SU(N) matrix is typically represented as N^2 complex numbers, with unitarity and the unit determinant condition enforced by hand. Real numbers are typically represented as 64-bit double-precision floating point numbers (or briefly, doubles). Hereafter, we refer to this as the "matrix-of-doubles" representation of SU(N_c) matrices.

In a standard double, one bit s represents the sign of the number, 11 bits represent the integer exponent $-1024 < e < 1023$,[3] and the remaining 52 bits represent the normalized significand. Denoting the bits of the normalized significand as m_i, the significand $1 \leq S < 2$ is a fixed-point number whose value is given by

$$S = 1 + \sum_{i=1} \frac{m_i}{2^i}. \tag{2}$$

Taken together, the value represented by the 64 bits of the double is $(-1)^s \times S \times 2^e$. A double can represent values as small as $2^{-1024} \sim 10^{-39}$ and as large as $2^{1024} \sim 10^{39}$, and the values that a double can represent grow exponentially denser closer to zero. This representation is not optimal. Most glaringly, the real numbers in an SU(N) matrix are bounded between -1 and 1, so half of the values a double can represent ($e > -1$) are wasted.

A less wasteful representation for the real numbers in an SU(N) matrix is the simple fixed-point numbers, with the most significant digit starting at 1/2 [18]. Denoting the bits of the fixed-point number as f_i, with $f_0 = s$ as a sign bit, the value represented by p-bits of fixed-point precision is

$$(-1)^s \sum_{i=1}^{p-1} \frac{f_i}{2^i} \tag{3}$$

which can represent 2^p values evenly spaced between -1 and 1. This distribution of possible values is better-suited for lattice data: the distribution of values in typical lattice data is closer to flat than to exponentially spiked about zero[4].

[3] $2^{11} = 2048$.

[4] It is straightforward to plot the distribution of SU(N) values and check that it is closer to flat rather than being exponentially spiked at the origin.

Our fixed-point truncation scheme is simply to convert the doubles of an SU(2) matrix to p-bit fixed-point numbers. More specifically, we first convert the doubles a, b, and c to 64-bit fixed-point numbers. To truncate each number, we cut off the $64 - p$ least-significant bits of each number and then, to avoid having to implement fixed-point arithmetic, convert them back to doubles. To maintain the unit determinant condition to p bits of precision, we compute $d^2 = 1 - a^2 - b^2 - c^2$ at full precision (while keeping the original sign of d) and then apply the same truncation procedure to the resulting value. It is important to note that this operation is only unitary and gauge-invariant to p bits. We return to this key point in in Section IV.

B. Indexed Mesh Digitization

The above fixed-point digitization scheme has several drawbacks. First, representing an SU(N) matrix using fixed-point values is still wasteful. The independent real degrees of freedom a, b, c in fixed-point representation parametrize an even grid of $(2^p)^3$ points over the box $(-1, 1)^3$, but because $|d| \leq 1$ and $d^2 = 1-a^2-b^2-c^2$, all values of $a^2+b^2+c^2 > 1$ are wasted ($\sim$ 48% of all possible values). Furthermore, in practice, all of the points lie slightly off the unit sphere and are thus not elements of SU(2), leading to violations of unitarity and gauge invariance. In principle, this issue is shared by high-precision floating point numbers.

For a different digitization of the gauge group, we may consider simply choosing some finite subset of SU(2) elements as the digitization. It is easy to visualize SU(2) as a unit three-sphere and imagine the subset as a discrete "mesh" of v points lying on the sphere. Each element of the subset may then be represented by its index in this subset. This requires only ceil $\log_2 v$ bits per gauge link[5].

[5] With N bits we can represent $v = 2^N$ numbers.

The most obvious choice for a mesh is a discrete subgroup of SU(2). Abelian subgroups are clearly inadequate, since the non-Abelian nature of the group plays a critical role in its nonperturbative dynamics. Creutz, Jacobs, and Rebbi considered this problem previously from a slightly different perspective [6]. They found that the finite subgroups of SU(2) with sufficient non-Abelian structure to avoid significant distortions of physical results are

1. the 24-element tetrahedral subgroup $\bar{T}$,
2. the 48-element octahedral subgroup $\bar{O}$, and
3. the 120-element icosahedral subgroup $\bar{I}$ or $\bar{Y}$.

Whether these subgroups are large enough for practical use as a digitization scheme on a quantum computer is an important question, which we revisit in Sec. IV.

To get finer digitizations, we can simply pick a larger subset of elements which are distributed approximately evenly across the unit three-sphere. As the three-sphere is generally not diffeomorphically equivalent to the polytope, one cannot find a general exactly uniformly distributed mesh. Geodesic meshes are the familiar solution to this problem in three dimensions and generalize straightforwardly to 4D.

We generate geodesic meshes using the `mvmesh` R package [19]. To generate the meshes, we use the "edgewise" algorithm, which begins with a 4D octohedron then uniformly tesselates each simplicial face with smaller simplices before "inflating" the resulting mesh to an approximate sphere. The package also offers a "dyadic" algorithm, which also begins with the octohedron, but recursively tesselates each simplicial face with the simplest simplicial tesselation. These dyadic meshes are more even, but are defined for fewer different values of mesh size v. For similar mesh sizes, all observables that we have examined agreed, so

we only present results for edgewise meshes. This approach is similar to the one considered in Ref. [20], but less sophisticated in that we do not generate our meshes by subdividing polytopes corresponding to the finite subgroups of SU(2).

Multiplication in meshes must be implemented using a lookup table [10, 20]. If the mesh is a subgroup of SU(2), then multiplication is exact. If the mesh is not a closed subset of SU(2), then the result of multiplying two elements together must be projected back onto the mesh. In this exploratory study we immediately return to the usual matrix-of-doubles representation after projecting links to meshes, and do not perform any operations in the indexed mesh representation. As such, we are unable to quantify the effect of projection after multiplication. Our approach is equivalent to using lookup tables to compute traces of multiple mesh elements exactly, as used in Ref. [20].

In this study, we project SU(2) matrices from existing lattice gauge fields computed in the ultrafine digitization to a coarser mesh digitization. This introduces another potential source of error which compounds the error due to digitization alone, thus it is important to also consider how we perform projections. To get an idea for how much error is due to projection rather than digitization, we tried several different projection schemes, described below.

1. *L2 Norm*

One can project into the mesh by replacing the SU(2) matrix by its nearest neighbor in the mesh. This requires a metric on the group, for which we use the natural invariant complex matrix normed distance $D(A, B) = ||A - B||$ where $||M|| = \mathrm{Tr}(M^\dagger M)$. This amounts to the L2 norm $\Delta a^2 + \Delta b^2 + \Delta c^2 + \Delta d^2$ between two points on the three-sphere representation of SU(2). Consequently, this projection scheme simply chooses the nearest mesh point on S_3.

2. *Action-Preserving Rounding*

Another idea is to engineer our projection method to preserve physical quantities. Ideally, we would like to project each link to the mesh such that all Wilson loops on a lattice are changed as little as possible. In practice, measuring longer Wilson loops is computationally expensive and finding the exact best projection is intractable, growing combinatorically with volume V and number of mesh points v. Instead, we define action-preserving rounding (APR) as projection which tries to preserve the local action density. The Wilson gauge action is a function of the plaquette operator only (as defined in Eq. (6)), so this amounts to trying to preserve the value of individual plaquettes. One may think of this method as trying to project gauge-invariantly.

In practice, for each plaquette, we replace one link at a time with an element from the mesh, choosing the mesh element which makes the new value of the plaquette closest to the original undigitized value. There is freedom in this algorithm to choose the order in which one replaces the links in each plaquette. Computationally, it is most straightforward to replace all links in each dimension before moving onto another. We arbitrarily choose the order $XYZT$ and have not examined the effects of choosing different orderings.

