Fault-tolerant simulation of Lattice Gauge Theories with gauge covariant codes

Luca Spagnoli[1,2], Alessandro Roggero[1,2], and Nathan Wiebe[3,4,5]

[1]Dipartimento di Fisica, University of Trento, via Sommarive 14, I–38123, Povo, Trento, Italy
[2]INFN-TIFPA Trento Institute of Fundamental Physics and Applications, Trento, Italy
[3]Department of Computer Science, University of Toronto, Toronto, ON M5S 2E4, Canada
[4]Pacific Northwest National Laboratory, Richland, WA 99354, USA
[5]Department of Physics, University of Washington, Seattle, WA 98195, USA
2025-12-30

We show in this paper that a strong and easy connection exists between quantum error correction and Lattice Gauge Theories (LGT) by using the Gauge symmetry to construct an efficient error-correcting code for an Abelian $\mathbb{Z}_2$ LGT. We identify the logical operations on this gauge covariant code and show that the corresponding Hamiltonian can be expressed in terms of these logical operations while preserving the spatial locality of the interactions. Furthermore, we demonstrate that these substitutions actually give a new way of writing the LGT as an equivalent hardcore boson model. Finally we demonstrate a method to perform fault-tolerant time evolution of the Hamiltonian within the gauge covariant code using both product formulas and qubitization approaches. This opens up the possibility of inexpensive end to end dynamical simulations that save physical qubits by blurring the lines between simulation algorithms and quantum error correcting codes.

1 Introduction

The field of quantum computation keeps making big steps toward platforms that will be able to perform some calculations with an exponential speedup compared to classical computers. An important example of such calculations could be the simulation of Standard Model physics using Lattice Gauge Theories (LGTs), where classical computers struggle with an exponential complexity, while quantum computers are a promising way to solve the problem. Even though a formal proof that the entire gauge theory describing the Standard Model could be simulated on quantum computers with an exponential speedup is still missing, hints that this might be the case are available since the pioneering work on scalar field theory where the computation of scattering amplitudes can be shown to be BQP-complete [1, 2]. In the past years, many efforts have been made in order to design efficient algorithms for the digital quantum simulations of LGTs for a variety of phenomenologically important gauge groups (for recent reviews see [3, 4, 5, 6])

Luca Spagnoli: luca.spagnoli@unitn.it
Alessandro Roggero: a.roggero@unitn.it
Nathan Wiebe: nawiebe@cs.toronto.edu

An important challenge to overcome in order to enable practical large scale simulations using quantum computing platforms is the efficient and scalable control of hardware noise that, if left undisturbed, can lead to a catastrophic accumulation of errors in a quantum simulation. A possible solution to this challenge, inspired by classical error-correcting codes, has emerged since the early days of quantum computing and allows for controllable, and efficient, long time computations by means of Quantum Error Correction (QEC) [7, 8, 9, 10, 11, 12]. Much like classical error correction, in QEC one exploits a redundant description of the calculation in order to detect and correct localized errors that might appear during the execution. Since physical theories that describe both fermions as well as gauge fields on a lattice have natural redundancies generated by the presence of symmetries, they provide a fertile ground for exploring special purpose error correction strategies tailored to quantum simulations of these models. Indeed several recent works have explored the possibility of designing error detection and error correction protocols for these physical theories with the goal of both improving the performance of error correction but also as a mean to understand the properties of gauge theories from the point of view of QEC [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 45]. Inspired by the work of Stryker on error detection oracles for gauge theories employing a direct check of local Gauss' laws [13], a recent work by some of us described a complete and fault-tolerant error correction scheme for $\mathbb{Z}_2$ lattice gauge theories in one and two spatial dimensions [18]. In this work we extend the QEC scheme presented there in several ways. Our first new contribution is a generalization of the Gauss' Law code of Ref. [18] to arbitrary spatial dimensions and at the same time improving the resource requirements of the original proposals for $2D$. We then characterize the logical operations of the code and show how to leverage the error correction formulation to express the logical Hamiltonian operator in the physical subspace effectively removing the gauge constraint. In the models we consider here, the resulting logical Hamiltonian can be thought of as being obtained integrating out the fermionic degrees of freedom living on the lattice sites. The idea itself is not new and several explicit constructions for mapping a lattice gauge into a spin model by removing the fermionic matter have been proposed (see e.g. [23, 24, 25]). Typically these strategies are tailored to specific spatial dimensions and require representing the fermions in terms of Majorana modes, our approach on the other hand is independent on the spatial dimension and easy to construct. Finally, the original proposal for these Gauss' Law codes in Ref. [18] focused solely on the use of the error-correcting code to store information in a fault-tolerant way, in this work we show how to implement universal gate set on a Gauss' Law code and provide a complete fault-tolerant algorithm to perform Hamiltonian simulation in this framework. For simulations using product formulas we present a scheme that requires only 3 (logical) ancilla qubits in addition to the memory qubits required to encode the system.

In Section 2, we define the Hamiltonian and the LGT we take into account, as well as the corresponding gauge operators. In Section 3, we review the basic concepts behind the stabilizer formalism and we will show how to exploit the gauge symmetry to do error correction. Moreover, we will explicitly build a distance-3 error-correcting code. In Section 4 the Hamiltonian is written in terms of the logical operations of the error-correcting code, reducing the total number of degrees of freedom to those required to span the physical Hilbert space. We will also show that the new Hamiltonian can be written as a function of only hard-core bosonic degrees of freedom. Finally, in Section 5, we discuss how simulation algorithms can be implemented using Quantum Signal Processing [26, 27] and Trotterization [28] in a completely fault tolerant way using the error-correcting code developed in Section 3. This not only proves that the Gauss' Law can be exploited to correct errors, but also shows how to use it to perform fault-tolerant simulations.

2 Abelian lattice gauge theories and Gauss' Law

We will consider a $\mathbb{Z}_N$ model Hamiltonian which contains the fermionic mass and hopping terms, the electric field term, and the self-interaction of the gauge field, which translates into the plaquette interaction [29, 30, 31]. The theory is defined on a cubic lattice in d spatial dimensions with fermions defined on the lattice sites and gauge bosons on the links. We label the sites of the lattice as $S_{\vec{l}}$, where $\vec{l}$ are d-dimensional vectors with integer components, and define the set of d unit vectors in the positive directions as $\mathcal{D}$ (e.g., for $d = 2$ we have $\mathcal{D} = \{(1, 0), (0, 1)\}$. With this notation, the Hamiltonian of the system can be written as:

$$H = H_M + H_{hop} + H_E + H_P \tag{1}$$

The different terms correspond, in order, to a mass term H_M, an hopping term H_{hop}, an electric term H_E and a plaquette, or magnetic, term H_P. The explicit definition of the mass term is as follows

$$H_M = m \sum_{\vec{l}} \sigma_{\vec{l}}\, \psi^\dagger_{\vec{l}} \psi_{\vec{l}}\,, \tag{2}$$

with $\psi_{\vec{l}}\,(\psi^\dagger_{\vec{l}})$ fermionic annihilation (creation) operators acting on the sites $S_{\vec{l}}$ of the lattice and

$$\sigma_{\vec{l}} = (-1)^{l_1+\cdots+l_d} := (-1)^{|l|}\,, \tag{3}$$

a sign factor. The hopping term is instead given by

$$H_{hop} = \epsilon \sum_{\vec{l}} \sum_{\hat{k}\in\mathcal{D}} \sigma_{\vec{l},\hat{k}} \left(\psi^\dagger_{\vec{l}} Q_{\vec{l},\hat{k}} \psi_{\vec{l}+\hat{k}} + \psi^\dagger_{\vec{l}+\hat{k}} Q^\dagger_{\vec{l},\hat{k}} \psi_{\vec{l}} \right)\,. \tag{4}$$

The operator $Q_{\vec{l},\hat{k}}$ acts on the link $L_{\vec{l},\hat{k}}$ which connects the site $S_{\vec{l}}$ to the site $S_{\vec{l}+\hat{k}}$ as a lowering operator in the computational basis

$$Q\,|m\rangle = |m-1\rangle\,, \tag{5}$$

while $\sigma_{\vec{l},\hat{k}}$ is a sign factor taking values (-1) in the 1-direction, $(-1)^{l_1}$ in the 2 direction and $(-1)^{l_1+\cdots+l_{d-1}}$ in the d direction. The electric term can be written as

$$H_E = \lambda_E \sum_{\vec{l}} \sum_{\hat{k}\in\mathcal{D}} \left(P_{\vec{l},\hat{k}} + P^\dagger_{\vec{l},\hat{k}} \right)\,, \tag{6}$$

with the operators $P_{\vec{l},\hat{k}}$ acting on the links $L_{\vec{l},\hat{k}}$ as

$$P\,|m\rangle = e^{i\frac{2\pi}{N}m}\,|m\rangle\,, \tag{7}$$

in the computational basis. The P and Q operators satisfy the $\mathbb{Z}_N$ algebra

$$P^N = Q^N = 1 \qquad P^\dagger Q P = e^{i\frac{2\pi}{N}} Q\,. \tag{8}$$

In a $U(1)$ theory, the electric term is a sum of the Casimir operators $E^2_{\vec{l},\hat{k}}$. The connection with the Z_n case can be done by writing the electric term as

$$H_E = \lambda_E \sum_{\vec{l}} \sum_{\hat{k}\in\mathcal{D}} O_{\vec{l},\hat{k}} \tag{9}$$

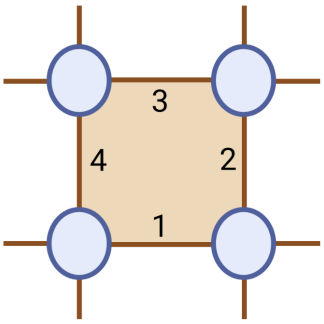


Figure 1: Labelling convention for links in a plaquette.

where the operator $O_{\vec{l},\hat{k}}$ is in general

$$O = \sum_j f(j) \left|j\right\rangle \left\langle j\right| \tag{10}$$

The only constraint is on $f(j)$, which as to approach $f(j) \to j^2$ as $N \to \infty$. So we can chose

$$f(j) = 4\sin^2\left(\frac{\pi j}{N}\right) \tag{11}$$

and expanding in complex exponentials we get

$$\sum_j f(j) \left|j\right\rangle \left\langle j\right| = \sum_j \left(e^{i\frac{2\pi}{N}j} + e^{-i\frac{2\pi}{N}j}\right) \left|j\right\rangle \left\langle j\right| = P + P^\dagger \tag{12}$$

Finally, the plaquette term takes the form

$$H_P = \lambda_P \sum_p \left(W_p + W_p^\dagger\right) \ , \tag{13}$$

where the sum is over all the plaquettes in the lattice and the operators $W_p = Q_1 Q_2 Q_3^\dagger Q_4^\dagger$ are formed by four Q operators acting on the links of the plaquette using the ordering convention shown in Fig. 1.

In terms of the basic building blocks defined above, the Gauss' Law operator can be written as

$$G_{\vec{l}} = q_{\vec{l}}^\dagger \prod_{\hat{k}\in\mathcal{D}} P_{\vec{l},\hat{k}}^\dagger P_{\vec{l}-\hat{k},\hat{k}} \ , \tag{14}$$

where $q_{\vec{l}}$ is a unitary operator connected with the charge $n_{\vec{l}}$ at a site

$$q_{\vec{l}} = e^{-i\frac{2\pi}{N}n_{\vec{l}}} \qquad n_{\vec{l}} = \psi_{\vec{l}}^\dagger \psi_{\vec{l}} - \frac{1}{2}\left(1 - (-1)^{|l|}\right) \ . \tag{15}$$

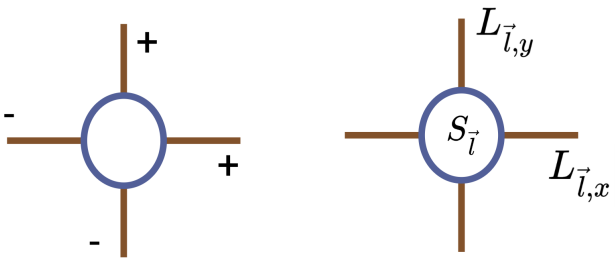


Figure 2: Convention for positive and negative links around a site on the left-hand side, and labelling convention on the right-hand side.

