


Zeno effect suppression of gauge drift in quantum simulationsCarter Ball^{*} and Thomas D. Cohen[†]*Department of Physics, University of Maryland, College Park, Maryland 20742, USA* (Received 31 May 2024; revised 19 July 2024; accepted 29 July 2024; published 12 August 2024)

Quantum simulation of lattice gauge theories is a promising tool for the study of many complicated problems including ones with real-time dynamics. For gauge theories, however, there is a major challenge in maintaining gauge invariance during time evolution. Such theories have a full Hilbert space that is larger than the physical space—the set of states which are gauge invariant or equivalently respect the Gauss law. While an exact implementation of Hamiltonian dynamics starting in the physical Hilbert space will keep the system in the physical space, various types of errors will inevitably produce components outside of it. This work proposes a method of suppressing this gauge drift via the Zeno effect. As in the standard picture of the Zeno effect, our method relies on frequent projection onto the physical subspace. Additionally, a technique is discussed to reduce the speed of the gauge drift, which helps to reduce the required frequency of projections. We demonstrate our method on a \mathbb{Z}_2 gauge theory toy model.

DOI: [10.1103/PhysRevA.110.022417](https://doi.org/10.1103/PhysRevA.110.022417)**I. INTRODUCTION**

Quantum computers promise to be vital tools for the simulation of many fundamental yet complicated problems [1–5] from a range of subfields, including particle physics, nuclear physics, and condensed matter. The simulation and study of complex systems often requires much more computing power than classical computers can provide [6,7]; thus, there is a present need to map these systems onto quantum devices. Analog quantum devices manufacture a system that can imitate, at least in some regime, the desired theory while digital quantum devices map a theory onto an array of qubits and control it via quantum gates [8]. Much research has gone into studying these quantum devices and how they can be used for quantum simulation [9].

Like many complicated theories, lattice gauge theories can, in principle, benefit enormously from efficient and reliable quantum simulation [10–25]. These theories typically place matter degrees of freedom on the sites of the lattice while gauge field operators are placed on the links; it is typical to work in a Hamiltonian formulation with discretized space but continuous time [26–28]. This is usually done in the temporal gauge [29] (which sets the time component of gauge fields to zero) and by taking the continuous time limit. Note that the temporal gauge does not fully exhaust the gauge freedom, leaving a residual gauge freedom for the spatial components. Transformations of this residual gauge are generated by Gauss law operators (in analogy to classical electrodynamics). Physical states are then defined as those that are invariant under all residual gauge transformations.

Much work has gone into the study of both abelian and nonabelian lattice gauge theories. Notably, maintaining

nonabelian gauge invariance is particularly difficult. When it comes to analog quantum simulation, many methods have been developed to address this gauge invariance problem, including constructing a system that makes gauge invariance a consequence of internal symmetry such as angular momentum [30,31]. Another analog simulation method is to impose an energy penalty on the unphysical states [16,32–40], often by adding a term to the Hamiltonian proportional to the square of the Gauss' law operators. Then, the lowest-energy levels of the system are the physical ones, so keeping the system in the low-energy regime results, to good approximation, in keeping the system within the physical subspace of the much larger Hilbert space. Variations on this method have been developed, including one where the addition to the Hamiltonian is directly proportional to the Gauss' law operators [41]. Another such variation on the energy penalty method uses local pseudogenerators, which exactly match the Gauss' law operators in the relevant physical subspace but not in unphysical sectors, allowing these operators to be much less complex and thus cheaper to engineer in practice [42–44]. Interestingly, for sufficiently large energy penalties, the protection term added to the Hamiltonian, constructed with local pseudogenerators, can be seen as a strong projector which constrain the dynamics of the system to quantum Zeno subspaces within a given timescale [44,45].

The method presented in this paper concerns digital quantum simulation. Ideally, any method to simulate a nonabelian gauge theory of interest, such as quantum chromodynamics (QCD), would only simulate the gauge invariant (i.e., physical) states. Methods of this kind have been proposed which solve the Gauss law, either by eliminating the gauge fields [20,46,47] or the matter fields [48–52]. Another method used dual formulations of the gauge theories to enforce the Gauss law [53–55]. While the methods discussed above did not have to deal with unphysical states, their physical implementation may be more challenging.

*Contact author: cball12@umd.edu†Contact author: cohen@umd.edu

Thus, many methods simulate all states in the Hilbert space of the system and must contend with keeping the system within the exponentially smaller physical subspace. Crucially, work has been done to develop methods of time evolution that respect gauge invariance [13,56]. Thus, ideally time evolving an initial state that is constructed to be physical would keep the system within the physical subspace. However, quantum noise, gate errors, and various errors from approximations can lead to the system picking up contributions from unphysical states. Previous work addressed this problem in a variety of ways. Perhaps the most straightforward was to simply accept the unphysical contributions on the premise that these contributions were reduced as the time step was decreased [15]. Another method was dynamical decoupling [57], a concept from quantum control theory where local symmetries were maintained via periodic driving. Also, oracles have been developed [58] to check for violations of Gauss' law for abelian lattice gauge theories, with similar oracles for nonabelian theories under development.

Furthermore, a method of engineering classical noise to constrain the system to the physical subspace [59] has been proposed. This method served as a classical analog to the quantum Zeno effect by using classical fluctuations of a perturbing field to dissipate unwanted evolution into unphysical regions of the Hilbert space. The method of this paper, in contrast, rests on the standard quantum Zeno effect [60], which uses frequent projections to keep the system within a desired space; for our purposes the desired space is, of course, the physical subspace of the system's Hilbert space.

This paper is organized as follows. Section II outlines our method, which takes inspiration from the rodeo algorithm [61] (a method of projection) and a technique of unitary gauge drift suppression. Section III demonstrates our method on a toy model of simulating pure \mathbb{Z}_2 gauge theory on a 2×2 lattice with periodic boundary conditions. Section IV goes into more detail about the implementation of our method, and Sec. V provides concluding thoughts and discusses the required future work.