III. LATTICE GAUGE THEORY METHODOLOGY

This work uses standard techniques in the lattice gauge theory literature; we refer the reader to [21, 22] for a discussion of this methodology. For this work, we generate SU(2) lattice gauge field configurations using MILC code [23] adapted to run $N_c = 2$. We use the Wilson plaquette action for our discretization of the pure gauge action [24]. In each ensemble, we save a gauge configuration after every 1000 Monte Carlo trajecto-

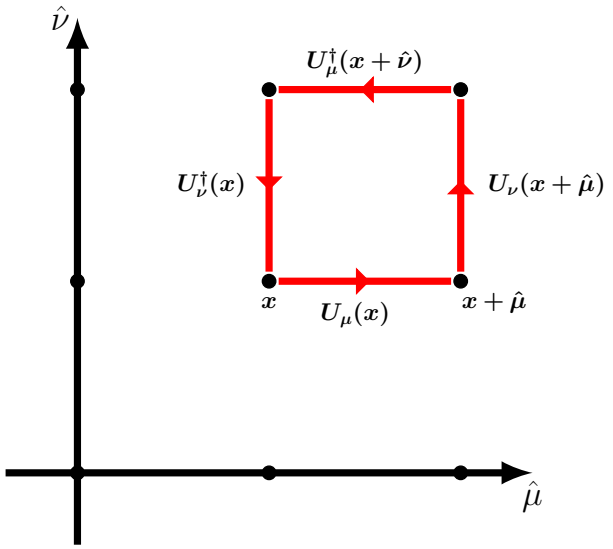

FIG. 1. The plaquette $\square_{\mu\nu}(x)$ constructed from the gauge links $U_\mu(x)$.

ries, where a trajectory is four overrelaxation steps and one quantum heat bath step. The typical autocorrelation times in our data set are such that 1000 trajectories is sufficient to decorrelate the observables that we consider.

We generated multiple different ensembles upon which to compute observables. One is a high-statistics zero-temperature ensemble of 1000 configurations with volume $V = 12^4$ at $\beta = 2$. The remaining ensembles contain 100 configurations each. They include 38 finite-temperature ensembles with $V = 12^3 \times 6$ and additional 6 zero-temperature ensembles with $V = L^4$ for $L \neq 12$.

For this initial investigation we generate "undigitized" ensembles with the standard classical Monte Carlo ultrafine digitization, which uses matrices of 64-bit doubles to represent gauge links. We do not generate Monte Carlo ensembles with any other coarsely digitized gauge group.

To study the effects of gauge group digitization, we take an undigitized ensemble and project all gauge links to a coarser digitization. For computational convenience, after projecting to the coarser digitization we return the gauge links to the matrix-of-doubles representation. On the resulting digitized ensemble, we measure observables and determine how they have been affected.

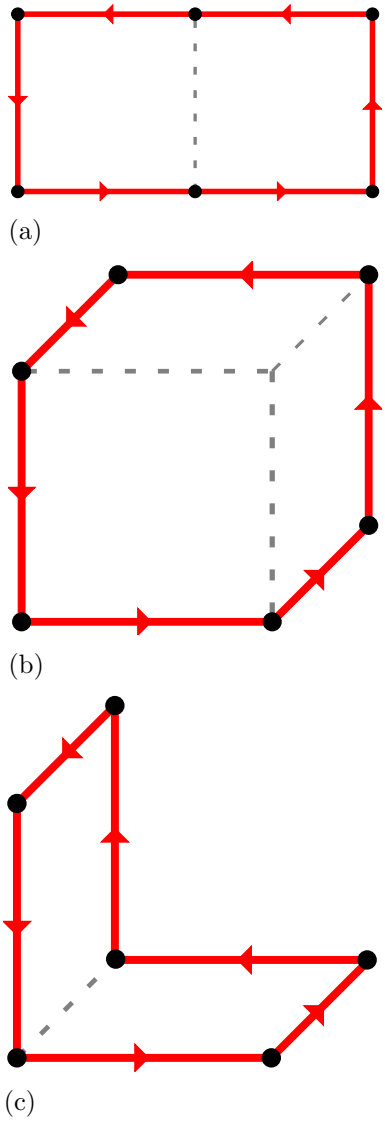

FIG. 2. The three perimeter six Wilson loops, the (a) rectangle $\mathcal{P}_1$, the (b) parallelogram $\mathcal{P}_2$, and (c) bent rectangle $\mathcal{P}_3$. Dashed lines are drawn to guide the eye.

A. Computed Physical Observables

This work focuses on the simplest gauge-invariant objects, Wilson loops. A Wilson loop is the trace of a product of gauge links

$U_{x,\mu}$ along a closed loop $\mathcal{L}$

$$W_{\mathcal{L}}[U] = \mathrm{Tr} \prod_{(x,\mu)\in\mathcal{L}} U_{x,\hat{\mu}}, \tag{4}$$

where x denotes the sites in the loop and $\hat{\mu}$ denotes the direction of the link. Wilson loops are best understood physically through their connection to the potential $V(r)$ between static color charges. Careful discussions of this connection are available in standard textbooks on lattice gauge theory [21, 22] as well as continuum field theory [25]. The present discussion only needs the fact that a rectangular Wilson loop W of spatial length r and temporal length t scales as [24]

$$W(t,r) \sim e^{-tV(r)}. \tag{5}$$

To extract the potential in practice, we construct rectangular Wilson loops for many different values of t and r. At fixed spatial separation r', we fit the lattice data to the decaying exponential in Eq. (5) in order to obtain a value for $V(r')$. Repeating this process for all spatial separations yields the potential $V(r)$. These techniques are standard in the lattice literature [21, 22]. This work however studies the behavior of the potential under different gauge group digitizations.

A Wilson loop of particular importance is the plaquette, the building block of the Wilson plaquette gauge action [24]. The plaquette is a 1×1 square loop of gauge links, as depicted in Fig. 1 and is the simplest gauge-invariant observable that can be measured on a hypercubic lattice. More quantitatively, the plaquette at site x on the 4D lattice with extent in the μ and ν directions is

$$\square_{x,\mu\nu} = \mathrm{Re}\,\mathrm{Tr}\left[U_{x,\hat{\mu}} U_{x+\mu,\hat{\nu}} U^{\dagger}_{x+\nu,\hat{\mu}} U^{\dagger}_{x,\hat{\nu}}\right] \tag{6}$$

where $U_{x,\hat{\mu}}$ is the gauge link in the μ direction at site x. The three topologically distinct perimeter-six Wilson loops $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_3$, depicted in Fig. 2, are longer cousins of the plaquette. The ensemble expectation values of these quantities are their averages over all orientations, every site on the lattice, and each gauge configuration in the ensemble. The unimproved Wilson gauge action is a function of the plaquette only, while improved actions are typically functions of longer Wilson loops, like the perimeter-six loops. We include perimeter-six Wilson loops in the study not just as additional observables, but also because we anticipate improved actions may play a role in simulations on quantum computers.

The Polyakov loop is the shortest Wilson loop that winds around the temporal direction of the lattice once, i.e.,

$$\Omega_{\mathbf{x}} = \mathrm{Tr} \prod_{t=1}^{t=N_t} U_{(\mathbf{x},t),\hat{t}} \tag{7}$$

where $U_{(\mathbf{x},t),\hat{\mu}}$ is the gauge link in the $\hat{\mu}$ direction at site $(\mathbf{x}, t)$. In SU(N_c) pure gauge theory, the Polyakov loop is an order parameter for the finite-temperature deconfinement transition [26]. Unlike for $N_c > 2$ where it is generally complex, the Polyakov loop is real-valued in SU(2). At low temperatures when the system is in a confined phase, the Polyakov loop is protected by a symmetry and therefore vanishes. Increasing the temperature of the system (corresponding to simulations at larger β or shorter N_t) eventually results in a phase transition where this symmetry is spontaneously broken and the Polyakov loop acquires a nonzero expectation value, which is interpreted as a sign of deconfinement.

B. Mesh digitization and importance sampling

Before showing our results, we discuss our expectations for how mesh digitization will affect observables, based on properties of Monte Carlo simulations of lattice field theories. A naive approach to simulating gauge theories would generate SU(N) gauge field configurations with gauge links randomly distributed by the Haar measure, and weight

them by the action term e^{-S} when computing observables[6]. Monte Carlo simulations instead use importance sampling, which includes the e^{-S} term as part of the measure when randomly generating configurations. This induces correlations (over an ensemble of gauge configurations generated with importance sampling) between all gauge links.