The term $\psi^\dagger_{\vec{l}}\psi_{\vec{l}}$ in $n_{\vec{l}}$ gives 0 if an even site is empty or an odd site is full, and 1 in the other cases. The convention used in the definition of Eq. (14) is to consider the incoming links into a site to be negative and the outgoing as positive (see Fig. 2). Physical states of the theory, i.e. those that satisfy the local $\mathbb{Z}_N$ symmetry, are thus eigenstates of the operators $G_{\vec{l}}$ at every site of the lattice

$$G_{\vec{l}}|phys\rangle = e^{-i\frac{2\pi}{N}\epsilon_{\vec{l}}}|phys\rangle \quad \forall\vec{l}\,, \tag{16}$$

where the $\epsilon_{\vec{l}}$ denote the possible static charge present on the site $S_{\vec{l}}$. Different assignments of these charges define sectors of the physical theory which do not mix between themselves. For simplicity in the rest of the present work we will work in the sector with zero static charges but, if needed, other sectors can be described with the same techniques. In the zero charge sector, physical states obey therefore

$$G_{\vec{l}}|phys\rangle = |phys\rangle \quad \forall\vec{l}\,. \tag{17}$$

This relation can be connected to Gauss' Law by first introducing electric field operator $E_{\vec{l},\hat{k}}$ as

$$P^\dagger_{\vec{l},\hat{k}} = e^{i\frac{2\pi}{N}E_{\vec{l},\hat{k}}}\,, \tag{18}$$

Then, Eq. 17 becomes:

$$e^{i\frac{2\pi}{N}\left(\boldsymbol{\nabla}\cdot E_{\vec{l}}-n_{\vec{l}}\right)}|phys\rangle = |phys\rangle \tag{19}$$

and it is equivalent to the following condition

$$\left[\boldsymbol{\nabla}\cdot E_{\vec{l}}-n_{\vec{l}}\right]_{\text{mod }N}|phys\rangle = \left[\sum_{\hat{k}\in\mathcal{D}}\left(E_{\vec{l},\hat{k}}-E_{\vec{l}-\hat{k},\hat{k}}\right)-n_{\vec{l}}\right]_{\text{mod }N}|phys\rangle = 0 \tag{20}$$

Since Eq. (17) and Eq. (20) are equivalent, we will use the condition in Eq. (17) since it is unitary.

As it is done in [18], we will limit ourselves to the choice of the electric field cutoff equal to 2, which means that the symmetry group becomes $\mathbb{Z}_2$. It further allows us to implement the stabilizer operations using low-weight Pauli operations and also allows us to represent the field via a single physical qubit. In this case the P and Q operators from Eq. (7) and Eq. (5), and the Gauss law operator, can be chosen as

$$P = Z, \quad Q = X, \quad G_{\vec{l}} = (-1)^{|l|}Z_{S_{\vec{l}}}\prod_{\hat{k}\in\mathcal{D}}Z_{L_{\vec{l},\hat{k}}}Z_{L_{\vec{l}-\hat{k},\hat{k}}} \tag{21}$$

where X, Z are the standard Pauli matrices σ_X, σ_Z. In this case of $\mathbb{Z}_2$, those operators P, Q, and G are all Pauli operators, which means that we can interpret the set of G as the generator of the stabilizer group. Of course, it is not enough since it commutes with every Z error, but it is not a problem since we can always concatenate two classical error-correcting codes to get a quantum one.

3 Quantum error correction with Gauss' Law

In this section we introduce the error-correcting code that uses Gauss' law as generators of the stabilizer group generalizing the results derived in Ref. [18]. First, let us recall how stabilizers work: we need an abelian subgroup of the Pauli group that will be the stabilizer group. The generators of this group are then called stabilizers, while the codewords (states

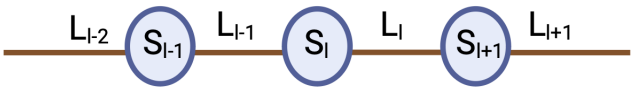


Figure 3: Representation of a 1-dimensional lattice, showing the labelling convention for sites and links.

G_{l-1}	G_l	G_{l+1}	error location
$+$	$+$	$+$	none
$-$	$-$	$+$	L_{l-1}
$+$	$-$	$+$	S_l
$+$	$-$	$-$	L_l

Table 1: In this table we reported only the errors on S_l and the links attached to it, so where the G_l fails. Other outcomes correspond to multiple errors that we cannot correct, or to errors in $S_{l\pm1}$ for which we need to look at $G_{l\pm2}$.

protected against errors by the code) are defined as the eigenvectors with eigenvalue $+1$ of every stabilizer operator. By measuring the eigenvalue of those stabilizers (which means extracting the error syndrome) we can understand if an error occurred with the further intention of deducing which qubit the error happened on. Of course, all errors cannot be detected in this manner: only errors that anti-commute with at least one stabilizer S_i can be measured in this fashion since it will change the eigenvalue of that stabilizer from $+1$ to -1 when measured. Errors that commute with the stabilizer group lead to logical errors, which cannot be corrected.

Now, we will prove that the group generated by the Gauss' Law operators is a stabilizer group of an error-correcting code able to correct every single-qubit X error, which is the same of saying it generates a classical distance-3 error-correcting code.

Theorem 1. *Let H be the Hamiltonian of Eq (1), with a $\mathbb{Z}_2$ gauge group, and let $G_{\vec{l}}$ of Eq. (14) be the generators of the local symmetry. Given a d-dimensional lattice with N sites and dN links, mapped into $N + dN$ qubits, a distance 3 quantum error-correcting code that corrects single-qubit X errors (a classical distance-3 error-correcting code) can be built using only $G_{\vec{l}}$ as stabilizers for every $N \geq 3^d$.*

Proof. For simplicity consider the 1-dimensional system depicted in Fig. 3. Since the Gauss' law operator on the l-th site is $G_l = (-)^l Z_{L_{l-1}} Z_{S_l} Z_{L_l}$, it commutes with every Z error, so the stabilizer group generated by the Gauss' law will never be able to correct Z errors. However, if an X error happens on the site S_l, the operator G_l will give -1 eigenvalue, while if an X error happens on the link L_l, both G_l and G_{l+1} will give -1 eigenvalue. Indeed, by measuring G_{l-1}, G_l, and G_{l+1} we are able to correct every bit-flip error on L_{l-1}, S_l, or L_l, as shown in Table 1. This is an error-correcting code that, acting on N sites and N links (where every link and every site are one qubit each), encodes N logical qubits. We can say that it is a

$$[2N, N, 3] \tag{22}$$

classical error-correcting code, with $2N$ physical qubits, N stabilizers and so $2N - N$ logical qubits. Since we just proved that it can catch every single-qubit X error, it has distance 3.

In the 2 dimensional case, looking at Eq. (14), the Gauss' law becomes a weight 5 operator acting on 1 site and 4 links. Again, if only one Gauss' law gives -1 means that the error is on the site, while if two neighbouring Gauss' laws give -1 the error is on

the link they share. It is easy to generalize both the Gauss' law and the decoding to d dimensions as follows: the Gauss' law will always act on 1 site and all neighbouring links, and if only one Gauss' law gives -1 the error is on the site, while if two neighbouring Gauss' law give -1 the error is on the link they share.

As for the parameters of the code, let us remind that in a code with n physical qubits and $n-k$ stabilizers we have k logical qubits. Since we will always have dN links and N sites, we will have a total of $N+dN$ physical qubits and N stabilizers, which leads to dN logical qubits and the code can be written as

$$[N+dN, dN, 3] \tag{23}$$

We assume periodic boundary conditions, and so we need for this code the neighbouring sites we are considering to be different. In the $d=1$ case, every site has 2 neighbours, which means that we need at least $N=3$. For a general d, we need a grid of 3^d sites at least. □

We proved that the redundancy given by the Gauge symmetry is enough to build an error-correcting code able to correct one type of errors. Now, we will use it to build two different types of distance-3 quantum error-correcting codes by using concatenation.

Lemma 2. *Consider the classical distance-3 error-correcting code of Theorem 1, built on a d-dimensional lattice with $N+dN$) qubits and that uses the Gauss' law operators G_l as stabilizers. That code has parameters $[N+dN, dN, 3]$, $\forall N \geq 3^d$. Using concatenation, it is possible to build a CSS quantum error-correcting code with parameters*

$$[[N+dN+O(\log(N+dN)), dN, 3]] \quad \forall N \geq 3^d. \tag{24}$$

The proof of Lemma 2 is given in Appendix A. We now focus on a class of error-correcting codes that are not as optimal in terms of memory, but have improved locality for the stabilizer measurements. Let us define the quantum low-density parity check (QLDPC) codes as codes in which the number of qubits involved in each check and the number of checks acting on each qubit are both upper bounded by a constant [32]. We can then use this definition to see that the Gauss' law error-correcting code is then a CSS QLDPC code. We provide a formal statement of this observation while providing the parameters for the code below.

Lemma 3. *Consider the classical distance-3 error-correcting code of Theorem 1, built on a d-dimensional lattice with $N+dN$ qubits, and that uses the Gauss' law operators as stabilizers. This code has parameters $[N+dN, Nd, 3]$, for $N > 3^d$. Using concatenation, it is possible to build a CSS QLDPC code with parameters*

$$[[3(N+dN), Nd, 3]] \quad \forall N > 3^d \tag{25}$$

Proof. From Theorem 1, the error-correcting code built from the Gauss' law operators can detect and correct single-qubit bit-flip error (X errors). To transform it into a distance-3 quantum error-correcting code, it has to be concatenated with another error-correcting code able to correct every single-qubit phase-flip error (Z errors). The simplest and smallest phase-flip code is the 3 qubit repetition code, which takes 3 physical qubits and encodes them into one logical qubit. The stabilizers for this code are X_1X_2 and X_2X_3 and

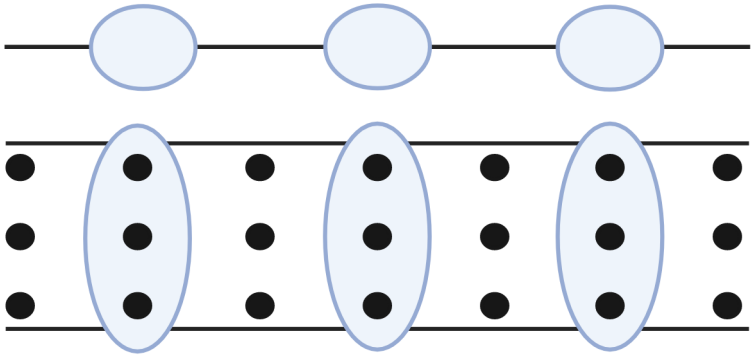


Figure 4: Graphical representation of the additional qubits needed for the concatenation. This figure shows how we can associate every site and link to 3 qubits now, instead of being one qubit each as before.

we have two possibilities to concatenate it to the Gauss' law code. One possibility is to follow Ref. [18] and transform every site and link of Fig. 3 into logical qubits so that every link and every site will correspond to 3 physical qubits. Then, those logical qubits already protected against phase-flip errors will be used as physical qubits for the bit-flip Gauss' law code. To do that, we have to substitute the Z operations in the Gauss' law, with the $\bar{Z}$ logical operation on the logical qubit of the phase-flip code. Since the logical operations for the repetition code are $\overline{X} = X_i, \overline{Z} = Z_1 Z_2 Z_3$, the Gauss' operator G_l becomes a weight 9 operator. A representation of the qubits layout and of this code are shown in Fig. 4 and Fig. 5 respectively.