II. METHOD

The method exploits the quantum Zeno effect to maintain a close approximation of gauge invariance. For a nonabelian theory, the gauge invariant, or physical, states are given by those states $|\psi\rangle$ that satisfy

$$G_x^a |\psi\rangle = 0 \quad (1)$$

for all a and x , where G_x^a is the a th Gauss' law operator at site x . Note that for compact Lie gauge groups [e.g., $SU(N)$], these G_x^a operators generate local gauge transformations.

To leverage the Zeno effect, we must invoke a method of projection onto the physical subspace. Our method adapts previous work on the rodeo algorithm [61–64]. We also incorporate a method of suppressing coherent gauge drift, discussed below.

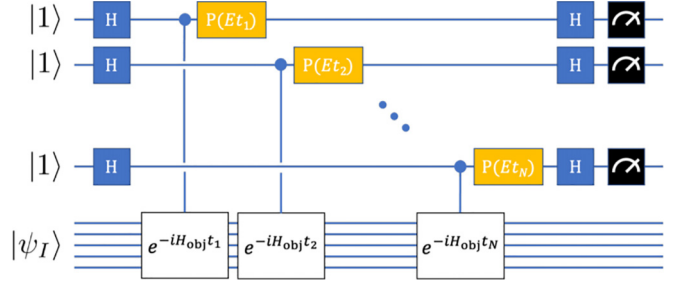


FIG. 1. The circuit diagram for the rodeo algorithm [61].

A. Projection via the rodeo algorithm

The rodeo algorithm was developed to prepare eigenvectors of a given Hamiltonian. See Fig. 1 for the circuit diagram of the algorithm [61].

The rodeo algorithm aims to construct, to good approximation, an eigenstate of a system with a chosen energy by “shaking off” amplitudes of other, unwanted eigenstates, like a bull shaking off its rider at a rodeo. The algorithm accomplishes this task with ancillary qubits as shown in the circuit diagram of Fig. 1. To see how the rodeo algorithm constructs the desired eigenstate, we will write out how it works for one ancillary qubit. Starting with the system in the initial state $|\psi_I\rangle$ and the ancillary qubit initialized to $|1\rangle$, we have

$$\begin{aligned} & \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & Ie^{iEt} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & e^{-iH_{obj}t} \end{bmatrix} \\ & \times \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 0 \\ |\psi_I\rangle \end{bmatrix} \quad (2) \end{aligned}$$

$$\begin{aligned} & = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & Ie^{iEt} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & e^{-iH_{obj}t} \end{bmatrix} \\ & \times \begin{bmatrix} \frac{1}{\sqrt{2}} |\psi_I\rangle \\ -\frac{1}{\sqrt{2}} |\psi_I\rangle \end{bmatrix} \quad (3) \end{aligned}$$

$$\begin{aligned} & = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} |\psi_I\rangle \\ -\frac{1}{\sqrt{2}} e^{-i(H_{obj}-E)t} |\psi_I\rangle \end{bmatrix} \quad (4) \end{aligned}$$

$$\begin{aligned} & = \begin{bmatrix} \frac{1}{2}(I - e^{-i(H_{obj}-E)t}) |\psi_I\rangle \\ \frac{1}{2}(I + e^{-i(H_{obj}-E)t}) |\psi_I\rangle \end{bmatrix}, \quad (5) \end{aligned}$$

where H_{obj} is the Hamiltonian of the system of interest (i.e., the object system) and E is the desired energy of the constructed state. Equation (2) writes out the actions of (reading from right to left) the Hadamard, controlled time evolution, phase shift, and Hadamard gates, respectively, on the initial state. Note that the first Hadamard gate gives equal amplitude, up to a sign, to the up (i.e., $|0\rangle$) and down (i.e., $|1\rangle$) components of the ancillary qubit [Eq. (3)], then the controlled time evolution and phase shift gates give a phase only to the down component [Eq. (4)], and finally the last Hadamard gate causes interference between the up and down components [Eq. (5)].

The final step in the rodeo algorithm is to measure the ancillary qubit(s). A successful implementation of the rodeo

algorithm measures every ancillary qubit in the state $|1\rangle$ at the end. In the single ancillary qubit example above, this results in the transformation

$$|\psi_I\rangle \rightarrow \frac{1}{2}(I + e^{-i(H_{\text{obj}}-E)t})|\psi_I\rangle. \quad (6)$$

To see the benefits of this transformation, we decompose our initial state into energy eigenstates $|\psi_I\rangle = \sum_n c_n |n\rangle$ where $H_{\text{obj}}|n\rangle = E_n|n\rangle$. The amplitude of each energy eigenstate then transforms as

$$c_n \rightarrow \frac{1}{2}c_n(1 + e^{-i(E_n-E)t}), \quad (7)$$

which means the probability of each energy eigenstate transforms as

$$|c_n|^2 \rightarrow |c_n|^2 \frac{1}{2}(1 + e^{-i(E_n-E)t})^2 \quad (8)$$

$$\rightarrow |c_n|^2 \cos^2\left((E_n - E)\frac{t}{2}\right). \quad (9)$$

Thus, the probability for the system to be in a particular energy eigenstate E_n is multiplied by a factor $\cos^2((E_n - E)\frac{t}{2})$, which is notably 1 if $E = E_n$ and less than 1 otherwise. This is how the rodeo algorithm works: it constructs an approximate version of the state of energy E by suppressing all other states with a $\cos^2(\frac{1}{2}t\Delta E)$ factor. Stronger suppression of contributions from unwanted energy eigenstates can be accomplished with multiple rounds of this algorithm, as discussed in Refs. [61–63]. Note that to implement a round of the rodeo algorithm, a time t must be chosen. If the times are chosen strategically for the various rounds, remarkably strong suppression can be achieved with very modest resources [64].

We note here that the rodeo algorithm will fail to project onto the state of interest with finite probability (i.e., if any ancillary qubit is not measured in $|1\rangle$), where failure means that the resulting state is orthogonal to the target state. Crucially, we note that the probability of a successful projection is proportional to the overlap of the initial state of the system with the target state of the projection. As we plan to use this projection to leverage the Zeno effect, which means making sufficiently many measurements so that the overlap with the target subspace is close to unity, the rate of failure can be made negligibly small. The chance of failure serves as the probabilistically defined time span for which our method works; for example, if the probability for a failed projection is one in a million, we would expect one failure within one million successive projections.