The Haar measure has the property that

$$\int dU \, \mathrm{Tr}\, U = 0 \tag{8}$$

where U is an SU(N) matrix. The integral Eq. (8) can be thought of as the expectation value of a Wilson loop in a theory where $e^{-S} = 1$. It follows that the correlations between gauge links due to the e^{-S} term are what allow for nonzero Wilson loops, and that the expectation value of any Wilson loop constructed from completely random SU(N) matrices is zero. We thus expect that making gauge links more random (or equivalently, less correlated) will suppress the expectation values of Wilson loops.

To interpret the systematic error seen in our digitized results below in Sec. IV, note that projecting an ultrafine-digitized SU(2) matrix to a coarser digitization can be thought of as displacing that matrix in the group manifold. Displacements applied to different gauge links are completely uncorrelated for fixed-point truncation and projection to meshes with the L2 scheme, and less so for APR projection. We can thus think of the effect of truncation or projection as random kicks which add incoherent noise to the gauge links. This noise washes out correlations between gauge links induced by importance sampling.

Taken together, this suggests that projection to coarser meshes will disrupt the correlations between gauge links induced by importance sampling and thereby suppress the expectation values of Wilson loops. When projecting to finer meshes the correlations between gauge links are damped but remain significant. However, when projecting to increasingly coarser meshes correlations become small and the gauge links appear random within SU(2). In this case, it becomes comparable to performing a path integral with $e^{-S} = 1$ and restricting the group integration to those matrices in the mesh, approximating the integral over SU(2) in Eq. (8).

As discussed in Sec. IV, our results are broadly consistent with this narrative. The values of all Wilson loops that we measure are increasingly suppressed by projections to coarser digitizations. Furthermore, correla-

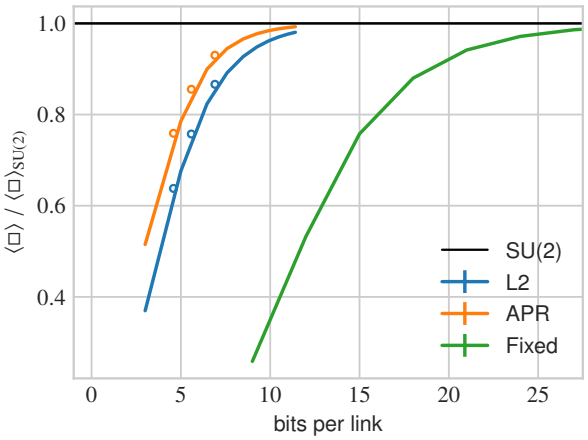

FIG. 3. The relative error in the plaquette expectation value (defined in Sec. III A) as a function of bits-per-link for different digitization and projection schemes, normalized by its undigitized value. These results are computed on an ensemble of 1000 configurations with $V = 12^4$ and $\beta = 2$. Representing an SU(2) link as a mesh element requires $\log_2 v$ bits, where v is the number of mesh points. A fixed-point digitization of precision p requires $3p$ bits. In the legend, L2 labels projection to a mesh using the L2 norm, while APR labels projection using the action (plaquette) preserving scheme, both defined in Sec. II B. Circles indicate projections onto the finite subgroups $\bar{T}_{24}$, $\bar{O}_{48}$, and $\bar{I}_{120}$.

[6] This approach is intractably inefficient, as the vast majority of possible gauge configurations are exponentially suppressed by the action term.

tions over longer distance scales (i.e., large scale structure in the gauge fields) should be more robust against the addition of incoherent noise compared to correlations over short distance scales (i.e., fine structure in the gauge fields). We observe that for a particular digitization and projection scheme the error induced in the static potential $V(r)$ is less than that induced in any of the shorter-range observables that we examine. We also see the error in the static potential $V(r)$ smoothly decreases as r increases (cf. Fig. 10 below).

IV. RESULTS

Here we empirically quantify the systematic error that is introduced by the different digitization and projection schemes that we have studied. We compare the different schemes based on the number of bits required to represent a gauge link, or "bits-per-link". If p is the number of bits of precision for each fixed-point number (including the sign bit), and it takes three fixed-point numbers to represent an SU(2) matrix parsimoniously, then the bits-per-link is $3p$ for a fixed-point digitization. To be explicit, in this scheme there can be 2^{3p} (potentially bad[7]) different representations of SU(2) matrices to p bits of precision. On the other hand, for a mesh of size v the number of bits-per-link is the number of bits required to index the mesh, $\log_2 v$. Again, to give an explicit example, a mesh with $v = 4$ SU(2) matrices requires a two-bit index to provide $2^2 = 4$ unique labels, with one label for each matrix. In practice we do not have fractional bits, so $\log_2 v$ must be rounded up to the nearest integer. We do not do this here to keep our curves smooth so that the reader may interpolate.

We now show our results. First, in Fig. 3, we show the plaquette expectation value against bits-per-link for different digitization and projection schemes. We empirically find that decreasing the bits-per-link of our digitization schemes reduces the plaquette expectation value towards zero, consistent with our arguments in Sec. III B. The fixed-point scheme performs drastically worse than the mesh-based schemes, requiring at least twice as many bits to achieve the same error. In part, this is due to the wastefulness of the fixed-point representation as discussed in Sec. II A. Additionally, each matrix is only unitary to p-bit precision in this scheme, in contrast to the mesh-based schemes where each matrix is exactly in SU(2). For the mesh-based schemes, projection with APR outperforms projection using the L2 norm, especially on finer meshes, where the induced systematic error is less than half the magnitude. We observe this discrepancy throughout our data, which suggests that the dominant source of error in our data is projection, rather than inherent to digitization. Projections onto discrete subgroups induce slightly less error than projections onto geodesic meshes of equivalent size. However, projections to sufficiently fine geodesic meshes outperform even the largest discrete subgroup of SU(2).

Next, Fig. 4 shows the effects of projection and digitization on the expectation values of the three perimeter-six Wilson loops. These observables are affected by digitization similarly to the plaquette (cf. Fig 3). It also appears that each operator at some specific bits-per-link is suppressed by the same factor consistent with the hypothesis of projection adding incoherent noise to the gauge links, as discussed in Sec. III B.

Fig. 5 shows the volume dependence of the effect of projection and digitization on the plaquette expectation value. We examine L^4 lattices varying L while keeping all other physical scales (i.e., β) fixed. The $L = 12$ data of this plot is the same as is used in Fig. 3. As can be observed, curves for different volumes essentially overlap on this y-axis

[7] See Sec. II A about the wastefulness of the fixed-point representation.

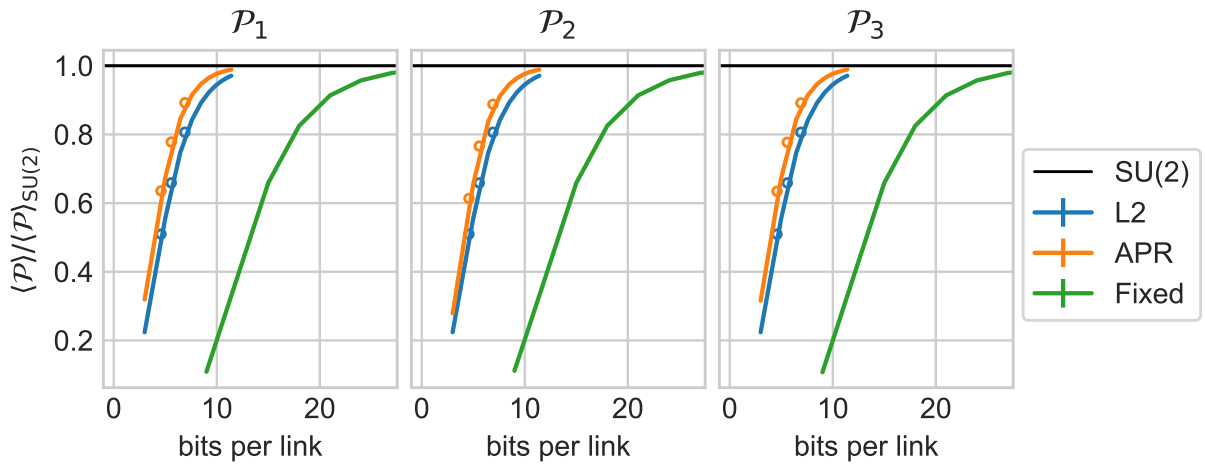

FIG. 4. The relative systematic error from digitizing the three topologically-distinct perimeter-six Wilson loops (shown in Fig. 2) as a function of bits-per-link for different digitization schemes, normalized to its undigitized value. These results are computed on an ensemble of 1000 configurations with $V = 12^4$ and $\beta = 2$. Refer to Fig. 3 for more details.