The second possibility is the other order of concatenation, by building the Gauss' law code before and then repeating it to build the repetition code on top. In this case, we will have 3 copies of the Gauss' law codes, with logical operations $\overline{Z}_l = Z_{L_l}$ and $\overline{X}_l = X_{S_l} X_{L_l} X_{S_{l+1}}$. Then we have to take the stabilizers of the repetition code and extend them from $X_1 X_2$ to $\overline{X}_1 \overline{X}_2$ and so on, as we did for the Gauss' law in the previous concatenation. A graphical representation of this code is shown in Fig. 6. Both concatenations have the same number of qubits (3 for every link and site in the system) and the same logical operations which are $\overline{X}_l = X^i_{S_l} X^j_{L_l} X^k_{S_{l+1}}$ with $i, j, k \in \{1, 2, 3\}$ label the 3 qubits of a link/site, and $\overline{Z}_l = Z^1_{L_l} Z^2_{L_l} Z^3_{L_l}$. The major difference between the two concatenations is the maximum weight of the stabilizers (9 and 6) and the fact that, with the first concatenation,

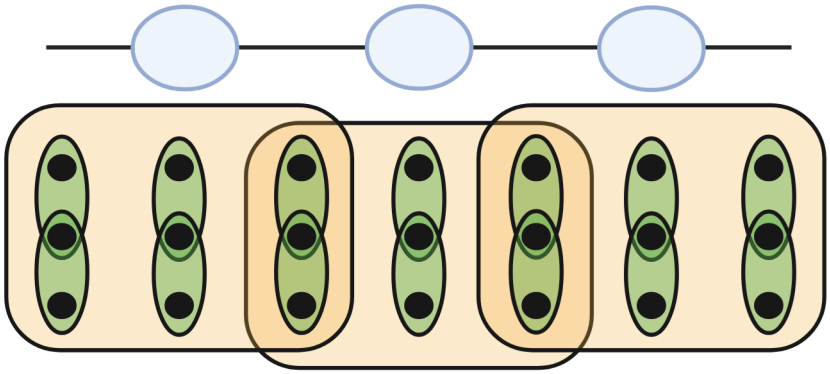


Figure 5: Graphical representation of the quantum error-correcting code. The green circle represents X stabilizers, which are all of weight 2. The orange area instead represents the weight 9 Gauss' law stabilizer. On top, the 1-dimensional system is represented to show explicitly that every link and site is in reality made of the 3 physical qubits below it.

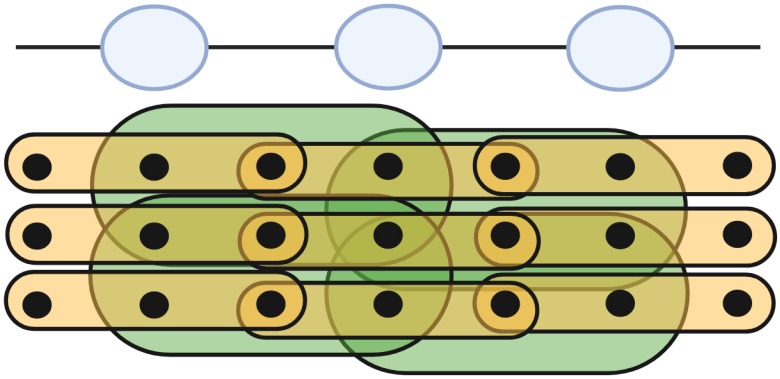


Figure 6: Graphical representation of the quantum error-correcting code with the second possible concatenation. The green areas represent the weight 6 X stabilizers of the phase-flip repetition code. The orange areas instead represent the weight 3 Gauss' law stabilizers. On top, the 1-dimensional system is represented to show explicitly that every link and site is in reality made of the 3 physical qubits below it.

we were able to correct 1 phase-error for every link and every site independently, while the second concatenation requires only 1 phase-flip error in a larger patch of the system. However, while the first concatenation has a very large patch in which only one bit-flip error can occur, the second concatenation has more symmetry between the errors. So one could decide which concatenation to use according to the error model of the hardware being used. Both concatenations give a distance 3 quantum error-correcting code, and in both cases the maximum weight (and the locality) of stabilizers is bounded.

In the 2-dimensional case nothing changes. The Gauss' law acts on a site and 4 links instead of 2, but the code construction remains the same because we can still use the 3 qubits repetition code and concatenate it to the Gauss' law code. Even if the Gauss' law operator changes slightly, we can still encode every link and site in the repetition code and use those logical qubits for the Gauss code, or we can make 3 copies of the Gauss code and use on top of that the repetition code. This time, the maximum weight of operators for the two concatenations is respectively 15 and 6. The 2 dimensional system is shown in Fig. 7.

For a generic value of d, we already discussed how to generalize the classical Gauss' law code with Theorem 1. To generalize the full quantum error-correcting code we have to describe how the 3-qubit phase-flip code behaves. This depends on the choosen concatenation: if we build the phase-flip as first code, we encode every link and site in a logical qubit already protected against phase-flip errors, and we only need to build on top of that the Gauss' law code as described in Theorem 1. If instead we build the Gauss' law code first, we need, as before, 3 copies of it, and as one can see from the 2-dimensional case we will have 2 phase-flip stabilizers (that are weight 6 operators) on every link. For completeness, let us report the maximum weight operators for the different concatenations: for the first concatenation, the operator with higher weight will always be the Gauss' law with weight $3(2d+1)$ (corresponding to 3 qubit for the site and 3 qubits for every link which are 2 for each dimension). For the second concatenation, the weight of the Gauss' law operator will be $2d+1$ (with only one qubit for the site and every link), and the maximum weight will be the max between $2d+1$ and 6 which is the weight of the phase-flip stabilizers. □

In the 2-dimensional case we need to measure 5 Gauss' law operators to fully correct a site and its 4 links. This means that, for our code to work, we have to assume that only 1 error happens at a time in those 5 sites and the corresponding 16 links. This is a big

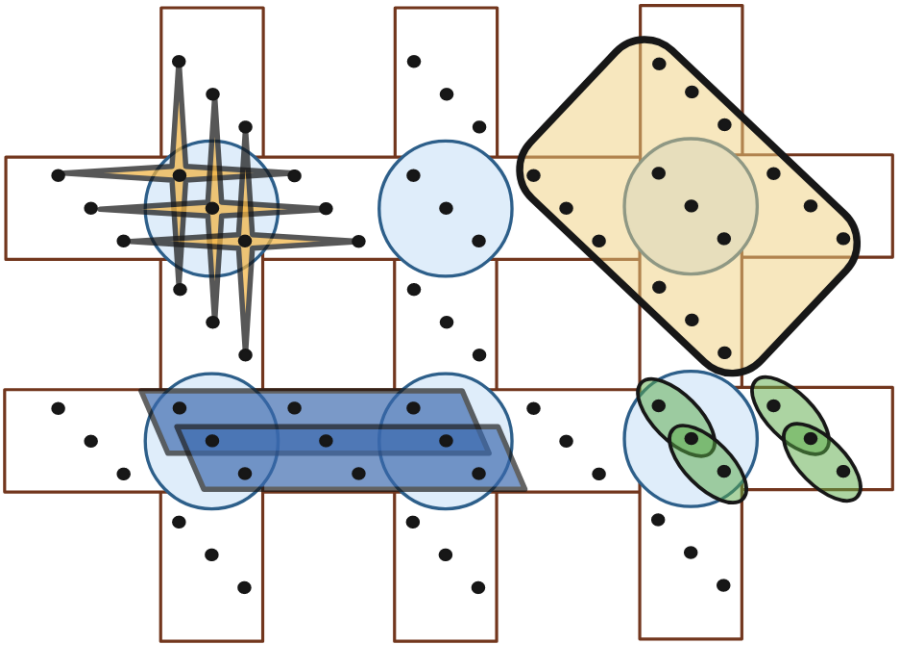


Figure 7: Graphical representation of the stabilizers and logical operations on a 2-dimensional lattice. The first concatenation leads to the orange rectangles, which is the Gauss' law (weight 15 operator), and the green ellipses that are the phase-flip stabilizers. The second concatenation instead leads to 3 Gauss' law per site, depicted as the yellow stars, and the blue parallelograms that are the phase-flip weight 6 stabilizers.

assumption because we would need a quite low error rate, but we can relax the assumption. Once we measure a Gauss' law and it gives -1, to know whether the error is on a specific link we need to measure a neighboring Gauss' law extending the measured system to 1 site and other 3 links. To avoid it, we can add 3 qubits, corrected with a phase-flip repetition code, and add the ZZ stabilizer that measures the parity between this additional logical qubit and the already existing link. This procedure is shown in Fig. 8, underlying the qubits that an additional Gauss' law would bring into the considered patch, compared to the single additional link. Of course, this would require more qubits for the error-correcting procedure, but if the assumption is that we can have only 1 error at a time for every K qubits, we can adjust K according to the error rate in this way. If we add no qubits, we ask for only 1 error in 63 qubits, corresponding to 5 sites and their 16 links (where every site and link corresponds to 3 qubits), while if we double every link, we are asking for only 1 error in 4 doubled links and 1 site for a total 27 physical qubits.

Fault-tolerant measurement of error syndromes can be implemented using the "logical to physical CNOT" explained in [18], and one flag qubit. With this observation in place, we see that we can perform a fault-tolerant implementation of the syndrome measurements. Further, the fact that the code is a CSS code will be invaluable later when we discuss how to simulate dynamics within these codes on a quantum computer.

4 Hamiltonian in terms of logical operations

Above we showed how we can construct quantum error-correcting codes using the Gauss' law checks. However, in order to simulate lattice Gauge theories within the quantum error-correcting code we need to provide a representation of the Hamiltonian expressed in terms of the corresponding logical operations. In this section, we will rewrite the Hamil-

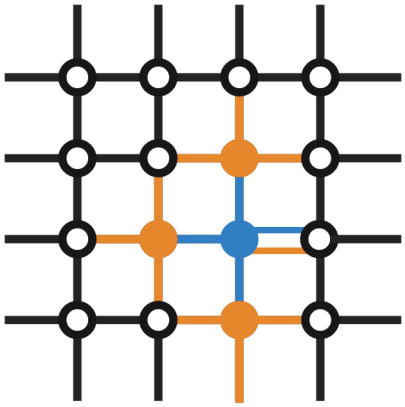


Figure 8: Graphical representation of the patch of the system we need to measure to correct 1 site and its 4 links. We start by measuring the Gauss' law on that site, and then we have to check if the error is on the site or a link. To do that, for every link we can either measure another Gauss' law, adding a site and 3 links (12 qubits in total), or add 3 qubits and measure the parity between the existing link and the additional one.

tonian in terms of the logical operations of the Gauss' law error-correcting code described in Theorem 1. Then, we will be able to define new creation and annihilation operators and write the Hamiltonian in terms of these new degrees of freedom. This will be equivalent to integrating fermions on sites, and we will see that the new degrees of freedom will be hardcore bosons.

So, let us start by writing the Hamiltonian in terms of the logical operations, reducing the degrees of freedom.

Theorem 4. *Let H be the Hamiltonian of Eq. (1) with the gauge group $\mathbb{Z}_2$. It is always possible to write H in terms of the logical operations of the error-correcting code of Theorem 1. By doing so, the degrees of freedom in the Hamiltonian are reduced from $N + dN$ to dN.*

Here we will discuss only the results for the 1 dimensional case, while the proof of Theorem 4, as well as the complete calculation for the 1 and 2 dimensional cases, are reported in Appendix B.