So far, we focused on the rodeo algorithm due to its key feature of leaving desired amplitudes (i.e., the amplitude of the desired energy eigenstate) untouched while suppressing all other amplitudes. Here, we note that there is no reason this algorithm works only for the construction of the eigenstates of the Hamiltonian. Thus, going forward, we will refer to rodeo projections as any projection of the form expressed in Fig. 1 with some hermitian operator in place of the Hamiltonian.

Recall that our goal is to use frequent projections onto the physical subspace to maintain gauge invariance of the system via the Zeno effect. With this in mind, we now discuss the form of a rodeo projection that we will use to project onto the physical subspace. We replace the Hamiltonian with a new

hermitian operator that we will call G^2 :

$$G^2 = \sum_{a,x} (G_x^a)^2. \quad (10)$$

From the above definition of the Gauss' law operators, we see that all physical states are eigenstates of G^2 with eigenvalue 0. Furthermore, all the other eigenstates will be unphysical and with eigenvalues greater than 0 since G^2 is positive definite by construction. It is this spectrum of G^2 that we will exploit when using rodeo projections: instead of constructing some eigenstate of a given Hamiltonian, we will construct a physical state by constructing a ground state of G^2 via rodeo projections. This means replacing H_{obj} in Fig. 1 with G^2 ; furthermore, as the eigenvalue of the desired eigenstate (i.e., a physical state) is 0, we will remove the phase shift gates since the required phase shift is $0 * t = 0$.

This is the core of the method outlined in this paper: a rodeo projection with this G^2 operator will leave all physical amplitudes untouched while suppressing all unphysical amplitudes. Doing this projection frequently enough will keep the system in a close approximation of a gauge invariant state via the Zeno effect.

There is one minor detail that we must clarify before moving on. The effectiveness of this projection is helped if the spectrum of the operator, in our case G^2 , is discrete. That way there will be less contamination from the eigenstates nearest to the target eigenstate. Luckily, the spectrum of G^2 for our principal case of interest SU(3) gauge theories is, in fact, discrete. This can be seen very straightforwardly using basic quantum mechanics, though we will point out one small detail: the basic argument implicitly relies on the fact that all states have nonnegative norms. While this is obviously true for physical states it is not necessarily true for the full Hilbert space of QCD. However, we are working within a Hamiltonian formulation and using the temporal gauge, which is a ghostless gauge [65], meaning that we do, in fact, have nonnegative norms for all states. Thus, all components G^a have discrete spectra, so G^2 has a discrete spectrum as well. This means the ground state of G^2 is gapped, which is helpful when it comes to projection.

B. Suppressing coherent gauge drift

The basic idea of the Zeno effect is that if you project back to the desired space before straying too far out of it (i.e., frequently enough), then the system stays in the desired space with high probability. This is for a simple reason: projections depend on the probability that the system is in the desired state, which corresponds to the square of the magnitude of the amplitude. Thus, if one projects when the amplitude for the system drifting into an unwanted state is some small value ϵ , the probability of failure is ϵ^2 —a much smaller value.

Consider what this means in practice if one wishes to study a system's time evolution over a time, T , and gauge-invariance-violating errors in the amplitude accrue at a rate r . Suppose one makes N evenly spaced projections (which for the sake of this analysis are assumed to be perfect) during the time evolution. The probability of a failure—obtaining a state outside the physical space—at each measurement is $(rT/N)^2$, and the probability of a successful run for the full

time evolution (i.e., the probability of N successes and 0 failures) is given by

$$P_s = \left(1 - \frac{r^2 T^2}{N^2}\right)^N \approx 1 - \frac{r^2 T^2}{N}, \quad (11)$$

where the approximate equality holds in the limit of large N and $1 - P_s \ll 1$. The key point is that for $N \gtrsim r^2 T^2$ there is a substantial probability that the full time evolution can occur with the system restricted to the physical space. Note that this means that by making sufficient measurements, the evolution can stay in the physical space even if the evolution time is much larger than $1/r$. This is a major improvement.

Of course, if one makes a finite number of projections there is a finite probability of failure—obtaining a state outside the physical space. If that happens, the only recourse is to stop the calculation and begin again—an expensive prospect. Thus, there is an optimization problem in choosing N : if N is too large, there is the computational cost of making more projections than are necessary, whereas if N is too small than the failure rate is too high requiring the additional cost of starting over. The optimal value depends on the computation costs of the projection as well as the costs of running the simulation.

With this in mind, the method has been enlarged to include an additional way to suppress gauge drift, an approach developed in Ref. [66]. We show that a mixed approach using both quantum Zeno and the approach of Ref. [66] is likely to be more efficient than either one separately.

The method of Ref. [66] is essentially an alternative version of the energy penalty method for suppressing coherent gauge drift. Here, we define coherent gauge drift as a unitary process U_D that transforms a physical state $|\psi\rangle$ to

$$U_D |\psi\rangle = \sqrt{1 - \epsilon^2} |\psi\rangle + \epsilon |\omega\rangle, \quad (12)$$

where $|\omega\rangle$ is a nonphysical state and ϵ characterizes the strength of the drift. Recall that an energy penalty method constructs a Hamiltonian H_G with the behavior that $H_G |\psi\rangle = 0$ for physical states $|\psi\rangle$ and $H_G |\omega\rangle > 0$ for unphysical states $|\omega\rangle$. Adding $\lambda_G H_G$ for $\lambda_G \gg 0$ to the Hamiltonian of the system suppresses transitions to unphysical states via an energy penalty. The authors of Ref. [66] noted that, in principle, building H_G out of the fundamental fields of the lattice gauge theory would be quite difficult for a nonabelian theory; however, quantum simulation via Trotterized time steps depended upon $e^{-iH_G t}$ not solely H_G . Furthermore, they pointed out that this operator basically performs a random gauge transformation on the state of the system, which leaves physical states unchanged while unphysical states take on a phase. Thus, this approach dispenses with the difficult task of building H_G and instead simply implements the principal effect of including H_G via a random gauge transformation. In summary, the method of Ref. [66] is to implement, after every Trotterization time step, a random gauge transformation.