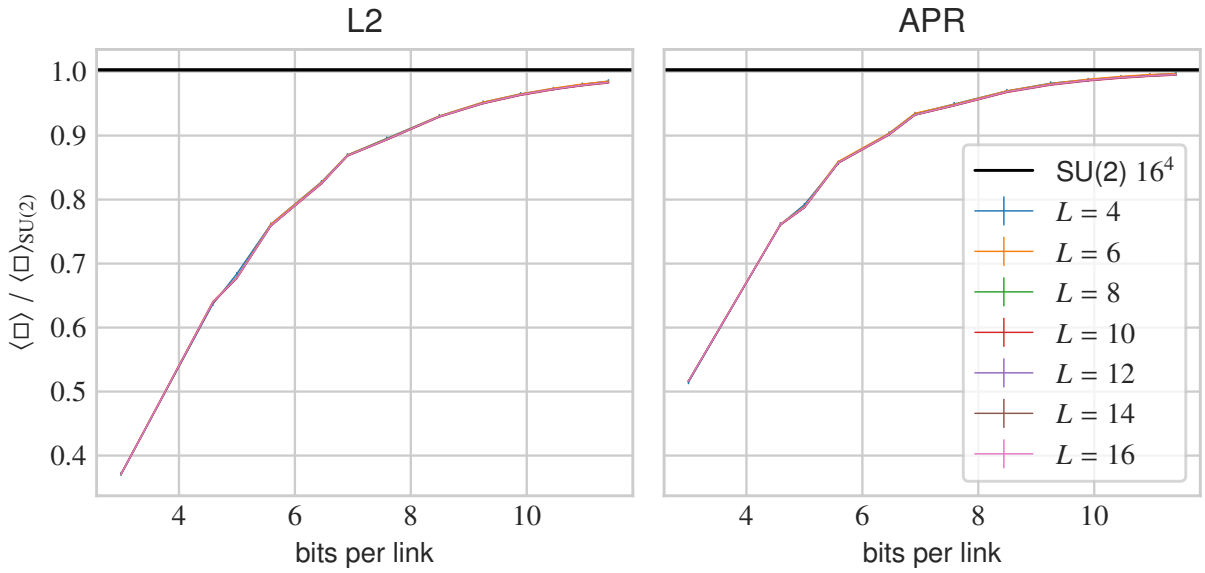

FIG. 5. The plaquette expectation value (defined in Sec. III A) as a function of bits-per-link for different lattice volumes, normalized to the undigitized value for 16^4. These results are computed on ensembles of 100 configurations with $V = L^4$ and $\beta = 2$ (except for the $V = 12^4$, which has 1000 configurations). Refer to Fig. 3 for more details.

scale, indicating that the volume dependence of the error is much smaller than the mesh size dependence (for the range plotted). We see similar volume independence in the three perimeter-six Wilson loops. Although we have not seen any significant volume dependence, this result may be observable-specific as the plaquette and perimeter-six Wilson loops are short distance quantities which only require a small volume in order to saturate.

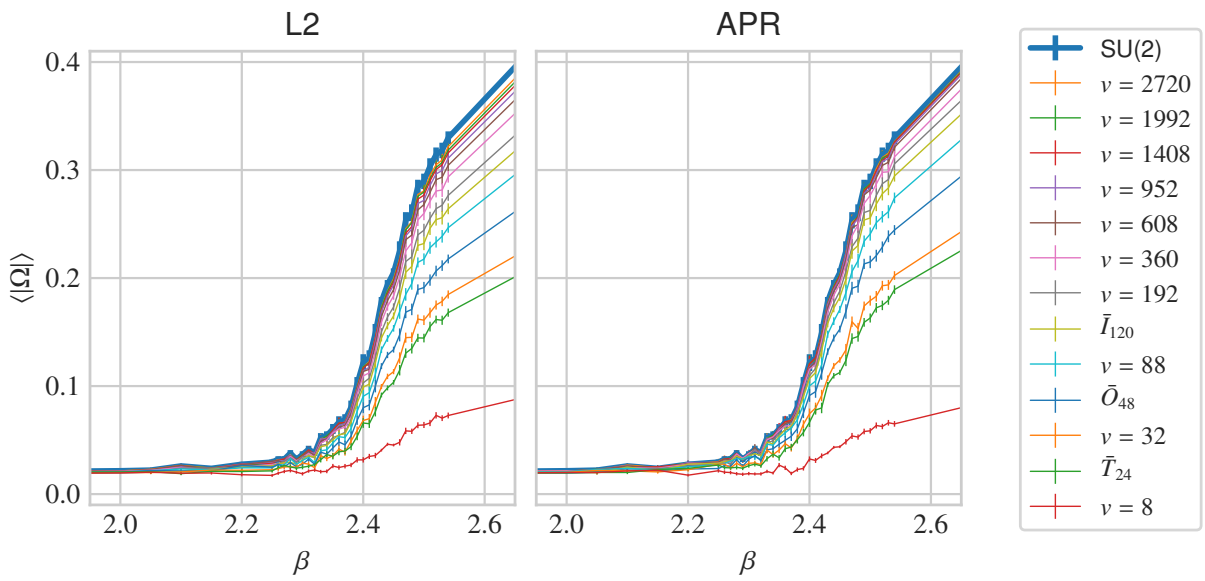

FIG. 6. The expectation value of the absolute value of the Polyakov loop (defined in Eqn. (7)) as a function of bare gauge coupling $\beta = 2N/g^2$ for different mesh-based digitization schemes. These results are computed on ensembles of 100 configurations with $V = 12^3 \times 6$ and with β values interpolated across the deconfinement transition. The data points are indicated by the presence of error bars, and interpolated between for the readers convenience. Refer to Fig. 3 for more details.

In Fig. 6, we examine how different mesh digitizations affect the Polyakov loop expectation value as a function of the bare gauge coupling $\beta = 2N/g^2$. As with the other observables, digitization and projection suppresses the Polyakov loop expectation value at all β values. Figure 7 plots the relative systematic error induced by the different digitization schemes for the data shown in Fig. 6. The Polyakov loop is close to zero in the confined phase, and so we see predominantly noise at lower βs. However, in the deconfined phase, the curves in Fig. 7 appear to be flat, indicating that the effect of the digitization for each β value is simply an overall multiplication by a constant smaller than one. Figure 8 shows the relative error averaged over the range $2.4 \leq \beta \leq 2.6$ as a function of bits-per-link, making it clear that this multiplicative constant approaches zero as the bits-per-link are reduced, again consistent with our arguments in Sec. III B. Figure 8 also shows convergence to the undigitized result explicitly. Projection with APR produces less error than with the L2 norm and appears to converge to the undigitized value quicker, but our data are unable to determine whether any systematic error survives in the limit of large bits-per-link for either scheme. We see no error due to digitization and projection in the critical value of β where the system deconfines, a positive indication as projection should not change the phase dynamics.