Recall the 1 dimensional Hamiltonian of Eq. (1)

$$H = m\sum_l (-1)^l \psi_l^\dagger \psi_l + \epsilon \sum_l (\psi_l^\dagger Q_l \psi_{l+1} + \psi_{l+1}^\dagger Q_l^\dagger \psi_l) + 2\lambda_E \sum_l P_l \tag{26}$$

With a $\mathbb{Z}_2$ symmetry we already said in the first chapter that $P_{L_l} = Z_{L_l}$ and $Q_{L_l} = X_{L_l}$. As for the fermionic operators, we can use the Jordan-Wigner representation, so that we can write the Hamiltonian in terms of the operations Z and X on links and sites:

$$H = \frac{m}{2}\sum_l (-1)^l (1 - (-1)^l Z_{S_l}) + \frac{\epsilon}{2}\sum_l \left(1 + Z_{S_l} Z_{S_{l+1}}\right) X_{S_l} X_{S_{l+1}} X_{L_l} + 2\lambda_E \sum_l Z_{L_l} \tag{27}$$

Remember that, due to the staggered fermions representation, the operator $|0\rangle\langle 1|$ is the annihilation operator for even sites while it is the creation operator for odd sites. Now, recall the logical operations of the error-correcting code of the last chapter and the Gauss' law operators:

$$\overline{X}_l = X_{S_l} X_{L_l} X_{S_{l+1}} \quad \overline{Z}_l = Z_{L_l} \quad G_l = (-1)^l Z_{L_{l-1}} Z_{S_l} Z_l \tag{28}$$

Here we are neglecting the presence of the phase-flip code, but the only difference with it is that every Z operation would be a weight 3 logical $\bar{Z}$. From here, using the relations of Eq. (28), we can express the Hamiltonian in terms of logical operations only:

$$\begin{aligned} H &= \frac{m}{2}\sum_l (-1)^l (1 - G_l \overline{Z}_{l-1}\overline{Z}_l) + \frac{\epsilon}{2}\sum_l \left(1 - G_l G_{l+1}\overline{Z}_{l-1}\overline{Z}_{l+1}\right)\overline{X}_l + 2\lambda_E \sum_l \overline{Z}_l \\ &= \frac{m}{2}\sum_l (-1)^l (1 - \overline{Z}_{l-1}\overline{Z}_l) + \frac{\epsilon}{2}\sum_l \left(1 - \overline{Z}_{l-1}\overline{Z}_{l+1}\right)\overline{X}_l + 2\lambda_E \sum_l \overline{Z}_l \end{aligned} \tag{29}$$

Note that in the second line the Gauss' laws have been simplified since they correspond to the logical identity. This form of the Hamiltonian is the one that Theorem 4 states to exist and the degrees of freedom have been reduced from $2N$ to N as expected. Apart from the explicit staggered mass term, this manifestly gauge invariant form of the Hamiltonian for a one-dimensional $\mathbb{Z}_2$ theory coupled to spinless fermions has been proposed already in Ref. [24] where it was derived a Majorana representation for the fermions. Our construction is however simpler and allows for straightforward generalizations to higher spatial dimensions.

Now that we have the Hamiltonian in term of the logical degrees of freedom, we can introduce bosonic degrees of freedom and write the entire Hamiltonian in terms of those.

Corollary 5. *Let H be the Hamiltonian of Eq. 1 with the gauge group $\mathbb{Z}_2$. It is possible to write H in terms of only bosonic degrees of freedom.*

Proof. Let us start from the Hamiltonian in the form given by Theorem 4, such that the Hamiltonian H is only a function of X_i^k and Z_i^k ($i \in [1, N]$ labels sites in the lattice, while $k \in [1, d]$ labels the different directions in the lattice). In general, the Hamiltonian will be written in terms of $2dN$ operators, that correspond to dN degrees of freedom (for a d dimensional lattice with N sites). Now, let us define a set of bosonic degrees of freedom as follows:

$$\phi_i^k = \frac{1}{2}\left(1 - Z_i^k\right) X_i^k \qquad (\phi_i^k)^\dagger = \frac{1}{2}\left(1 + Z_i^k\right) X_i^k \tag{30}$$

It is easy to prove the commutation relations to see that those operators are the creation and annihilation operators corresponding to hardcore bosons:

$$\begin{aligned} &\{\phi_i^k, \phi_i^k\} = \{\phi_i^{k\,\dagger}, \phi_i^{k\,\dagger}\} = 0 \qquad \{\phi_i^k, \phi_i^{k\,\dagger}\} = 1 \\ &[\phi_i^k, \phi_j^l] = [\phi_i^{k\,\dagger}, \phi_j^{l\,\dagger}] = [\phi_i^k, \phi_j^{l\,\dagger}] = 0 \qquad \forall i \neq j, \forall k \neq l \end{aligned} \tag{31}$$

Then, we can invert the relation and isolate operators that appear in the Hamiltonian:

$$Z_i^k = 2\phi_i^{k\,\dagger}\phi_i^k - 1 \quad X_i^k = \phi_i^{k\,\dagger} + \phi_i^k \tag{32}$$

Now those operators can be substituted in the Hamiltonian H and in this way we find the Hamiltonian written in terms of only hardcore bosonic degrees of freedom. □

As before, we now show the result for the 1 dimensional case, but the full calculations both for the 1 dimensional and 2 dimensional cases are shown in Appendix B. An advantage of the construction is that this derivation works for every boundary condition (by adapting the corresponding Gauss' laws) and every spatial dimension d of the lattice, which means that we will always be able to write the Hamiltonian in terms of hardcore bosonic degrees

of freedom, keeping a local Hamiltonian (without all-to-all interactions). In 1 dimension we define the hardcore bosonic operators:

$$\phi_l = \frac{1}{2}\left(1 - Z_l\right) X_l \qquad \phi_l^\dagger = \frac{1}{2}\left(1 + Z_l\right) X_l \tag{33}$$

Then, by defining the number operator of bosons as $N_l = \phi_l^\dagger \phi_l$, we can write the Hamiltonian in terms of these new degrees of freedom (full calculations in Appendix B):

$$H = 2m \sum_l (-1)^{l+1} N_{l-1} N_l + \epsilon \sum_l (N_{l-1} - N_{l+1})^2 (\phi_l + \phi_l^\dagger) + 2\lambda_E \sum_l (2N_l - 1) \ , \tag{34}$$

where we used the fact that $N_l = N_l^2$ for hard-core bosons.

As mentioned before, this form of the Hamiltonian underlines that sites have been integrated out since we used Gauss' law to write everything only as a function of the bosonic degrees of freedom. In Appendix B we show the full calculations for the 2 dimensional case following the same procedure. In the same appendix, we also show another possible choice of logical operations which is non-local and does not lead to bosonic degrees of freedom but could be interesting for other reasons, since it could eliminate some non-locality given by the Jordan-Wigner representation. It is important to notice that there are two notions of locality here: the first one, of the physical interaction, is the one we claim to preserve. When integrating out matter sites, the new interaction are smeared out from the use of Gauss' law but are still spatially local. The second notion of locality is the locality of the encoded operators in the qubit space used for the embedding. In particular, the encoding of fermionic degrees of freedom into qubit could lead to non-local qubit operators as is the case with the Jordan-Wigner mapping used here. This non-locality is however independent of the particular gauge theory used, and in fact can be present even when there are no gauge bosons. Our procedure for integrating out the matter sites using Gauss' law preserves the qubit locality of the underlying qubit Hamiltonian in a way that is independent of the particular fermionic encoding. This means that, by using a different encoding from Jordan-Wigner, it could be possible to have also a local encoded Hamiltonian. In summary, the locality of the logical operations in our error correcting code is controlled only by the Gauss' law which is local, while the locality of the Hamiltonian of the system depends on the fermionic encoding employed and is preserved by our construction.

There are other ways to transform fermionic degrees of freedom into hardcore bosons, as done e.g. in Refs.[33, 23], and starting from a Hamiltonian already written in terms of hardcore bosons could simplify the 2 dimensional case since it could avoid the non-local Pauli strings of Jordan-Wigner.

5 Time evolution

In previous sections, we built an error-correcting code exploiting the Gauss' Law. However, having an error-correcting code is not enough, if it cannot be used to perform operations. In this section, we will show how that code can be used to perform fault-tolerant operations on the system. In particular, we will provide a method to perform a universal gate set by state-injection, and methods to do a fault-tolerant time evolution of the Hamiltonian presented in the first section by using the error-correcting code described by Lemma 3. First, the Quantum Signal Processing (QSP) algorithm [26, 27] will be used to prove that, given an ancilla register with a logarithmic number of (logical) qubits,

all non-Clifford operations can be performed on the ancilla qubits, applying only Clifford operations on the error-correcting code. Then, we will discuss a method that uses product formulas, providing a fault-tolerant implementation and proving that a time evolution of H with product formulas can be performed using at most 3 logical ancilla qubits (Theorem 6).

5.1 Quantum Signal Processing

Let us consider a time evolution via the QSP algorithm. We will use the linear combination of unitaries (LCU) algorithm [34] to implement the Hamiltonian using an ancilla register. Then, from LCU to QSP one only needs to add another ancilla qubit, but for now, let us consider LCU only. For simplicity, let us look at the 1-dimensional Hamiltonian, and recall H in terms of the logical operations from Theorem 4:

$$H = -\frac{m}{2}\sum_{l=1}^{N}(-1)^l Z_{l-1}Z_l + \frac{\epsilon}{2}\sum_{l=1}^{N}X_l - \frac{\epsilon}{2}\sum_{l=1}^{N}Z_{l-1}Z_{l+1}X_l + 2\lambda_E\sum_{l=1}^{N}Z_l \tag{35}$$

To use the LCU algorithm, we need the normalized Hamiltonian $\tilde{H} = H/(N\eta)$ where N is the number of links in the system and $\eta = \frac{m}{2} + \epsilon + 2\lambda_E$:

$$\tilde{H} = \frac{1}{N}\sum_{l=1}^{N}\left(\frac{m}{2\eta}(-1)^{l+1}Z_{l-1}Z_l + \frac{\epsilon}{2\eta}X_l + \frac{\epsilon}{2\eta}(-Z_{l-1}Z_{l+1}X_l) + \frac{2\lambda_E}{\eta}(-Z_l)\right) \tag{36}$$

In general, considering the d dimensional lattice with N sites, we can always write the Hamiltonian as a sum over the dN degrees of freedom (represented in our code by the links of the lattice) and a sum over the K different operators with their coupling constants:

$$\tilde{H} = \frac{H}{\eta dN} = \frac{1}{dN}\sum_{l=1}^{dN}\left(\sum_{k=0}^{K}\frac{\eta_k}{\eta}O_{k,l}\right) \tag{37}$$

where $\eta = \sum_k \eta_k$. Note that we included the minus signs inside the operators so that all coupling constants are positive. From here, we can define the prepare and select operators:

$$\text{PREP}\,|0\rangle_a = \left(\frac{1}{\sqrt{2^n}}\sum_{l=0}^{2^n-1}|l\rangle_{a_1}\right)\otimes\left(\sum_{k=0}^{K}\sqrt{\frac{c_k}{C}}\,|k\rangle_{a_2}\right) \tag{38}$$

$$\text{SEL} = \sum_{k=0}^{K}|k\rangle_{a_2}\langle k|_{a_2}\left(\sum_{l=0}^{dN-1}|l\rangle_{a_1}\langle l|_{a_1}\otimes O_{k,l} + \sum_{l=dN}^{2^n-1}|l\rangle_{a_1}\langle l|_{a_1}\otimes\mathbb{I}\right) \tag{39}$$

The a subscript denotes the ancilla register, divided into the a_1 and a_2 registers, while $n = \lceil\log_2(dN)\rceil$ is the smallest integer such that $2^n \geq dN$. In general, we should use dN instead of 2^n in the prepare operator, but having a power of 2 makes everything much easier. So we use the smallest power of 2 larger than dN, paying the price of implementing $\frac{dN}{2^n}\tilde{H}$ instead of just $\tilde{H}$. Since $\frac{1}{2} < \frac{N}{2^n} < 1$, it means that we will have to at most fator of two in the normalization.