Our method aims to leverage this method wholesale: frequent projections back to the physical subspace is expensive, so we include a random gauge transformation after each Trotterization time step. This reduces the gauge drift per time step, allowing us to do projections to the physical subspace less frequently while still staying close enough to the

physical subspace to leverage the quantum Zeno effect, which allows the suppression of whatever gauge drift that survives the random gauge transformations.

C. Calibration

Our method, as outlined above, is to conduct a random gauge transformation after every Trotterization time step as well as to conduct projections in the form discussed above to project back to the physical subspace. One important detail that needs to be specified is the frequency of projections. To this end, we include a calibration phase in the method—a set of initial studies—to best estimate the frequency of projections that constitutes a useful compromise between the minimization of gauge drift and the minimization of the number of projections. We note up front that, in principle, doing a projection after every time step would be most effective in terms of reducing gauge drift as the probability of failure of the projection is very low. The problem with this is simply that it is extremely expensive computationally. In the next section we will outline results from a toy model that show that other, less expensive forms of the method work well. This suggests that realistic calculations can balance the resource costs with the requirements on effective gauge drift suppression.

We propose an approach that should be useful in situations when a simulation needs to be done repeatedly—in that case, a simple calibration phase to optimize the calculation is done at the outset: start with an initial trial frequency of projections based on whatever information and intuition one has about the system. Here, any previous knowledge about the system being simulated should be exploited, especially concerning the speed of gauge drift. With an initial frequency chosen, the time evolution of the system is run a small number of times with the chosen trial frequency of projections to see how long before the projection step fails. If it appears that failure will in general occur before the desired length of time of the simulation, the frequency of projections can be increased. This process can be repeated until an acceptable balance is found.

Behind this simple calibration phase is the fact that the average gauge drift is linearly proportional to the number of time steps between projections. Intuitively, this also means the probability of a failed projection is linearly proportional to the number of time steps between projections. With these relations in mind, it is simple to tune the frequency of projections given a relatively few trial runs.

III. PERFORMANCE

To study our algorithm, we consider a small “toy” model. It is a pure \mathbb{Z}_2 gauge theory in two spatial dimensions [67]. We simulate this on a 2×2 lattice with periodic boundary conditions (see Fig. 2). The purpose of using such a small simulation is simply that it allows one to explore the method with very little cost. The Hamiltonian for the toy model is

$$H = -g \sum_{i=A}^H \sigma_i^x - k \sum_{j=i}^4 \square_j, \quad (13)$$

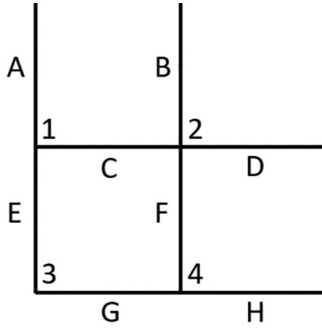


FIG. 2. The diagram of the toy model with four sites and eight links.

where g and k are parameters, σ_i^x is the Pauli x operator on the i th link, and \square_j is the j th plaquette operator (where the j th plaquette is defined as the plaquette with the i th site in the bottom left corner). A plaquette operator in this context is a product of the σ^z operators living on the four sides of a plaquette in the lattice; for example, the first plaquette operator is

$$\square_1 = \sigma_A^z \sigma_B^z \sigma_C^z \sigma_D^z. \quad (14)$$

In this system, each site has an associated local gauge transformation g_i , constructed by taking the product of the σ^x operators living on the four links emanating from the i th site on the lattice:

$$g_1 = \sigma_A^x \sigma_C^x \sigma_D^x \sigma_E^x, \quad (15)$$

$$g_2 = \sigma_B^x \sigma_C^x \sigma_D^x \sigma_F^x, \quad (16)$$

$$g_3 = \sigma_A^x \sigma_E^x \sigma_G^x \sigma_H^x, \quad (17)$$

$$g_4 = \sigma_B^x \sigma_F^x \sigma_G^x \sigma_H^x. \quad (18)$$

We note here that our gauge group \mathbb{Z}_2 is discrete, which means we cannot formulate Gauss' law via generators of gauge transformations. Instead, we note that the Hilbert space of the system is split into sectors defined by the eigenvalues of the gauge transformations in Eqs. (15)–(18). Note that each g_i has eigenvalues ± 1 . With this, we define a physical state as a state $|\psi\rangle$ that is unchanged by all local gauge transformations

$$g_i |\psi\rangle = |\psi\rangle \quad \forall i. \quad (19)$$

At this point we will note that in previous sections we described our algorithm using a G^2 operator [defined in Eq. (10)], but this was for systems governed by compact Lie gauge groups. Here, we have the discrete group \mathbb{Z}_2 , which requires two small alterations. First, we replace the G^2 operator with a new operator G_{tot} :

$$G_{\text{tot}} = g_1 + g_2 + g_3 + g_4. \quad (20)$$

We note that the spectrum of G_{tot} is -4 , 0 , and 4 ; furthermore, using Eqs. (19) and (20), we have that

$$G_{\text{tot}} |\psi\rangle = 4 |\psi\rangle \quad (21)$$

for all physical states. This G_{tot} operator serves the same purpose as the G^2 operator: every physical state is an eigenstate with the same eigenvalue (4 for G_{tot} and 0 for G^2) and all unphysical states have different eigenvalues. Thus, G_{tot} will function the same in our algorithm for our discrete group as G^2 will for compact Lie gauge groups.

Here, we must highlight the second alteration needed for our discrete group. Figure 1 shows phase rotation gates as part of the circuit diagram, but we argue that they are not required in our algorithm in Sec. II A due to physical states having an eigenvalue of 0 . As just discussed, this is not true for our toy model, so we must include the phase rotation gates with $E = 4$, the eigenvalue of physical states for our system.