Finally, we turn to the static potential. Fig. 9 shows the static potential $aV(r)$ as a function of distance r/a, computed in the usual lattice QCD ultrafine digitization. Due to the large lattice spacing of this ensemble (i.e., the strong bare coupling), the potential is dominantly linear in all distance scales in our simulation. Above $r/a \approx 6$, the data become unreliable due to the exponentially decreasing signal in the Wilson loop as shown in Eq. (5). We restrict our subsequent discussion and figures to the region $r/a \lesssim 6$.

Fig. 10 shows the digitization error in the static potential as a function of distance. The most interesting feature of this figure is the

distance dependence. For any given mesh size, within our statistical precision, the error induced by digitizing $V(r)$ decreases with distance until saturating around $r/a \approx 3$ where our statistical error becomes appreciable. It is also worth noting that the static potential gets larger as the bits-per-link gets smaller, a consequence of the expectation values of Wilson loops approaching zero for projections to coarser digitizations. The APR projection outperforms the L2 projection at short distances. At longer distances $r/a \gtrsim 3$ the situation is not as clear: APR appears to perform slightly better, but this is not statistically significant for finer meshes.

Fig. 11 gives the digitization error in the static potential as a function of bits-per-link. There are several interesting features of this data. For a fixed number of bits-per-link, the systematic error from digitizing the static potential is much smaller than the systematic error in the plaquette, Wilson loops, or Polyakov loop. For example, with six bits-per-link the systematic error in the static potential at $r/a = 2.0$ can be as low as 6%, while the other quantities have at least 15% error. As expected, the digitized potential converges to the usual lattice QCD ultrafine-digitized result as the mesh becomes sufficiently large, with as few as 10 bits-per-link being indistinguishable from the ultrafine digitization. As expected from Fig. 10, we observe less systematic error at larger distances.

Figures 10 and 11 convey that long-distance physics is, evidently, less sensitive to digitization. In Sec. III B we argue that projecting to a coarse digitization is roughly akin to adding uncorrelated random noise to all the gauge links. The large effect at short length scales, approaching the Coulombic region of the potential, is consistent with what we have already observed in the plaquette and perimeter-six Wilson loops. At longer distances, we see a much smaller effect. It appears that correlations over longer length scales are less susceptible to the addition of incoherent noise, consistent with our arguments in Sec. III B. The statistically insignificant difference between APR and L2 projections at long distances suggests that projection is no longer the dominant source of error at long distances. If this is the case, then the error purely due to digitization is already at the sub-percent level at 9 bits-per-link.

In our finite-temperature data, we find that the induced error does not depend significantly on β. However, the coupling dependence of digitization effects could be qualitatively different for zero-temperature data. To check this, we use an ensemble with $\beta = 3$ and $V = 16^4$, and repeat our examination of the effects of digitization and projection on the plaquette, the perimeter six Wilson loops, and the static potential. As shown in Appendix A, the results indicate that there is no significant β dependence for zero-temperature data either.

V. DISCUSSION AND CONCLUSIONS

In this work we have empirically quantified the systematic error associated with digitizing the SU(2) gauge group in different ways using lattice gauge theory on classical computers. Section IV contains the main results of this work. Figs. 3, 4, 8, and 11 show the relative systematic error in the plaquette, the perimeter-six Wilson loops, the Polyakov loop, and the static potential as a function of the (qu)bit requirements, for each digitization and projection scheme, in units of bits-per-link. Across all observables, we observe several behaviors consistently. Projection to a coarser digitization suppresses the values of Wilson loops, consistent with our arguments in Section III B. For a given mesh, action-preserving rounding induces less error and asymptotes more quickly to the undigitized result than L2 norm projection. Projecting to a subgroup mesh appears to induce less error than projecting to a geodesic mesh of similar size, but the finer geodesic

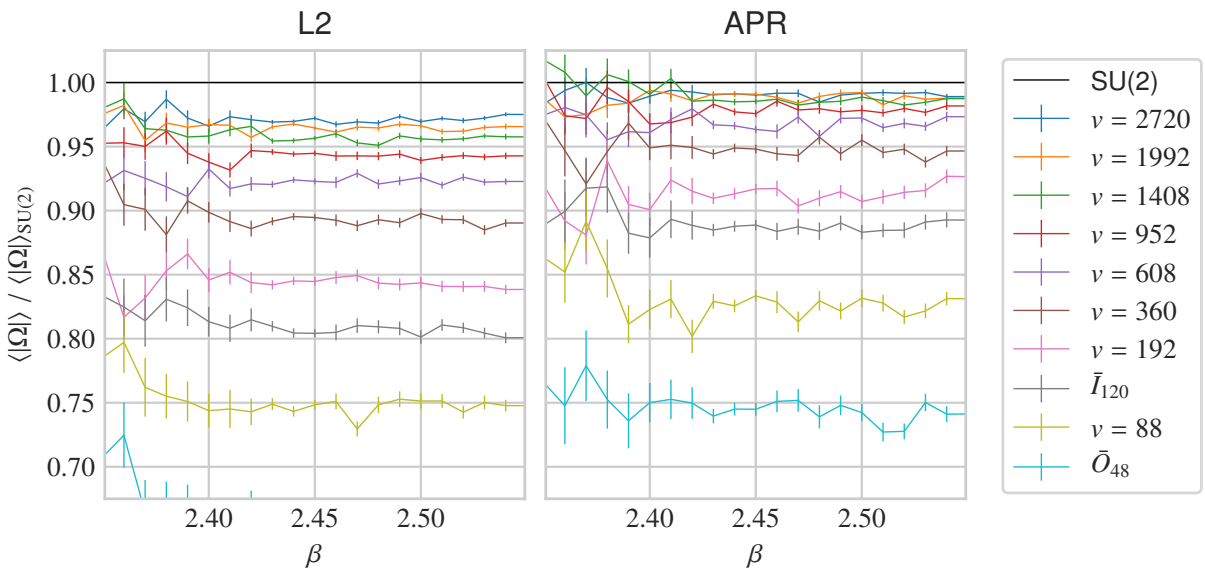

FIG. 7. The relative systematic error from digitizing the absolute value of the Polyakov loop as a function of $\beta = 2N/g^2$ for different mesh-based digitization schemes. Note that the denominator of the y-axis is the undigitized value for each ensemble. This data is the same as in Fig. 6. Refer to Fig. 6 for further details.

meshes outperform the largest discrete subgroup of SU(2). Finally, we also observe that long-distance physics is less sensitive to digitization and projection than short-distance physics.

Taken together, our results indicate that O(10) bits per link suffice to capture the essential physics of SU(2) gauge theory. Compared to modern lattice simulations, which default to 512 bits-per-link using double-precision floating-point numbers, this is an improvement of nearly two orders of magnitude. This observation has important implications for the types of physics that will be accessible to NISQ era quantum computers.

Moreover, the digitization and projection schemes discussed above amounts to forms of lossy compression for gauge links. The digitization error shown in this work is smaller than other dominant sources of error in many current classical lattice gauge theory calculations, and mesh-based schemes offer more than an order of magnitude compression over the commonly-used floating point representations. Thus, this work may have applications in modern classical lattice calculations if link compression can be used to overcome bandwidth bottlenecks.

We found that mesh-based schemes dramatically outperform the truncated fixed-point scheme, achieving similar accuracies with less than half the bits-per-link. Some of this may be attributed to the wastefulness of the fixed-point representation as discussed in Sec. II B, but this only accounts for a single bit of the difference. More importantly, each fixed-point representation matrix is only unitary to p bits of precision, and unitarity is a key building block of gauge theories. This emphasizes an important lesson: different digitization schemes are possible, but standard principles of quantum physics remain a guiding light for constructing optimal digitizations as new technology is explored.[8]

[8] It was also necessary to enforce gauge invariance and locality when applying machine learning techniques to lattice QCD in order to produce results [27].