As for the size of the ancilla registers, $a_2 = \lceil\log_2(K)\rceil$ is a small register (in the 1 dimensional case it is only 2 qubits, while in the 2 dimensional case it will be 3 qubits), while a_1 has the number of qubits equal to n (which scales logarithmically in the number of sites of the system). The prepare operator is easy to do since it will be made of n Hadamards applied on a_1, while on a_2 we need arbitrary state preparation, but we do not expect the size of this register to grow with the system size.

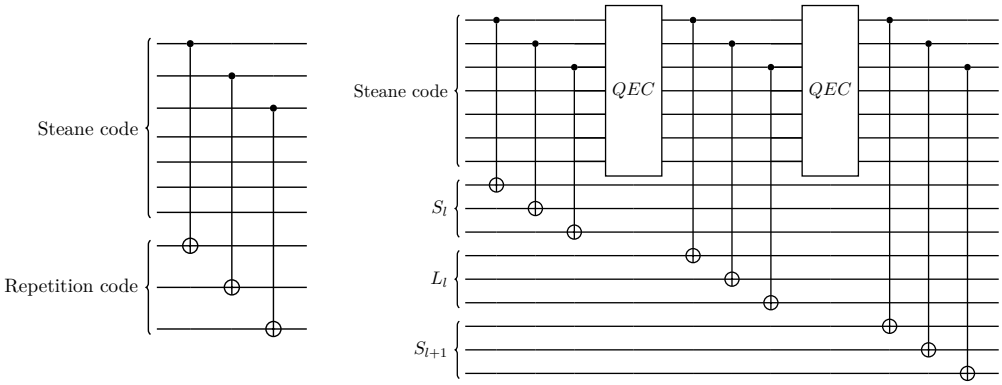


Figure 9: On the left-hand side, the implementation of a logical CNOT between the Steane code and the phase-flip 3-qubits repetition code, which is transversal. On the right-hand side, the fault-tolerant implementation of a logical $X_l = X_{S_l} X_{L_l} X_{L_{l+1}}$.

The select operator instead is more complicated, and we can estimate its complexity in terms of the number of Toffoli gates it requires. Given a circuit made of NOT, CNOT, and Toffoli (a so-called reversible NCT circuit), one can show that the lower bound to the number of Toffoli gates required to implement the select circuit of Equation 39 is $2^n - n - 1$ ([35], Lemma 4). The upper bound to the number of Toffoli gates instead is given by $1.5 \times 2^n - 4$ ([36], Lemma G.7), using $n - 1$ ancilla qubits.

We can now use the LCU oracles defined above to perform time-evolution using QSP by adding an additional qubit to the ancilla register which will be used to perform rotations and a controlled version of the SEL operation. All Toffoli gates and non-Clifford operations are done on the ancilla registers, which will have their own error-correcting code, while on our system will be applied only the controlled-$O_{k,l}$ operators. By definition, the $O_{k,l}$ are Pauli operators and the control will be a single qubit of the logical ancilla register, which means that we need to apply to the lattice Clifford operations only.

To ensure that those Clifford operations are fault-tolerant, let us assume that the ancilla qubit from which we control all the controlled-$O_{k,l}$ gates is in the 7-qubit Steane code [37, 8]. This code has logical operations $\overline{X} = X_1 X_2 X_3$ and $\overline{Z} = Z_1 Z_2 Z_3$, which are the same logical operations of the phase-flip repetition code used on the lattice. Since they are both CSS codes with the same logical operations, it means that the CNOT is transversal between the two codes, as shown in the left panel of Figure 9. So, if $O_{k,l}$ is a weight 1 operator for the Gauss' Law code, it is transversal and fault-tolerant by definition on the repetition code. If it has higher weight, it means that we have to repeat several times the circuit on the left panel of Figure 9, and if an X error happens on the control qubit it will spread on the system. To prevent this, the most trivial solution is to perform an error-correcting cycle on the ancilla register between every controlled logical operation, as shown in the right panel of Figure 9. An alternative, that could be more efficient, is to use flag qubits to catch specific errors that are problematic [38, 39].

5.2 Trotterization

In the last section, we showed that, using a logarithmic number of ancilla qubits, we are able to restrict the application on our error-correcting code of only Clifford operations, and perform a fault-tolerant time evolution of the Hamiltonian. Now, we will extend this argument by proving that not only this is possible, but also that the same can be achieved with a constant number of ancilla qubits. Of course, both methods have their

own advantages, and there could be cases where Trotter-based simulations have a better asymptotic scaling than QSP (see e.g. [40, 41]).

Theorem 6. *Let H be the Hamiltonian of Equation 1, and consider the lattice encoded in the quantum error-correcting code described in Lemma 3. When performing the time evolution of H using ancilla qubits, it is possible to perform all non-Clifford operations on the ancilla qubits and apply to the lattice Clifford operations only. This can be done with at most 3 ancilla qubits encoded in the 7-qubits code.*

Proof. Let us consider the trotterization method to perform the time evolution of the Hamiltonian. By using this method, and remembering that the Hamiltonian contains only Pauli operators, we will need to apply to the system exponentials like e^{itP} where P is a Pauli operator. Note that $e^{itP} = \cos(t) + i\sin(t)P$ for every P in the Pauli group, and we can implement this sum by using LCU. To do so, consider an ancilla in the 7-qubit code, already prepared in the superposition $\cos(t)|0\rangle - i\sin(t)|1\rangle$. We can use this ancilla to implement e^{itP} or e^{-itP} with probability $1/2$ by using the circuit in the right panel of Fig. 10 (where the success, and the application of the right exponential, will happen when the measurement outcome will be $|0\rangle$). The ancilla qubit is considered to be encoded in the 7-qubit code since we already showed before how to do perform the controlled-pauli operator in a falut-tolerant way using that error-correcting code. Then, in order to prepare the ancilla in the right superposition, we will need to perform state injection using another ancilla encoded in the 7 qubit code as well. The circuit to perform state injection is shown in the left panel of figure 10.

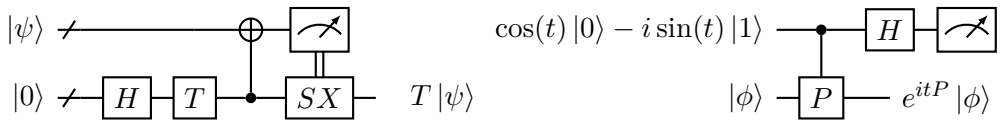
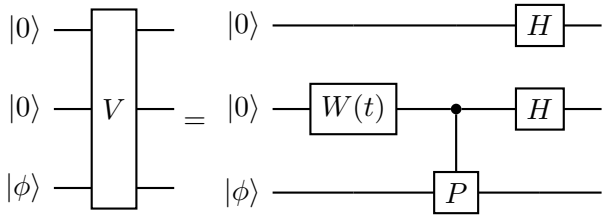


Figure 10: On the left-hand side the circuit performs state-injection on a 7-qubit code. On the right-hand side, the circuit applies e^{itP} on $|\phi\rangle$ using an ancilla in the right superposition. In both circuits every operation is transversal. The second circuit has a success probability of $1/2$.

With the circuit we described, we can implement every exponential of the type e^{itP} with success probability of $1/2$, but we can use oblivious amplitude amplification [42] to make sure it always succeeds. To apply it, first we have to lower the success probability to $1/4$, which can be done by adding a qubit in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and by defining success the measured state to be $|00\rangle$ on the two ancilla qubits. So, let us define the V operator as:



(40)

where the gate $W(t)$ performs the following rotation:

$$W(t)|0\rangle = \cos(t)|0\rangle - i\sin(t)|1\rangle \tag{41}$$

So, as mentioned before, the application of the gate V will success with probability $1/4$ when we measure the two ancilla qubits. Now, let us define the following projection and reflection operators:

$$\begin{aligned} \Pi_0 &= |00\rangle\langle 00| \otimes \mathbb{1} \\ R &= 2\Pi_0 - \mathbb{1} \end{aligned} \tag{42}$$

where the tensor product is ment to distinguish between the two ancilla qubits, and the system encoded in the state $|\phi\rangle$. Note that the reflection operator R is nothing but a control Z gate on the two ancilla qubits (which means it is also hermitian). Finally, we can define the operator for oblivious amplitude amplification $S = -VRV^\dagger R$. With those definitions, if we apply the operator $SV = -VRV^\dagger RV$, we will apply on the system (the $|\phi\rangle$ register) the right exponential e^{itP} with probability 1 (and using a single iteration of the operator S).

In the end, we can implement every exponential coming from the trotterization method (fault-tolerantly) using at most 3 logical ancilla qubits encoded in the 7 qubit Steane code. Two are the ones of Equation (40), while the last one is the one needed to perform state injection and apply the gate $W(t)$. □

We just showed how one can apply e^{itP} on the system fault-tolerantly for every P in the Pauli group. However, even if we used it only to perform the time evolution of the Hamiltonian, this procedure can be used to achieve a universal gate set on our error-correcting code.

Theorem 7. *A universal gate set can be performed on the system encoded in the error-correcting code of Lemma 3 using at most* 3 *logical ancilla qubits initialized in the* 7*-qubit code.*

Proof. Pauli operations are transversal as well as the CNOT, since the code is CSS. Then, since we are able to apply on the system $e^{i\theta P}$ for every angle θ and Pauli operator P, we can use this to apply operations that are not already transversal. It is trivial to perform arbitrary Z rotations via their definition as $R_Z(\theta) = e^{i\theta Z}$, while we can perform the Hadamard gate by using the definition $H = XR_y(\frac{\pi}{2})$ and $R_y(\theta) = e^{i\theta Y}$. □

6 Conclusions

We extended the original idea of Ref. [18] which builds quantum error-correcting codes by exploiting the redundancy given by the Gauge invariance in a Lattice Gauge Theory in several ways. Our first new contribution is the extension of the original proposal for a $\mathbb{Z}_2$ gauge theory in one and two dimensions to arbitrary spatial dimensions while being agnostic on the choice of the phase-flip error-correcting code employed to make the whole protocol fault-tolerant. This leads to either a logarithmic or linear overhead in the number of qubits to perform the error correction, depending whether we want local checks or not. We have then shown that by encoding the Hamiltonian with this error-correcting code, H can be written in terms of the logical operations of the error-correcting code itself, which is equivalent to integrating out fermions. Using this procedure, it becomes easy to transform the original fermionic Hamiltonian into a hardcore-bosonic Hamiltonian.

The technique of writing the Hamiltonian of a lattice gauge theory in terms of the logical operations allowed in the corresponding Gauss' Law code is a powerful and simple strategy to find dual theories to the original model. Then, we have provided an explicit fault-tolerant algorithm to perform a universal set of operations on our Gauss' Law code

and use it to show how to perform Hamiltonian simulation. We described in detail simulations using both the Quantum Signal Processing and Trotterization methods, showing that only a logarithmic and constant number of ancilla qubits is required for the two methods respectively. We have achieved that by showing that, on the error-correcting code exploiting the Gauss' law, every operation needed for Hamiltonian simulation is Clifford and can be done transversally. If the computation will require non-Clifford operations, they will be always done on an ancilla register, that is assumed to have it's own error-correcting code.

Several extensions of the present work are possible including a complete characterization of the Gauss' Law code for larger discrete Abelian groups such as $\mathbb{Z}_N$ as well as the design of error-correcting codes tailored for non-Abelian discrete gauge groups. Non-Abelian theories cannot generate a stabilizer group, which is assumed to be an Abelian group, and so one way to solve the problem could be non-stabilizer error-correcting codes. As for increasing the gauge group dimension from $N = 2$ to larger values, one can either add qubits to every link of the lattice, or work with qudits-links. The qudit case appears easier form the theoretical point of view, since the Gauss' Law remains in the generalized Pauli group. However, it is not clear what is the real noise model for multi-level systems, and so many questions have still to be addressed even in this case.