Now, our aim is to study gauge drift, so we include in our toy model an artificial gauge drift operator D with each time step. We base our constructed gauge drift operators off of Eq. (12); specifically, we connect each of the, in our case 32 , physical states of the system to a randomly selected unphysical state in exactly the way laid out in Eq. (12). All other states are unchanged. This provides a way for the system to drift out of the physical subspace of its Hilbert space in a unitary way, and a parameter, ϵ , to control the strength of this drift. For our simulations, we set this parameter to $\epsilon = 0.01$. Note that each time our method calls for time evolution, we select a different set of 32 unphysical states to appear in the gauge drift operator so as to not bias any unphysical states.

With this defined, the algorithm can be used for the toy model. Each time step takes a state $|\psi\rangle$ to $\tilde{g}_i D U_t |\psi\rangle$ where \tilde{g}_i is a randomly chosen gauge transformation, D is a gauge drift operator, and $U_t = \exp -iHt$ is the standard time evolution operator. Note that \tilde{g}_i represents any gauge transformation on the system, including the g_i 's in Eqs. (15)–(18) as well as any product of them. The remaining part of the algorithm is the rodeo projection, which occurs after a set number of time steps (determined in the calibration phase). As discussed above, a successful projection takes a state $|\psi\rangle$ to $\frac{1}{2} [1 + \exp[-i(G_{\text{tot}} - 4)t_r]] |\psi\rangle$ where here we note that the time used within the projection t_r is, in general, different from the time step size t . Recall that all physical states are eigenstates of G_{tot} with an eigenvalue of 4 , so they are unaffected while unphysical states will suffer destructive interference.

Figure 3 demonstrates the basic performance of the method. It shows no suppression, a method of using a random gauge transformation after each time step, and our method of using a random gauge transformation after each time step plus projections, in this case after every time step. It can be seen both that the use of random gauge transformations after every time step greatly suppresses the gauge drift as well as that frequent rodeo projections does in fact keep the system very close to the physical subspace.

In the rest of this section we will consider the frequency of projections. To study this, we ran simulations to test many different regimes to compare their effectiveness. In particular, systems without any gauge transformations after time steps with a frequency of rodeo projections ranging from after one time step to after ten time steps. We also considered systems with a random gauge transformation after each time step with a frequency of gauge projections ranging from after one time

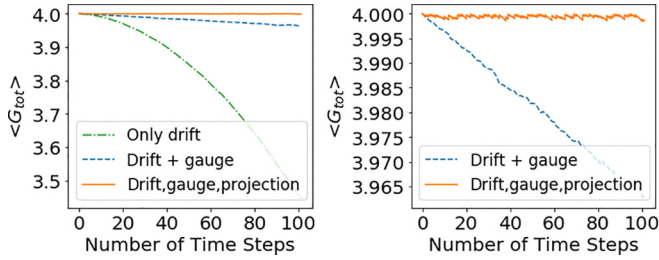


FIG. 3. Comparing gauge drift suppression methods, including no suppression methods (dash-dot line), a random gauge transformation after each time step (dashed line), and a random gauge transformation followed by a projection after each time step (solid line). The right plot is the same as the left plot without the dash-dot plot so as to zoom in on the behavior of the other two.

step to after 30 time steps. For each regime, we ran our algorithm on 200 systems and calculated the average of the farthest from physical each run reached (which is to say the lowest value of $\langle G_{\text{tot}} \rangle$).

Figure 4 shows these results. It is clear that including the random gauge transformations into the method significantly helps to reduce the required frequency of gauge projections. In particular, it shows the following approximate equivalences: a projection after every time step performs similarly to a projection every two time steps when gauge transformations are included; a rodeo projection after every two time steps performs similarly to a gauge projection every five time steps when gauge transformations are included; and a gauge projection after every three time steps performs similarly to a gauge projection every ten time steps when gauge transformations are included. This shows that including the gauge transformations reduces the required frequency of rodeo projections by more than half. In particular, we see that for a system that drifts out of the physical subspace slowly, one could perform projections infrequently and still be able to utilize the Zeno effect to great advantage.

IV. IMPLEMENTATION

It is important to consider the implementation of this method, as well as the associated costs. As our method

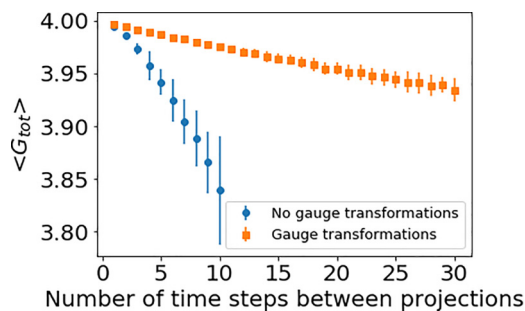


FIG. 4. Comparing performance of regimes with no gauge transformations and various frequencies of projections (circle markers) with regimes with gauge transformations and various frequencies of projections (square markers).

involves two main operations (rodeo projections and gauge transformations), we will take them in turn.

A. Implementing Rodeo projections

With a circuit diagram of Fig. 1 plus the small alterations discussed in Sec. II A in hand, there remain two main questions when it comes to the implementation of our version of rodeo projections: how should the time parameters, call them t_r , be chosen, and how many ancillary qubits per projection are required?

First, we address the issue of the selection of the t_r parameters. Here, we note that previous work has shown that the rodeo algorithm can be optimized and made far more efficient [64] via the considered choices for the parameter t_r . We suggest that this optimized approach makes sense for our purposes as well. This work notes that the dominant cost of the rodeo algorithm is the controlled time evolution gate. In our case it is then the controlled U gate with $U = \exp(-iG^2 t_r)$. We will also note that constructing G^2 will be much more difficult than H so our version of a rodeo projection suffers a much heftier cost in construction of the required gates, but so will alternative projection schemes. Due to the controlled U gate dominating costs, the authors of [64] argue that the total time to conduct a rodeo projection acts as a proxy for computational costs and then proceeds to lay out a method for selecting the t_r s that maximizes efficiency of the algorithm given fixed computational resources [64]. The authors note that choosing the t_r s randomly (e.g., via Gaussian distributions) can lead to exponentially large fluctuations in the suppression of undesired amplitudes; furthermore, they show that considered choices of the t_r s, specifically choosing t_r s that vary across exponentially many timescales, can eliminate the problem of these large fluctuations, thus leading to a more efficient implementation of the rodeo projection.