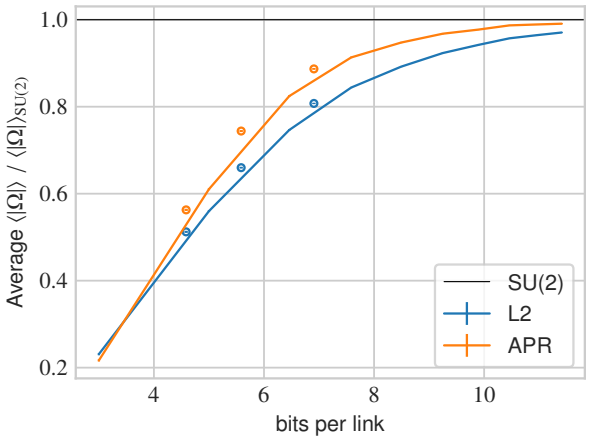

FIG. 8. The relative systematic error from digitizing the absolute value of the Polyakov loop as plotted in Fig. 7, here averaged over β in the range from 2.4 to 2.6, as a function of bits-per-link. The curves are results for geodesic meshes, while points are results for discrete subgroups.

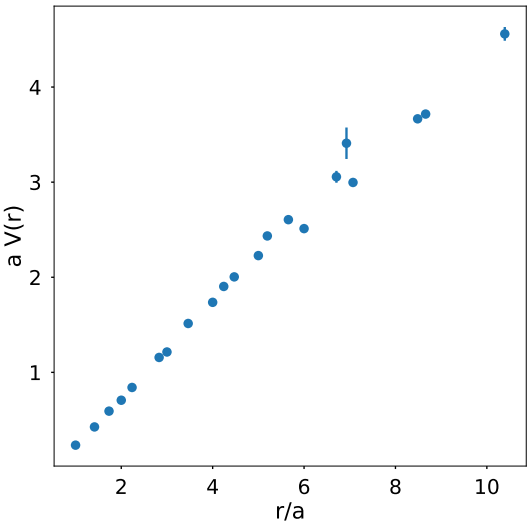

FIG. 9. The lattice static potential (defined in Eq. (5)) as a function of distance in lattice units. The data in this plot was generated using an ensemble of 1000 configurations with $V = 12^4$ and $\beta = 2.0$. We refer readers to the text for further details.

If gauge links are represented in an indexed mesh representation, multiplication must be implemented using lookup tables. There is existing work which considers how to do this efficiently on classical computers [10, 20]. An important question for future research is whether this can be easily and efficiently implemented on a quantum computer.

In this study, we examined SU(2) pure gauge theory. However, we are obviously more interested in QCD, whose gauge group is SU(3). Repeating this study for SU(3) pure gauge theory and for theories with dynamical fermions are obvious next steps. We note that previous work which used indexed mesh digitizations to simulate SU(3) pure gauge theory found that a mesh with 1080 elements was too coarse to avoid artifact phases near interesting values of β, but a mesh with 38880 elements was viable [9]. Translating these numbers to our bits-per-link metric, this suggests that ~ 10 bits is insufficient to represent an SU(3) gauge link, but ~ 15 bits may be enough.

As noted above, we do not generate coarsely digitized ensembles. Systematically studying the error induced when simulating using a coarse digitization is an interesting and complementary direction for future work. In this study, we generate ensembles in the standard lattice ultrafine digitization, then project to coarser digitizations. This introduces error specific to projection, which is difficult to disambiguate from error due to digitization alone. Generating data directly in a coarse digitization does not require projection, and thus would allow estimation of digitization error without this confounding factor. However, simulating with coarse digitizations would introduce new errors that our method is immune to, and it may be that these errors are large (cf. the lattice artifact phases seen by Refs. [7–9]). Thus, coarsely digitized simulation and our approach of projection can provide independent probes of digitization error.

Our results are already very promising, but it is likely possible to reduce (qu)bit

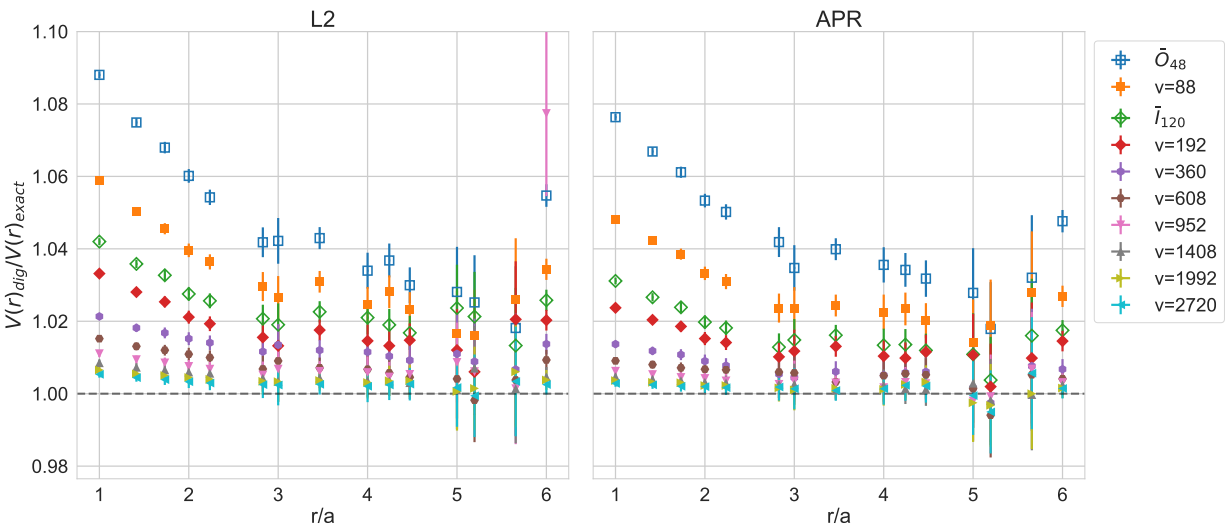

FIG. 10. The digitization error in the static potential $V(r)$ as a function of distance. Filled symbols denote edgewise meshes, while open symbols denote discrete subgroups of SU(2). The left pane shows the L2 projection, while the right pane shows the APR projection.

requirements even further. One possible method is by using better meshes and projection schemes. We observed that projecting to discrete subgroups of SU(2) induced less error than projecting to geodesic meshes. This suggests that meshes made by interpolating between points in discrete subgroups (as proposed in Ref. [20]) might perform better than geodesic meshes. Our action-preserving rounding method for projecting onto meshes performs substantially better than projection using the L2 norm, even on fine meshes where one might expect the projection method to matter less. Pushing further and developing more sophisticated action-preserving or gauge-invariant projection schemes would be an interesting direction for future work. This is a particularly interesting direction for further thought when considered in the context of using lattice simulations on classical computers to prepare states for quantum computers, and for gauge link compression for high-performance computing.

Another way of reducing (qu)bit resource requirements is by removing non-physical degrees of freedom from the theory. The authors of [28] show that when using an eigenstate truncation method the vast majority of states in the naive finite-dimensional Hilbert space are unphysical. There they explore how to construct the theory on the physical subspace alone. It would be worthwhile to explore if such techniques could also be applied to mesh digitizations. Similarly, implementing a digitization scheme which uses gauge-fixing could also reduce (qu)bit resource requirements. Exploring the mapping between the eigenstate truncation method described in Refs. [11–13] and the group-value digitization scheme explored in this work may provide further insights.

Our work is particularly applicable to the quantum link formulation of SU(N) gauge theories (with matter fields) on quantum computers [29, 30]. In these formulations, the SU(N) link matrices build the quantum-link Hamiltonian. Our results for four-dimensional pure gauge SU(2) indicate that only a small number of (qu)bits per link may be needed to achieve accurate results in these theories. If QCD is formulated on a quantum computer in this way, future work is needed to empirically quantify the (qu)bit requirements of these SU(N) implementations.