Acknowledgments

We would like to thank P. Hauke, J. Mildenberger and A. Rajput for many useful discussions about the topics discussed in this work. A.R. is funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101080086 NeQST.

A Proof of Lemma 2

Proof of Lemma 2. Consider the Quantum Hamming codes [43] with parameters

$$[[2^r - 1, 2^r - 1 - 2r, 3]] \,, \tag{43}$$

with $r \geq 3$. These are CSS quantum error-correcting code made by a concatenation of 2 classical Hamming codes. If we consider $2^r - 1 - 2r \geq N + dN$, we can use the logical qubits of this code as physical qubits for the classical error-correcting code of Theorem 1 (that has parameters $[N + dN, dN, 3]$). Since this second code is classical, it corrects only one type of errors by definition and it cannot increase the distance of the Hamming code. Now, in order to choose r we need $2^r - 1 - 2r \geq N + dN$. We can take for instance

$$r = \log_2(N + dN + 1) + \log_2\left(1 + \alpha \frac{\log_2(N + dN + 1)}{N + dN + 1}\right) \,, \tag{44}$$

so that the condition we need is

$$(\alpha - 2)\log_2(N + dN + 1) - 2\log_2\left(1 + \alpha \frac{\log_2(N + dN + 1)}{N + dN + 1}\right) \geq 0 \,. \tag{45}$$

We can make sure this is satisfied for any $N \geq 1$ by using $\alpha = 6$. The total number of physical qubits is thus

$$2^r - 1 = N + dN + 1 + \alpha \log_2(N + dN + 1) \,. \tag{46}$$

The Quantum Hamming code can be written as:

$$[[2^r - 1, N + dN, 3]] \tag{47}$$

and with the concatenation of the code from Theorem 1 we get a code with parameters:

$$[[2^r - 1, dN, 3]] \tag{48}$$

Finally, using the choice of r from Eq. (44) and $\alpha = 6$ together with Eq. (46) we can write the final parameters of the code as

$$[[N + dN + 6 \log_2(N + dN + 1), N + dN, 3]] \tag{49}$$

Furthermore, since the concatenation involves a CSS code and a classical code, X and Z stabilizers will never be mixed, so the final code will be again CSS.

□

It is possible to build more efficient codes with respect to the one described in the proof of Lemma 2. This is due to the fact that we do not need in general a full quantum error-correcting code to concatenate the code of Theorem 1. We could instead build a quantum error-correcting code by considering a classical Hamming code for Z errors, and the code of Theorem 1 for X errors. This will not modify the scaling but it could improve the qubit overhead needed. However, it could happen in this case that not every error syndrome is unique, and we could need some additional qubits ($O(1)$) to make every error syndrome unique.

An example of such a code is shown in Table 2, where we are considering a 2 sites 2 links system with periodic boundary conditions. In this case, we used a classical Hamming code to encode qubits in the phase-flip code, while for the bit-flip we used Gauss' law operators, and only 2 additional qubits to make every error syndrome of the bit-flip code unique.

B Bosonic encoding

In this section we report the proof of Theorem 4, as well as the full calculations to write the 1 and 2 dimensional Hamiltonian as a function of bosonic degrees of freedom integrating out the fermions. At the end of the section, we also show another possible choice for the logical operations of the error-correcting code, which is non-local and it's harder to deal with. However, we believe that those non-localities raising when writing H in terms of logical operations could cancel non-localities coming from Jordan-Wigner in the 2 dimensional case, or any way they could be advantageous in some cases.

B.1 Proof of Theorem 4

Proof of Theorem 4. Remember the Hamiltonian of Eq. (1):

$$H = H_M + H_{hop} + H_E + H_P \tag{50}$$

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}
S_1	X	1	X	1	X	1	X	1	X	1	X
S_2	1	X	X	1	1	X	X	1	1	X	X
S_3	1	1	1	X	X	X	X	1	1	1	1
S_4	1	1	1	1	1	1	1	X	X	X	X
$\bar{Z}_1 = A_0$	Z	Z	Z	1	1	1	1	1	1		
$\bar{Z}_2 = L_0$	1	Z	Z	Z	Z	1	1	1	1		
$\bar{Z}_3 = S_1$	1	1	1	Z	Z	Z	Z	1	1		
$\bar{Z}_4 = L_1$	1	1	1	1	1	Z	Z	Z	Z		
$\bar{Z}_5 = S_0$	1	1	Z	Z	1	1	Z	1	1		
$\bar{X}_1$	X	1	1	1	1	1	1	1	1		
$\bar{X}_2$	X	1	X	X	X	1	1	1	1		
$\bar{X}_3$	1	1	1	X	X	1	X	X	1		
$\bar{X}_4$	1	1	1	1	1	1	1	X	1		
$\bar{X}_5$	1	X	X	1	1	1	1	1	1		
G_1	1	Z	Z	1	1	1	1	Z	Z	1	1
G_2	1	Z	1	1	Z	Z	1	Z	Z	1	1
G_3	Z	1	1	Z	Z	1	1	1	1	1	1
G_4	1	1	1	Z	1	1	Z	Z	1	1	Z
G_5	1	1	1	Z	1	1	Z	1	Z	Z	1

Table 2: Explicit representation of the Hamming phase-flip code, using a 2 sites system. In this case, we need 2 additional bit-flip stabilizers to make the code distance 3. The S_i operators are the phase-flip stabilizers, while G_i labels the bit-flip stabilizers. G_i with $i \leq 3$ are built by multiplying logical operations as in the definition of the Gauss' law.

$$
\begin{aligned}
H_M &= m \sum_{\vec{l}} \sigma_{\vec{l}}\, \psi^\dagger_{\vec{l}} \psi_{\vec{l}} \\
H_{hop} &= \epsilon \sum_{\vec{l}} \sum_{\hat{k}\in\mathcal{D}} \sigma_{\vec{l},\hat{k}} \left(\psi^\dagger_{\vec{l}} Q_{\vec{l},\hat{k}} \psi_{\vec{l}+\hat{k}} + \psi^\dagger_{\vec{l}+\hat{k}} Q^\dagger_{\vec{l},\hat{k}} \psi_{\vec{l}} \right) \\
H_E &= \lambda_E \sum_{\vec{l}} \sum_{\hat{k}\in\mathcal{D}} \left(P_{\vec{l},\hat{k}} + P^\dagger_{\vec{l},\hat{k}} \right) \\
H_M &= \lambda_P \sum_p (W_p + W_p^\dagger)
\end{aligned}
\tag{51}
$$

Since the stabilizer group is generated by the Gauss' Law operator, the Hamiltonian will commute with it. So, the fact that the Hamiltonian can always be written in terms of the logical operations follows trivially since the Gauss' Law and the logical operations are a complete basis of the centralizer of the stabilizer group (set of elements of the Pauli group that commute with the every element of the stabilizer group).

Let us now define the Jordan-Wigner transformation [44]. It is a method to write the fermionic creation and annihilation operators in term of the Pauli matrices, preserving the commutation relations. First, we can define the creation and annihilation operators as the operators $|1\rangle\langle 0|$ and $|0\rangle\langle 1|$ respectively for even sites, while for odd sites we invert the definition. In terms of Pauli matrices we can write:

$$
a_{\vec{l}} = \frac{1}{2}\left(1 + Z_{\vec{l}}\right) X_{\vec{l}} \qquad a^\dagger_{\vec{l}} = \frac{1}{2}\left(1 - Z_{\vec{l}}\right) X_{\vec{l}}\,. \tag{52}
$$

Looking at the same site, those operators have the right commutation relations:

$$
\{a_{\vec{l}}, a_{\vec{l}}\} = \{a^\dagger_{\vec{l}}, a^\dagger_{\vec{l}}\} = 0 \qquad \{a_{\vec{l}}, a^\dagger_{\vec{l}}\} = 1\,. \tag{53}
$$

However, operators on different sites commute, while fermionic operator should anticommute. To adjust those relations, let us assume for simplicity a 1-dimensional lattice, we can define the proper fermionic creation and anihilation operators as follows:

$$\psi_l = \left(\prod_{i=0}^{l-1} -Z_i\right) a_l \qquad \psi_l^\dagger = \left(\prod_{i=0}^{l-1} -Z_i\right) a_l^\dagger \tag{54}$$

In higher spatial dimensions, the definition is straightforwardly generalized by defining an ordering so that we can go through every site with a single index $n(\vec{l})$, one for each of the N sites. In this case the definition of the fermionic operators becomes:

$$\psi_{\vec{l}} = \left(\prod_{k=0}^{n(\vec{l})-1} -Z_{\vec{l}_k}\right) a_{\vec{l}} \qquad \psi_{\vec{l}}^\dagger = \left(\prod_{k=0}^{n(\vec{l})-1} -Z_{\vec{l}_k}\right) a_{\vec{l}}^\dagger \,, \tag{55}$$

where $k = n(\vec{l}_k)$. With this definition, it is easy to show that these operators satisfy the right commutation relations:

$$\{\psi_{\vec{l}}, \psi_{\vec{m}}\} = \{\psi_{\vec{l}}^\dagger, \psi_{\vec{m}}^\dagger\} = 0 \qquad \{\psi_{\vec{l}}, \psi_{\vec{m}}^\dagger\} = \delta_{\vec{l},\vec{m}} \,. \tag{56}$$

Using the Jordan-Wigner construction and the $\mathbb{Z}_2$ symmetry we can write H as follows:

$$\begin{aligned} H = {} & \frac{m}{2} \sum_{\vec{l}} \sigma_{\vec{l}} (1 - (-1)^{|\vec{l}|} Z_{S_{\vec{l}}}) \\ & + \frac{\epsilon}{2} \sum_{\vec{l}} \sum_{\hat{k}\in\mathcal{D}} \sigma_{\vec{l},\hat{k}} O_{\vec{l},\hat{k}} \left(1 + Z_{S_{\vec{l}}} Z_{S_{\vec{l}+\hat{k}}}\right) X_{S_{\vec{l}}} X_{L_{\vec{l},\hat{k}}} X_{S_{\vec{l}+\hat{k}}} \\ & + 2\lambda_E \sum_{\vec{l}} \sum_{\hat{k}\in\mathcal{D}} Z_{L_{\vec{l},\hat{k}}} + 2\lambda_P \sum_p X_{p_1} X_{p_2} X_{p_3} X_{p_4} \end{aligned} \tag{57}$$

where $O_{\vec{l},\hat{k}}$ is the Pauli string given by Jordan-Wigner. The last term sums over all plaquettes, and the numbering convention is according to Fig. 1 as explained in the first chapter of the main text. Recall that, for a d-dimensional lattice, the logical operations of the error-correcting code of Theorem 1 are

$$\overline{Z}_{\vec{l},\hat{k}} = Z_{L_{\vec{l},\hat{k}}} \qquad \overline{X}_{\vec{l},\hat{k}} = X_{S_{\vec{l}}} X_{L_{\vec{l},\hat{k}}} X_{S_{\vec{l}+\hat{k}}} \tag{58}$$

while the Gauss' law operator, which for code-words is nothing but the logical identity, is

$$G_{\vec{l}} = (-1)^{|\vec{l}|} Z_{S_{\vec{l}}} \prod_{\hat{k}\in\mathcal{D}} Z_{L_{\vec{l},\hat{k}}} Z_{L_{\vec{l}-\hat{k},\hat{k}}} \tag{59}$$