Additionally, we must address the question of qubit costs. Notably, if midcircuit measurements are allowed, then only one ancillary qubit is required, as it can be reused again and again. If this is not the case, then we estimate one projection would need roughly as many ancillary qubits as sites in the lattice. Future work aims to study how the qubit per projection requirement scales with system size.

B. Implementing gauge transformations

It is important to discuss the practical implementation of random gauge transformations as the authors of Ref. [66] outlined in Sec. III of their paper. First, we must sample from the Lie group G of gauge transformations. One way is to construct an ancillary “clock” register that increases its value by 1 after each gauge transformation is conducted in the course of our algorithm. Then a random circuit (e.g., of the kind developed in Ref. [68]) will take the value of the clock and select a gauge transformation for each site and store it on an ancillary G -register. Once sampled, these gauge transformations can be performed on the system.

The first major cost consideration of this process is qubit costs. The sampling process requires two additional registers (the clock and the G -registers); notably, this qubit cost does not scale with the size of the system. Furthermore, the actual

implementation of the gauge transformations on the system “requires about as many G -multiplication gates as the original quantum simulation did” [66].

A second cost consideration is the depth of the random circuit. Intuitively, shorter circuits will result in less fair sampling from the Haar measure of the Lie group than longer circuits. Luckily, we are helped significantly by the fact that we have no strict requirements on fair sampling from the Haar measure. Here, we point to a common method of sampling uniformly from the Haar measure of a group G , which is to select a small number (e.g., two) of elements of G , call them g_1 and g_2 , and construct strings of these elements of a chosen length (an example of a string of length 5 is $g_1 g_2 g_2 g_1 g_2$). Importantly, this method of sampling elements of G converges to the Haar measure in the limit of long strings. This tells us that unfair sampling can be made more fair by conducting multiple unfairly sampled gauge transformations. Thus, we can use a short random circuit when sampling gauge transformations and then perform multiple gauge transformations in a row if need be to increase the fairness of the sampling. Future work aims to clarify the relationship between the fairness of sampling and the performance of the algorithm.

V. CONCLUSION

We outlined an algorithm developed to maintain gauge invariance (or more precisely suppress gauge drift) during the simulation of lattice gauge theories. Our algorithm was based off of the Zeno effect, wherein we used frequent projections (i.e., altered rodeo projections) to keep the system within the physical subspace of its exponentially larger Hilbert space. We also included a technique of performing gauge transformations after every time step to help reduce the system’s gauge drift into the unphysical sector, which, in turn, allowed for less frequent projections while still taking full advantage of the Zeno effect. We showed that our method was effective for a simple small toy model of pure \mathbb{Z}_2 gauge theory. Future work aims to test our algorithm on larger systems, and more complicated gauge theories such as $SU(N)$ or gauge theories that include matter.

ACKNOWLEDGMENTS

This work was supported, in part, by the U.S. Department of Energy, Office of Nuclear Physics under Awards No. DE-SC0021143 and No. DE-FG02-93ER40762.

-
- [1] P. Benioff, The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines, *J. Stat. Phys.* **22**, 563 (1980).
 - [2] R. P. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* **21**, 467 (1982).
 - [3] S. Lloyd, Universal quantum simulators, *Science* **273**, 1073 (1996).
 - [4] K. L. Brown, W. J. Munro, and V. M. Kendon, Using quantum computers for quantum simulation, *Entropy* **12**, 2268 (2010).
 - [5] S. P. Jordan, K. S. M. Lee, and J. Preskill, Quantum algorithms for quantum field theories, *Science* **336**, 1130 (2012).
 - [6] G. Jaeger, Classical and quantum computing, *Quantum Information: An Overview* (Springer, New York, 2007), pp. 203–217.
 - [7] Y. Zhou, E. M. Stoudenmire, and X. Waintal, What limits the simulation of quantum computers? *Phys. Rev. X* **10**, 041038 (2020).
 - [8] A. Parra-Rodriguez, P. Lougovski, L. Lamata, E. Solano, and M. Sanz, Digital-analog quantum computation, *Phys. Rev. A* **101**, 022305 (2020).
 - [9] E. Zohar, Quantum simulation of lattice gauge theories in more than one space dimension—requirements, challenges and methods, *Philos. Trans. R. Soc. A* **380**, 20210069 (2021).
 - [10] M. C. Bañuls, R. Blatt, J. Catani, A. Celi, J. I. Cirac, M. Dalmonte, L. Fallani, K. Jansen, M. Lewenstein, S. Montangero, C. A. Muschik, B. Reznik, E. Rico, L. Tagliacozzo, K. Van Acoleyen, F. Verstraete, U.-J. Wiese, M. Wingate, J. Zakrzewski, and P. Zoller, Simulating lattice gauge theories within quantum technologies, *Eur. Phys. J. D* **74**, 165 (2020).
 - [11] M. Dalmonte and S. Montangero, Lattice gauge theory simulations in the quantum information era, *Contemp. Phys.* **57**, 388 (2016).
 - [12] T. Byrnes and Y. Yamamoto, Simulating lattice gauge theories on a quantum computer, *Phys. Rev. A* **73**, 022328 (2006).
 - [13] E. Zohar, A. Farace, B. Reznik, and J. I. Cirac, Digital lattice gauge theories, *Phys. Rev. A* **95**, 023604 (2017).
 - [14] E. Zohar, A. Farace, B. Reznik, and J. I. Cirac, Digital quantum simulation of \mathbb{Z}_2 lattice gauge theories with dynamical fermionic matter, *Phys. Rev. Lett.* **118**, 070501 (2017).
 - [15] A. Mezzacapo, E. Rico, C. Sabín, I. L. Egusquiza, L. Lamata, and E. Solano, Non-abelian $SU(2)$ lattice gauge theories in superconducting circuits, *Phys. Rev. Lett.* **115**, 240502 (2015).
 - [16] D. Banerjee, M. Dalmonte, M. Müller, E. Rico, P. Stebler, U.-J. Wiese, and P. Zoller, Atomic quantum simulation of dynamical gauge fields coupled to fermionic matter: From string breaking to evolution after a quench, *Phys. Rev. Lett.* **109**, 175302 (2012).
 - [17] E. Zohar, J. I. Cirac, and B. Reznik, Quantum simulations of gauge theories with ultracold atoms: Local gauge invariance from angular-momentum conservation, *Phys. Rev. A* **88**, 023617 (2013).
 - [18] A. Bazavov, Y. Meurice, S.-W. Tsai, J. Unmuth-Yockey, and J. Zhang, Gauge-invariant implementation of the Abelian-Higgs model on optical lattices, *Phys. Rev. D* **92**, 076003 (2015).
 - [19] J. Zhang, J. Unmuth-Yockey, J. Zeiher, A. Bazavov, S.-W. Tsai, and Y. Meurice, Quantum simulation of the universal features of the polyakov loop, *Phys. Rev. Lett.* **121**, 223201 (2018).
 - [20] N. Klco, E. F. Dumitrescu, A. J. McCaskey, T. D. Morris, R. C. Pooser, M. Sanz, E. Solano, P. Lougovski, and M. J. Savage, Quantum-classical computation of Schwinger model dynamics using quantum computers, *Phys. Rev. A* **98**, 032331 (2018).
 - [21] D. C. Hackett, K. Howe, C. Hughes, W. Jay, E. T. Neil, and J. N. Simone, Digitizing gauge fields: Lattice Monte Carlo results for future quantum computers, *Phys. Rev. A* **99**, 062341 (2019).