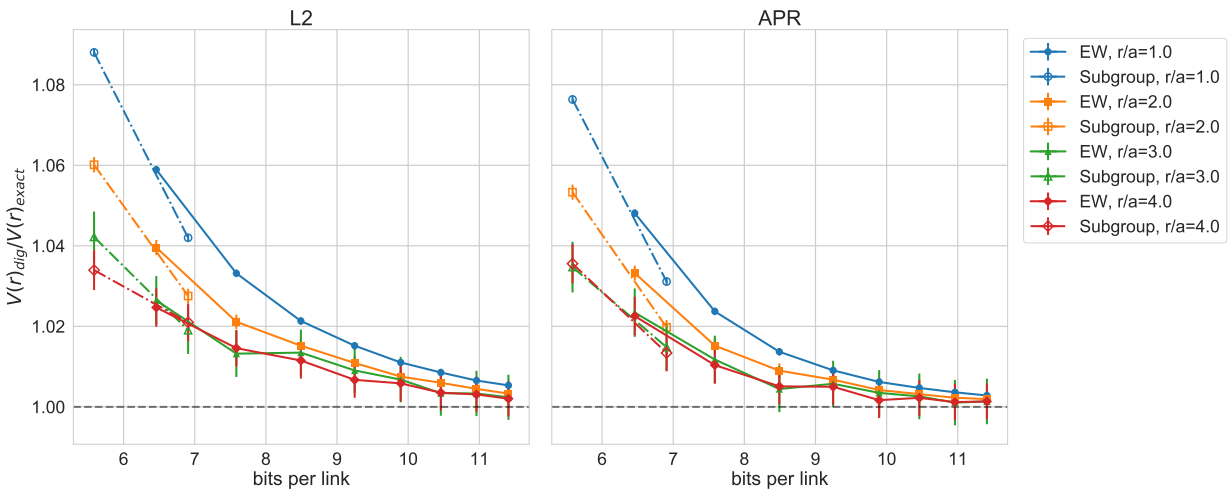

FIG. 11. The relative error from digitizing the static potential (defined in Eqn. (5)) as a function of bits-per-link. Each line in this figure corresponds to a different fixed value of the distance r/a. We refer the reader to Fig. 9 for further details on the ensemble data used to generate this plot. Fig. 3 contains additional details on terminology.

ACKNOWLEDGMENTS

We would like to thank Tom DeGrand and Andreas Kronfeld for useful discussions. We are also grateful to the MILC collaboration for the use of the source code adapted to generate the ensembles in this study. This work was supported in part by the U. S. Department of Energy under contract DE-SC0010005 (DH and EN). This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U. S. Department of Energy, Office of Science, Office of High Energy Physics. This work was partially supported by a HEP-QIS QuantISED award titled "Quantum Information Science for Applied Quantum Field Theory". Brookhaven National Laboratory is supported by the U. S. Department of Energy under contract no. DE-SC0012704. This work was partially supported by a HEP-QIS QuantISED award titled "Quantum Information Science for Applied Quantum Field Theory". CH and KH would like to thank the University of Colorado Boulder for hosting each, which directly led to this work.

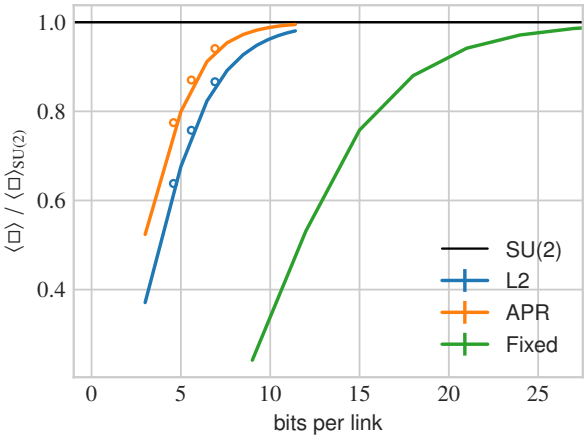

FIG. 12. Caption identical to Fig. 3 except that these results are computed on an ensemble of 100 configurations with $V = 16^4$ and $\beta = 3$.

Appendix A: Digitization at $\beta = 3$

In this appendix we present the results of digitizing a weaker coupling ensemble (compared to the $\beta = 2$ ensembles in Sec. IV) which contains 100 configurations with $\beta = 3$ and $V = 16^4$. Digitizing the plaquette with $\beta = 3$ is shown in Fig. 12. In Fig. 13, the digitization of the perimeter six Wilson loops is given. Figs. 14−16 show how digitization

effects the $\beta = 3$ static potential. Unlike in Fig. 9, the Coulombic part of the static potential is clearly visible in Fig. 14, indicating that the lattice spacing is shorter for this ensemble. All plots are qualitatively and quantitatively similar to those shown in Sec. IV with $\beta = 2$.

[1] A. S. Kronfeld, Ann. Rev. Nucl. Part. Sci. **62**, 265 (2012), arXiv:1203.1204 [hep-lat].

[2] M. J. Savage, *Proceedings, 34th International Symposium on Lattice Field Theory (Lattice 2016): Southampton, UK, July 24-30, 2016*, PoS **LATTICE2016**, 021 (2016), arXiv:1611.02078 [hep-lat].

[3] S. R. Beane, W. Detmold, K. Orginos, and M. J. Savage, Prog. Part. Nucl. Phys. **66**, 1 (2011), arXiv:1004.2935 [hep-lat].

[4] J. Preskill, arXiv:1801.00862 [quant-ph].

[5] C. Urbach, Comput. Phys. Commun. **224**, 44 (2018), arXiv:1710.07526 [hep-lat].

[6] M. Creutz, L. Jacobs, and C. Rebbi, Phys. Rept. **95**, 201 (1983).

[7] C. Rebbi, Phys. Rev. **D21**, 3350 (1980), [,210(1979)].

[8] D. Petcher and D. H. Weingarten, Phys. Rev. **D22**, 2465 (1980).

[9] G. Bhanot and C. Rebbi, Phys. Rev. **D24**, 3319 (1981).

[10] P. Lisboa and C. Michael, Phys. Lett. **113B**, 303 (1982).

[11] N. Klco, E. F. Dumitrescu, A. J. McCaskey, T. D. Morris, R. C. Pooser, M. Sanz, E. Solano, P. Lougovski, and M. J. Savage, Phys. Rev. **A98**, 032331 (2018), arXiv:1803.03326 [quant-ph].

[12] H.-H. Lu *et al.*, (2018), arXiv:1810.03959 [quant-ph].

[13] E. Zohar, J. I. Cirac, and B. Reznik, Phys. Rev. **A88**, 023617 (2013), arXiv:1303.5040 [quant-ph].

[14] D. Banerjee, M. Bögli, M. Dalmonte, E. Rico, P. Stebler, U.-J. Wiese, and P. Zoller, Phys. Rev. Lett. **110**, 125303 (2013).

[15] P. Silvi, E. Rico, M. Dalmonte, F. Tschirsich, and S. Montangero, Quantum **1**, 9 (2017).

[16] E. Zohar, T. B. Wahl, M. Burrello, and J. I. Cirac, Annals Phys. **374**, 84 (2016), arXiv:1607.08115 [quant-ph].

[17] M. C. Bañuls, K. Cichy, J. I. Cirac, K. Jansen, and S. Kühn, Phys. Rev. X **7**, 041046 (2017).

[18] M. A. Clark, R. Babich, K. Barros, R. C. Brower, and C. Rebbi, Comput. Phys. Commun. **181**, 1517 (2010), arXiv:0911.3191 [hep-lat].

[19] J. P. Nolan, "The `mvmesh` package for R, v1.5," (2017).

[20] P. Lisboa and C. Michael, Nucl. Phys. **B210**, 15 (1982).

[21] C. Gattringer and C. B. Lang, Lect. Notes Phys. **788**, 1 (2010).

[22] T. DeGrand and C. E. Detar, *Lattice methods for quantum chromodynamics* (2006).

[23] MILC Collaboration, `http://www.physics.utah.edu/~detar/milc/`.

[24] K. G. Wilson, Phys. Rev. **D10**, 2445 (1974), [,319(1974)].

[25] M. D. Schwartz, *Quantum Field Theory and the Standard Model* (Cambridge University Press, 2014).

[26] B. Svetitsky and L. G. Yaffe, Nucl. Phys. **B210**, 423 (1982).

[27] P. E. Shanahan, D. Trewartha, and W. Detmold, Phys. Rev. **D97**, 094506 (2018), arXiv:1801.05784 [hep-lat].