From here we can write the Hamiltonian H in terms of these logical operations. First, note that X operators in the hopping term appear already in the form of the logical operation $\overline{X}_{\vec{l},\hat{k}}$, so we can simply substitute it. The Z operators acting on links they translate directly into logical $\overline{Z}_{\vec{l},\hat{k}}$, while for the Z on sites we use Gauss' law to write

$$Z_{S_{\vec{l}}} = (-1)^{|\vec{l}|} G_{\vec{l}} Z_{L_{\vec{l},\hat{k}}} Z_{L_{\vec{l}-\hat{k},\hat{k}}} = (-1)^{|\vec{l}|} Z_{L_{\vec{l},\hat{k}}} Z_{L_{\vec{l}-\hat{k},\hat{k}}} \,. \tag{60}$$

The plaquette operator can be obtained by multiplication of 4 logical operations, so that X operations on sites cancel and we get the 4 X on links. In this way we can write as logical operations every operator in the Hamiltonian. As for the degrees of freedom, we started with a degree of freedom for every link and one for every site (which are $N + dN$), while now the Hamiltonian is written in function of the logical qubits (which we know from Theorem 1 they are dN). □

B.2 Full calculation for the 1-dimensional case

Let us start from the Hamiltonian written in terms of logical operations

$$H = \frac{m}{2}\sum_l (-1)^l (1 - Z_{l-1}Z_l) + \frac{\epsilon}{2}\sum_l (1 - Z_{l-1}Z_{l+1})\, X_l + 2\lambda_E \sum_l Z_l \tag{61}$$

and the hardcore bosonic operators

$$\phi_l = \frac{1}{2}(1 - Z_l)\, X_l \qquad \phi_l^\dagger = \frac{1}{2}(1 + Z_l)\, X_l \tag{62}$$

We define the number operator $N_l = \phi_l^\dagger \phi_l = \frac{1}{2}(1 + Z_l)$ and note that it is idempotent ($N_l = N_l^2$). So the various terms in the Hamiltonian can be written as:

$$Z_l = 2N_l - 1 \tag{63}$$

$$\begin{aligned}\sum_l (-1)^l (1 - Z_{l-1}Z_l) &= \sum_l (-1)^l (1 - (2N_{l-1} - 1)(2N_l - 1)) \\ &= -4\sum_l (-1)^l N_{l-1}N_l + 2\sum_l (-1)^l N_l + 2\sum_l (-1)^{l-1} N_l \\ &= -4\sum_l (-1)^l N_{l-1}N_l\end{aligned} \tag{64}$$

$$\begin{aligned}1 - Z_{l-1}Z_{l+1} &= 1 - (2N_{l-1} - 1)(2N_{l+1} - 1) \\ &= 2(N_{l-1} + N_{l+1} - 2N_{l-1}N_{l+1}) \\ &= 2(N_{l-1} - N_{l+1})^2\end{aligned} \tag{65}$$

We note that the disappearance of the terms linear in N_l from the second term is a direct consequence of using Periodic Boundary Conditions, in more general settings boundary terms might still be present. In the end, the Hamiltonian will be:

$$\begin{aligned}H = {} & 2m\sum_l (-1)^{l+1} N_{l-1}N_l \\ & + \epsilon \sum_l (N_{l-1} - N_{l+1})^2 (\phi_l + \phi_l^\dagger) + 2\lambda_E \sum_l (2N_l - 1)\end{aligned} \tag{66}$$

B.3 Full calculations for the 2-dimensional case

Let us consider a lattice with N_x by N_y sites. We start by writing the Hamiltonian as a function of the Pauli matrices, explicitly taking into account the fact that now Jordan-Wigner has non-local Pauli strings:

$$\begin{aligned}H = {} & \frac{m}{2}\sum_{i,j} (-1)^{i+j}(1 - (-1)^{i+j} Z_{S_{i,j}}) \\ & + \frac{\epsilon}{2}\sum_{i,j} \left(1 + Z_{S_{i,j}} Z_{S_{i,j+1}}\right) X_{S_{i,j}} X_{L^x_{i,j}} X_{S_{i,j+1}} \\ & + \frac{\epsilon}{2}\sum_{i,j} (-1)^j P_{i,j} \left(1 + Z_{S_{i,j}} Z_{S_{i+1,j}}\right) X_{S_{i,j}} X_{L^y_{i,j}} X_{S_{i+1,j}} \\ & + 2\lambda_E \sum_{i,j,k} Z_{L^k_{i,j}} + 2\lambda_P \sum_{i,j} X_{L^x_{i,j}} X_{L^y_{i,j+1}} X_{L^x_{i+1,j}} X_{L^y_{i,j}}\end{aligned} \tag{67}$$

where $P_{i,j} = \left(\prod_{n=j}^{N_x} Z_{S_{i,n}}\right)\left(\prod_{n=0}^{j} Z_{S_{i+1,n}}\right)$ is the Pauli string given by vertical links. In these expressions we used, for $\vec{l} = (i,j)$, the following definitions to ease the notation

$$S_{\vec{l}} = S_{i,j} \qquad L_{\vec{l},(0,1)} = L^x_{i,j} \qquad L_{\vec{l},(1,0)} = L^y_{i,j} \,. \tag{68}$$

The logical operations are

$$\begin{aligned} Z^x_{i,j} = Z_{L^x_{i,j}} \qquad Z^y_{i,j} = Z_{L^y_{i,j}} \qquad Z_{S_{i,j}} = (-)^{i+j} Z^x_{i,j} Z^y_{i,j} Z^x_{i,j-1} Z^y_{i-1,j} \\ X^y_{i,j} = X_{S_{i,j}} X_{L^y_{i,j}} X_{S_{i+1,j}} \qquad X^x_{i,j} = X_{S_{i,j}} X_{L^x_{i,j}} X_{S_{i,j+1}} \end{aligned} \tag{69}$$

and substituting we get:

$$\begin{aligned} H = & \frac{m}{2} \sum_{i,j} (-1)^{i+j} (1 - Z^x_{i,j} Z^y_{i,j} Z^x_{i,j-1} Z^y_{i-1,j}) \\ & + \frac{\epsilon}{2} \sum_{i,j} \left(1 - Z^x_{i,j-1} Z^y_{i,j} Z^y_{i-1,j} Z^y_{i,j+1} Z^x_{i,j+1} Z^y_{i-1,j+1}\right) X^x_{i,j} \\ & + \frac{\epsilon}{2} \sum_{i,j} (-1)^j P_{i,j} \left(1 - Z^x_{i,j-1} Z^y_{i-1,j} Z^x_{i,j} Z^x_{i+1,j} Z^y_{i+1,j} Z^x_{i+1,j-1}\right) X^y_{i,j} \\ & - \lambda \sum_{i,j,k} Z^k_{i,j} + \frac{1}{\lambda} \sum_{i,j} X^x_{i,j} X^y_{i,j+1} X^x_{i+1,j} X^y_{i,j} \end{aligned} \tag{70}$$

This is the Hamiltonian in terms of the logical operations, that can be used to implement the time evolution. Then, we can define the hardcore boson operators in the same way as the 1-dimensional case

$$\begin{aligned} \phi^k_{i,j} &= \frac{1}{2}(1 - Z^k_{i,j}) X^k_{i,j} \\ \left(\phi^k_{i,j}\right)^\dagger &= \frac{1}{2}(1 + Z^k_{i,j}) X^k_{i,j} \end{aligned} \tag{71}$$

where $k = x, y$. Defining again $N^k_{i,j} = \phi^{k,\dagger}_{i,j} \phi^k_{i,j}$ as the number operator we have the following relations

$$Z^k_{i,j} = 2N^k_{i,j} - 1 \quad X^k_{i,j} = \phi^k_{i,j} + \phi^{k,\dagger}_{i,j} \tag{72}$$

Let us look at the mass term:

$$\begin{aligned} & \sum_{i,j} (-1)^{i+j} (1 - Z^x_{i,j} Z^y_{i,j} Z^x_{i,j-1} Z^y_{i-1,j}) = \\ & = \sum_{i,j} (-1)^{i+j} [1 - (2N^x_{i,j} - 1)(2N^y_{i,j} - 1)(2N^x_{i,j-1} - 1)(2N^y_{i-1,j} - 1)] \end{aligned} \tag{73}$$

All the other terms can be found by substituting into the Hamiltonian the relations of Equation 72. In this way, we will have a Hamiltonian in terms of the bosonic operators and the dependence on sites is implicit, contained in the logical operations.

B.4 Another choice of logical operations for the 1-dimensional case

The logical operations can be chosen differently, and here we give an example for the one-dimensional case. However, in general, the previous choice was the easiest to deal with because is local. Indeed, we will see that the choice we will explain now is highly non-local, and also the new degrees of freedom will no longer be bosons.

Enumerating sites and links from 1 to N, the logical operations are the following:

$$\begin{aligned} X_l &= X_{S_1} X_{S_l} \prod_{n=l}^{N} X_{L_n} \quad \forall l > 1, \qquad X_1 = \prod_{n=1}^{N} X_{L_n} \\ Z_l &= (-1)^l Z_{S_l} \quad \forall l > 1, \qquad Z_1 = Z_{L_1} \end{aligned} \tag{74}$$

which can be inverted using the stabilizers which will be then simplified, leading to:

$$\begin{aligned} Z_{L_l} &= \prod_{n=1}^{l} Z_n \\ Z_{S_l} &= (-1)^l Z_l \qquad Z_{S_1} = \prod_{n=2}^{N} Z_n \\ X_{S_l} X_{L_l} X_{S_{l+1}} &= X_l X_{l+1} \qquad X_{S_N} X_{L_N} X_{S_1} = X_N \end{aligned} \tag{75}$$

Now, let us define the following operators:

$$\begin{aligned} \phi_l &= \frac{1}{2}\left(1 + \prod_{n=1}^{l} (-1)^{n-1} Z_n\right) X_l \\ \phi_l^\dagger &= \frac{1}{2}\left(1 - \prod_{n=1}^{l} (-1)^{n-1} Z_n\right) X_l \end{aligned} \tag{76}$$

which satisfy the following commutation relations

$$\begin{aligned} &\{\phi_l, \phi_l\} = 0 = \{\phi_l^\dagger, \phi_l^\dagger\} \qquad \{\phi_l^\dagger, \phi_l\} = 1 \\ &\{O_i, O_j\}_{i<j} = O_i(\phi_j^\dagger + \phi_j) \end{aligned} \tag{77}$$

Where O_i is either ϕ_i or $\phi_i^\dagger$. With those definitions, it is easy to prove that:

$$\begin{aligned} Z_{S_l} &= 1 - 2\phi_l^\dagger \phi_l \\ Z_{L_l} &= (-1)^l (1 - 2\phi_l^\dagger \phi_l)(1 - 2\phi_{l-1}^\dagger \phi_{l-1}) \\ X_l &= \phi_l^\dagger \phi_l \end{aligned} \tag{78}$$

From here, we can already see that everything is local, while substituting into the Hamiltonian we get:

$$\begin{aligned} H &= m \sum_{l=1}^{N} (-1)^l \frac{1}{2}(1 - (-1)^l Z_{S_l}) + \epsilon \sum_{l=1}^{N} \frac{1}{2}(1 + Z_{S_l} Z_{S_{l+1}}) X_{S_l} X_{L_l} X_{S_{l+1}} + 2\lambda_E \sum_{l=1}^{N} Z_{L_l} \\ &= m \sum_{l=1}^{N} \frac{1}{2}\left((-1)^l - (1 - 2\phi_l^\dagger \phi_l)(1 - 2\phi_{l-1}^\dagger \phi_{l-1})\right) \\ &+ 2\epsilon \sum_{l=1}^{N} \left(1 - \phi_{l-1}^\dagger \phi_{l-1} - \phi_{l+1}^\dagger \phi_{l+1} + 2\phi_{l-1}^\dagger \phi_{l-1} \phi_{l+1}^\dagger \phi_{l+1}\right)(\phi_l^\dagger + \phi_l)(\phi_{l+1}^\dagger + \phi_{l+1}) \\ &+ \lambda_E \sum_{l=1}^{N} (1 - 2\phi_l^\dagger \phi_l) \end{aligned} \tag{79}$$

One possible advantage of this choice of logical operations is that, since ϕ is non-local, in some cases it could be used to simplify some non-locality raising from other things (like Jordan-Wigner or other encodings).