- [22] I. Raychowdhury and J. R. Stryker, Solving Gauss's law on digital quantum computers with loop-string-hadron digitization, *Phys. Rev. Res.* **2**, 033039 (2020).
- [23] T. V. Zache, F. Hebenstreit, F. Jendrzejewski, M. K. Oberthaler, J. Berges, and P. Hauke, Quantum simulation of lattice gauge theories using Wilson fermions, *Quantum Sci. Technol.* **3**, 034010 (2018).
- [24] J. Bender, E. Zohar, A. Farace, and J. I. Cirac, Digital quantum simulation of lattice gauge theories in three spatial dimensions, *New J. Phys.* **20**, 093001 (2018).
- [25] F. M. Surace, P. P. Mazza, G. Giudici, A. Lerose, A. Gambassi, and M. Dalmonte, Lattice gauge theories and string dynamics in Rydberg atom quantum simulators, *Phys. Rev. X* **10**, 021041 (2020).
- [26] J. B. Kogut, An introduction to lattice gauge theory and spin systems, *Rev. Mod. Phys.* **51**, 659 (1979).
- [27] J. Kogut and L. Susskind, Hamiltonian formulation of Wilson's lattice gauge theories, *Phys. Rev. D* **11**, 395 (1975).
- [28] J. Smit, *Introduction to Quantum Fields on a Lattice*, Cambridge Lecture Notes in Physics (Cambridge University Press, Cambridge, England, 2002).
- [29] G. Rossi and M. Testa, The structure of Yang-Mills theories in the temporal gauge: (I). General formulation, *Nucl. Phys. B* **163**, 109 (1980).
- [30] E. Zohar, J. I. Cirac, and B. Reznik, Cold-atom quantum simulator for SU(2) Yang-Mills lattice gauge theory, *Phys. Rev. Lett.* **110**, 125304 (2013).
- [31] L. Tagliacozzo, A. Celi, P. Orland, M. W. Mitchell, and M. Lewenstein, Simulation of non-Abelian gauge theories with optical lattices, *Nat. Commun.* **4**, 2615 (2013).
- [32] E. Zohar and B. Reznik, Confinement and lattice quantum-electrodynamic electric flux tubes simulated with ultracold atoms, *Phys. Rev. Lett.* **107**, 275301 (2011).
- [33] E. Zohar, J. I. Cirac, and B. Reznik, Simulating compact quantum electrodynamics with ultracold atoms: Probing confinement and nonperturbative effects, *Phys. Rev. Lett.* **109**, 125302 (2012).
- [34] D. Banerjee, M. Bögli, M. Dalmonte, E. Rico, P. Stebler, U.-J. Wiese, and P. Zoller, Atomic quantum simulation of U(n) and SU(n) non-Abelian lattice gauge theories, *Phys. Rev. Lett.* **110**, 125303 (2013).
- [35] D. Marcos, P. Rabl, E. Rico, and P. Zoller, Superconducting circuits for quantum simulation of dynamical gauge fields, *Phys. Rev. Lett.* **111**, 110504 (2013).
- [36] P. Hauke, D. Marcos, M. Dalmonte, and P. Zoller, Quantum simulation of a lattice Schwinger model in a chain of trapped ions, *Phys. Rev. X* **3**, 041018 (2013).
- [37] M. V. Damme, J. C. Halimeh, and P. Hauke, Gauge-symmetry violation quantum phase transition in lattice gauge theories, [arXiv:2010.07338](https://arxiv.org/abs/2010.07338).
- [38] J. C. Halimeh and P. Hauke, Reliability of lattice gauge theories, *Phys. Rev. Lett.* **125**, 030503 (2020).
- [39] J. C. Halimeh, H. Lang, and P. Hauke, Gauge protection in non-Abelian lattice gauge theories, *New J. Phys.* **24**, 033015 (2022).
- [40] J. C. Halimeh, L. Homeier, A. Bohrdt, and F. Grusdt, Spin exchange-enabled quantum simulator for large-scale non-Abelian gauge theories, [arXiv:2305.06373](https://arxiv.org/abs/2305.06373).
- [41] J. C. Halimeh, H. Lang, J. Mildenerger, Z. Jiang, and P. Hauke, Gauge-symmetry protection using single-body terms, *PRX Quantum* **2**, 040311 (2021).
- [42] J. C. Halimeh, L. Homeier, C. Schweizer, M. Aidelsburger, P. Hauke, and F. Grusdt, Stabilizing lattice gauge theories through simplified local pseudogenerators, *Phys. Rev. Res.* **4**, 033120 (2022).
- [43] M. V. Damme, J. Mildenerger, F. Grusdt, P. Hauke, and J. C. Halimeh, Suppressing nonperturbative gauge errors in the thermodynamic limit using local pseudogenerators, [arXiv:2110.08041](https://arxiv.org/abs/2110.08041).
- [44] L. Homeier, A. Bohrdt, S. Linsel, E. Demler, J. C. Halimeh, and F. Grusdt, Realistic scheme for quantum simulation of \mathbb{Z}_2 lattice gauge theories with dynamical matter in (2 + 1)D, *Commun. Phys.* **6**, 127 (2023).
- [45] P. Facchi and S. Pascazio, Quantum zeno subspaces, *Phys. Rev. Lett.* **89**, 080401 (2002).
- [46] E. A. Martinez, C. A. Muschik, P. Schindler, D. Nigg, A. Erhard, M. Heyl, P. Hauke, M. Dalmonte, T. Monz, P. Zoller, and R. Blatt, Real-time dynamics of lattice gauge theories with a few-qubit quantum computer, *Nature (London)* **534**, 516 (2016).
- [47] Y. Y. Atas, J. Zhang, R. Lewis, A. Jahanpour, J. F. Haase, and C. A. Muschik, Su(2) hadrons on a quantum computer via a variational approach, *Nat. Commun.* **12**, 6499 (2021).
- [48] E. Zohar and J. I. Cirac, Eliminating fermionic matter fields in lattice gauge theories, *Phys. Rev. B* **98**, 075119 (2018).
- [49] E. Zohar and J. I. Cirac, Removing staggered fermionic matter in U(N) and SU(N) lattice gauge theories, *Phys. Rev. D* **99**, 114511 (2019).
- [50] G. Pardo, T. Greenberg, A. Fortinsky, N. Katz, and E. Zohar, Resource-efficient quantum simulation of lattice gauge theories in arbitrary dimensions: Solving for Gauss's law and fermion elimination, *Phys. Rev. Res.* **5**, 023077 (2023).
- [51] R. Irmejs, M.-C. Bañuls, and J. I. Cirac, Quantum simulation of \mathbb{Z}_2 lattice gauge theory with minimal resources, *Phys. Rev. D* **108**, 074503 (2023).
- [52] P. P. Popov, M. Meth, M. Lewenstein, P. Hauke, M. Ringbauer, E. Zohar, and V. Kasper, Variational quantum simulation of U(1) lattice gauge theories with qudit systems, *Phys. Rev. Res.* **6**, 013202 (2024).
- [53] D. B. Kaplan and J. R. Stryker, Gauss's law, duality, and the Hamiltonian formulation of U(1) lattice gauge theory, *Phys. Rev. D* **102**, 094515 (2020).
- [54] J. F. Unmuth-Yockey, Gauge-invariant rotor Hamiltonian from dual variables of 3D U(1) gauge theory, *Phys. Rev. D* **99**, 074502 (2019).
- [55] C. W. Bauer and D. M. Grabowska, Efficient representation for simulating U(1) gauge theories on digital quantum computers at all values of the coupling, [arXiv:2111.08015](https://arxiv.org/abs/2111.08015).
- [56] H. Lamm, S. Lawrence, and Y. Yamauchi, General methods for digital quantum simulation of gauge theories, *Phys. Rev. D* **100**, 034518 (2019).
- [57] V. Kasper, T. V. Zache, F. Jendrzejewski, M. Lewenstein, and E. Zohar, Non-Abelian gauge invariance from dynamical decoupling, *Phys. Rev. D* **107**, 014506 (2023).
- [58] J. R. Stryker, Oracles for Gauss's law on digital quantum computers, *Phys. Rev. A* **99**, 042301 (2019).
- [59] K. Stannigel, P. Hauke, D. Marcos, M. Hafezi, S. Diehl, M. Dalmonte, and P. Zoller, Constrained dynamics via the zeno effect in quantum simulation: Implementing non-Abelian lattice gauge theories with cold atoms, *Phys. Rev. Lett.* **112**, 120406 (2014).