[28] D. B. Kaplan and J. R. Stryker, (2018), arXiv:1806.08797 [hep-lat].

[29] R. Brower, S. Chandrasekharan, and U. J. Wiese, Phys. Rev. **D60**, 094502 (1999), arXiv:hep-th/9704106 [hep-th].

[30] E. Zohar and M. Burrello, Phys. Rev. **D91**, 054506 (2015), arXiv:1409.3085 [quant-ph].

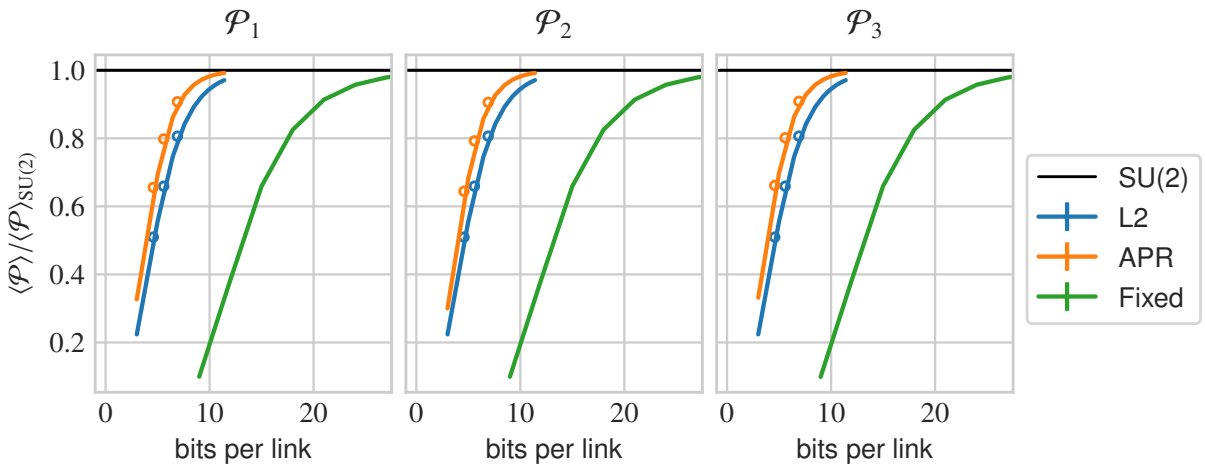

FIG. 13. Caption identical to Fig. 4 except that these results are computed on an ensemble of 100 configurations with $V = 16^4$ and $\beta = 3$.

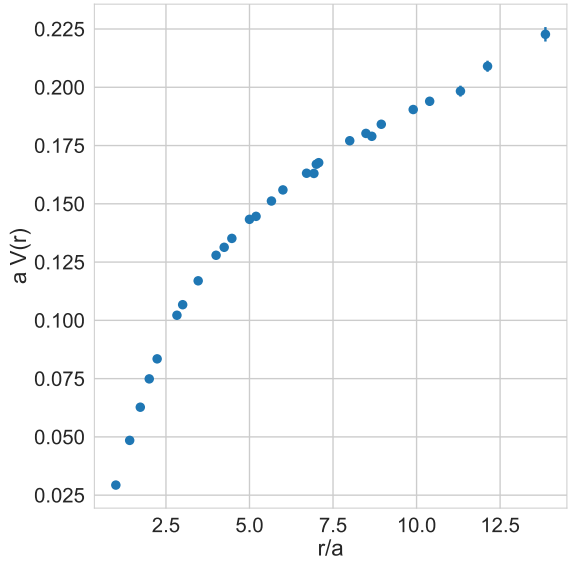

FIG. 14. Caption identical to Fig. 9 except that these results are computed on an ensemble of 100 configurations with $V = 16^4$ and $\beta = 3$.

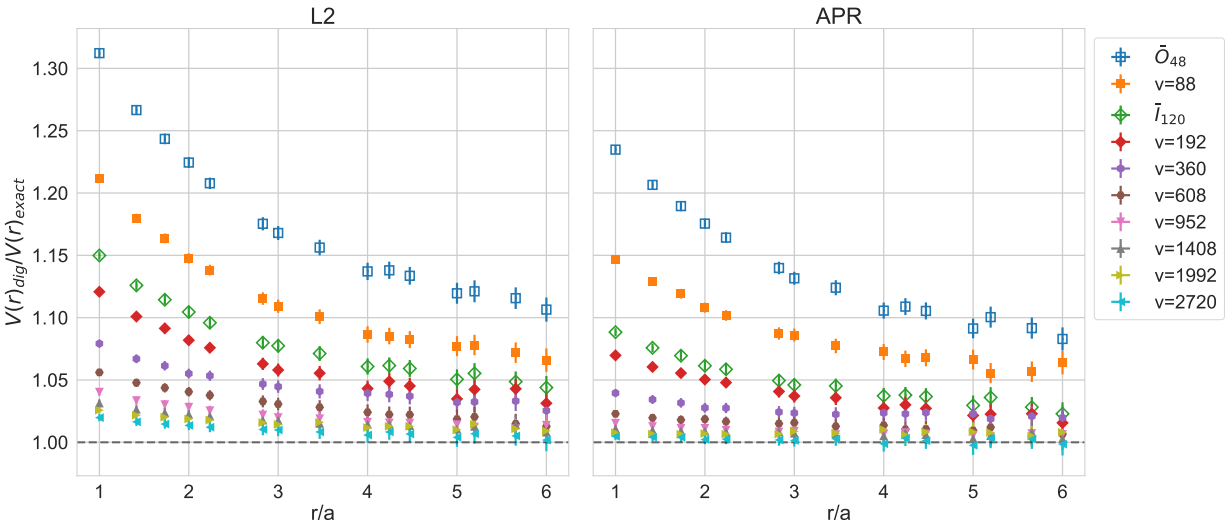

FIG. 15. Caption identical to Fig. 10 except that these results are computed on an ensemble of 100 configurations with $V = 16^4$ and $\beta = 3$.

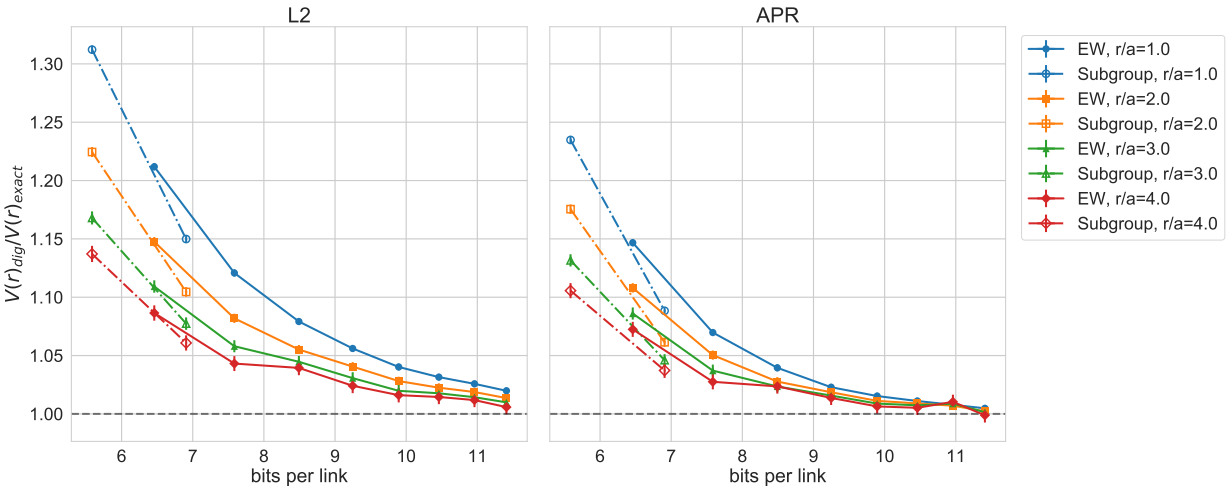

FIG. 16. Caption identical to Fig. 11 except that these results are computed on an ensemble of 100 configurations with $V = 16^4$ and $\beta = 3$.