References

[1] S. P. Jordan, K. S. M. Lee, and J. Preskill, "Quantum algorithms for quantum field theories," *Science*, vol. 336, no. 6085, pp. 1130–1133, 2012. [Online]. Available: https://doi.org/10.48550/arXiv.1111.3633

[2] S. P. Jordan, H. Krovi, K. S. M. Lee, and J. Preskill, "BQP-completeness of scattering in scalar quantum field theory," *Quantum*, vol. 2, p. 44, Jan. 2018. [Online]. Available: https://doi.org/10.22331/q-2018-01-08-44

[3] 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," *The European Physical Journal D*, vol. 74, no. 8, Aug. 2020. [Online]. Available: http://dx.doi.org/10.1140/epjd/e2020-100571-8

[4] E. Zohar, "Quantum simulation of lattice gauge theories in more than one space dimension—requirements, challenges and methods," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 380, no. 2216, Dec. 2021. [Online]. Available: http://dx.doi.org/10.1098/rsta.2021.0069

[5] N. Klco, A. Roggero, and M. J. Savage, "Standard model physics and the digital quantum revolution: thoughts about the interface," *Reports on Progress in Physics*, vol. 85, no. 6, p. 064301, May 2022. [Online]. Available: http://dx.doi.org/10.1088/1361-6633/ac58a4

[6] C. W. Bauer, Z. Davoudi, A. B. Balantekin, T. Bhattacharya, M. Carena, W. A. de Jong, P. Draper, A. El-Khadra, N. Gemelke, M. Hanada, D. Kharzeev, H. Lamm, Y.-Y. Li, J. Liu, M. Lukin, Y. Meurice, C. Monroe, B. Nachman, G. Pagano, J. Preskill, E. Rinaldi, A. Roggero, D. I. Santiago, M. J. Savage, I. Siddiqi, G. Siopsis, D. Van Zanten, N. Wiebe, Y. Yamauchi, K. Yeter-Aydeniz, and S. Zorzetti, "Quantum simulation for high-energy physics," *PRX Quantum*, vol. 4, p. 027001, May 2023. [Online]. Available: https://doi.org/10.1103/PRXQuantum.4.027001

[7] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, pp. R2493–R2496, Oct 1995. [Online]. Available: https://doi.org/10.1103/PhysRevA.52.R2493

[8] D. Gottesman, "Stabilizer codes and quantum error correction," 1997. [Online]. Available: https://doi.org/10.48550/arXiv.quant-ph/9705052

[9] ——, "Theory of fault-tolerant quantum computation," *Phys. Rev. A*, vol. 57, pp. 127–137, Jan 1998. [Online]. Available: https://doi.org/10.1103/PhysRevA.57.127

[10] E. Knill, R. Laflamme, and W. H. Zurek, "Resilient quantum computation," *Science*, vol. 279, no. 5349, pp. 342–345, 1998. [Online]. Available: https://doi.org/10.1126/science.279.5349.342

[11] B. M. Terhal, "Quantum error correction for quantum memories," *Rev. Mod. Phys.*, vol. 87, pp. 307–346, Apr 2015. [Online]. Available: https://doi.org/10.1103/RevModPhys.87.307

[12] J. Roffe, "Quantum error correction: an introductory guide," *Contemporary Physics*, vol. 60, no. 3, pp. 226–245, 2019. [Online]. Available: https://doi.org/10.1080/00107514.2019.1667078

[13] J. R. Stryker, "Oracles for gauss's law on digital quantum computers," *Phys. Rev. A*, vol. 99, p. 042301, Apr 2019. [Online]. Available: https://doi.org/10.1103/PhysRevA.99.042301

[14] I. Raychowdhury and J. R. Stryker, "Solving gauss's law on digital quantum computers with loop-string-hadron digitization," *Phys. Rev. Res.*, vol. 2, p. 033039, Jul 2020. [Online]. Available: https://doi.org/10.1103/PhysRevResearch.2.033039

[15] P. Faist, S. Nezami, V. V. Albert, G. Salton, F. Pastawski, P. Hayden, and J. Preskill, "Continuous symmetries and approximate quantum error correction," *Phys. Rev. X*, vol. 10, p. 041018, Oct 2020. [Online]. Available: https://doi.org/10.1103/PhysRevX.10.041018

[16] N. Klco and M. J. Savage, "Hierarchical qubit maps and hierarchically implemented quantum error correction," *Phys. Rev. A*, vol. 104, p. 062425, Dec 2021. [Online]. Available: https://doi.org/10.1103/PhysRevA.104.062425

[17] L. Kong and Z.-W. Liu, "Near-optimal covariant quantum error-correcting codes from random unitaries with symmetries," *PRX Quantum*, vol. 3, p. 020314, Apr 2022. [Online]. Available: https://doi.org/10.1103/PRXQuantum.3.020314

[18] A. Rajput, A. Roggero, and N. Wiebe, "Quantum error correction with gauge symmetries," *npj Quantum Inf.*, vol. 9, p. 41, Apr 2023. [Online]. Available: https://doi.org/10.1038/s41534-023-00706-8

[19] J. del Pino and O. Zilberberg, "Dynamical gauge fields with bosonic codes," *Phys. Rev. Lett.*, vol. 130, p. 171901, Apr 2023. [Online]. Available: https://doi.org/10.1103/PhysRevLett.130.171901

[20] E. J. Gustafson and H. Lamm, "Robustness of gauge digitization to quantum noise," 2023. https://doi.org/10.48550/arXiv.2301.10207

[21] Y.-A. Chen, A. V. Gorshkov, and Y. Xu, "Error-correcting codes for fermionic quantum simulation," *SciPost Phys.*, vol. 16, p. 033, 2024. [Online]. Available: https://doi.org/10.21468/SciPostPhys.16.1.033

[22] M. Carena, H. Lamm, Y.-Y. Li, and W. Liu, "Quantum error thresholds for gauge-redundant digitizations of lattice field theories," 2024. https://doi.org/10.1103/PhysRevD.110.054516

[23] E. Zohar and J. I. Cirac, "Eliminating fermionic matter fields in lattice gauge theories," *Phys. Rev. B*, vol. 98, p. 075119, Aug 2018. [Online]. Available: https://doi.org/10.1103/PhysRevB.98.075119

[24] U. Borla, R. Verresen, F. Grusdt, and S. Moroz, "Confined phases of one-dimensional spinless fermions coupled to Z_2 gauge theory," *Phys. Rev. Lett.*, vol. 124, p. 120503, Mar 2020. [Online]. Available: https://doi.org/10.1103/PhysRevLett.124.120503

[25] U. Borla, B. Jeevanesan, F. Pollmann, and S. Moroz, "Quantum phases of two-dimensional $\mathbb{z}_2$ gauge theory coupled to single-component fermion matter," *Phys. Rev. B*, vol. 105, p. 075132, Feb 2022. [Online]. Available: https://doi.org/10.1103/PhysRevB.105.075132

[26] G. H. Low and I. L. Chuang, "Optimal hamiltonian simulation by quantum signal processing," *Phys. Rev. Lett.*, vol. 118, p. 010501, Jan 2017. [Online]. Available: https://doi.org/10.1103/PhysRevLett.118.010501

[27] ——, "Hamiltonian Simulation by Qubitization," *Quantum*, vol. 3, p. 163, Jul. 2019. [Online]. Available: https://doi.org/10.22331/q-2019-07-12-163

[28] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, "Theory of trotter error with commutator scaling," *Physical Review X*, vol. 11, no. 1, p. 011020, 2021. https://doi.org/10.1103/PhysRevX.11.011020

[29] D. Horn, M. Weinstein, and S. Yankielowicz, "Hamiltonian approach to $z(n)$ lattice gauge theories," *Phys. Rev. D*, vol. 19, pp. 3715–3731, Jun 1979. [Online]. Available: https://doi.org/10.1103/PhysRevD.19.3715

[30] U.-J. Wiese, "Ultracold quantum gases and lattice systems: quantum simulation of lattice gauge theories," *Annalen der Physik*, vol. 525, no. 10-11, pp. 777–796, jul 2013. [Online]. Available: https://doi.org/10.1002%2Fandp.201300104

[31] E. Zohar, J. I. Cirac, and B. Reznik, "Quantum simulations of lattice gauge theories using ultracold atoms in optical lattices," *Reports on Progress in Physics*, vol. 79, no. 1, p. 014401, jan 2016. https://doi.org/10.1098/rsta.2021.0069

[32] N. P. Breuckmann and J. N. Eberhardt, "Quantum low-density parity-check codes," *PRX Quantum*, vol. 2, p. 040101, Oct 2021. [Online]. Available: https://doi.org/10.1103/PRXQuantum.2.040101

[33] F. Verstraete and J. I. Cirac, "Mapping local hamiltonians of fermions to local hamiltonians of spins," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09012–P09012, Sep. 2005. [Online]. Available: http://dx.doi.org/10.1088/1742-5468/2005/09/P09012

[34] A. M. Childs and N. Wiebe, "Hamiltonian simulation using linear combinations of unitary operations," *Quantum Info. Comput.*, vol. 12, no. 11–12, p. 901–924, nov 2012. https://doi.org/10.48550/arXiv.1202.5822

[35] Dmitri Maslov, "Optimal and asymptotically optimal NCT reversible circuits by the gate types". Available: https://doi.org/10.48550/arXiv.1602.02627

[36] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, "Toward the first quantum simulation with quantum speedup," *Proceedings of the National Academy of Sciences*, vol. 115, no. 38, pp. 9456–9461, sep 2018. [Online]. Available: https://doi.org/10.1073%2Fpnas.1801723115

[37] A. M. Steane, "Error correcting codes in quantum theory," *Phys. Rev. Lett.*, vol. 77, pp. 793–797, Jul 1996. [Online]. Available: https://doi.org/10.1103/PhysRevLett.77.793

[38] C. Chamberland and M. E. Beverland, "Flag fault-tolerant error correction with arbitrary distance codes," *Quantum*, vol. 2, p. 53, Feb. 2018. [Online]. Available: http://dx.doi.org/10.22331/q-2018-02-08-53

[39] R. Chao and B. W. Reichardt, "Flag fault-tolerant error correction for any stabilizer code," *PRX Quantum*, vol. 1, no. 1, Sep. 2020. [Online]. Available: http://dx.doi.org/10.1103/PRXQuantum.1.010302

[40] A. F. Shaw, P. Lougovski, J. R. Stryker, and N. Wiebe, "Quantum Algorithms for Simulating the Lattice Schwinger Model," *Quantum*, vol. 4, p. 306, Aug. 2020. [Online]. Available: https://doi.org/10.22331/q-2020-08-10-306

[41] A. Rajput, A. Roggero, and N. Wiebe, "Hybridized Methods for Quantum Simulation in the Interaction Picture," *Quantum*, vol. 6, p. 780, Aug. 2022. [Online]. Available: https://doi.org/10.22331/q-2022-08-17-780

[42] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, "Exponential improvement in precision for simulating sparse hamiltonians," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, ser. STOC '14. ACM, May 2014. [Online]. Available: http://dx.doi.org/10.1145/2591796.2591854

[43] A. M. Steane, "Simple quantum error-correcting codes," *Physical Review A*, vol. 54, no. 6, p. 4741–4751, Dec. 1996. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.54.4741

[44] P. Jordan and E. P. Wigner, "About the Pauli exclusion principle," *Z. Phys.*, vol. 47, pp. 631–651, 1928. http://dx.doi.org/10.1007/BF01331938

[45] Wauters, M., Ballini, E., Biella, A. & Hauke, P. Symmetry-protection Zeno phase transition in monitored lattice gauge theories. (2024), https://doi.org/10.1103/PhysRevB.111.094315