- [60] P. Facchi and S. Pascazio, Quantum zeno dynamics: Mathematical and physical aspects, *J. Phys. A: Math. Theor.* **41**, 493001 (2008).
- [61] K. Choi, D. Lee, J. Bonitati, Z. Qian, and J. Watkins, Rodeo algorithm for quantum computing, *Phys. Rev. Lett.* **127**, 040505 (2021).
- [62] Z. Qian, J. Watkins, G. Given, J. Bonitati, K. Choi, and D. Lee, Demonstration of the rodeo algorithm on a quantum computer, *Eur. Phys. J. A* **60**, 151 (2024).
- [63] M. Bee-Lindgren, Z. Qian, M. DeCross, N. C. Brown, C. N. Gilbreth, J. Watkins, X. Zhang, and D. Lee, Rodeo algorithm with controlled reversal gates, [arXiv:2208.13557](https://arxiv.org/abs/2208.13557).
- [64] T. D. Cohen and H. Oh, Optimizing the rodeo projection algorithm, *Phys. Rev. A* **108**, 032422 (2023).
- [65] Y. Dokshitzer, D. Dyakonov, and S. Troyan, Hard processes in quantum chromodynamics, *Phys. Rep.* **58**, 269 (1980).
- [66] H. Lamm, S. Lawrence, and Y. Yamauchi, Suppressing coherent gauge drift in quantum simulations, [arXiv:2005.12688](https://arxiv.org/abs/2005.12688).
- [67] S. Sachdev, \mathbb{Z}_2 gauge theory (2018).
- [68] A. Alexandru, P. F. Bedaque, and S. Lawrence, Quantum algorithms for disordered physics, *Phys. Rev. A* **101**, 032325 (2020).