

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

Suppression of effective noise in Hamiltonian simulations

Milad Marvian, Todd A. Brun, and Daniel A. Lidar

Phys. Rev. A **96**, 052328 — Published 22 November 2017

DOI: [10.1103/PhysRevA.96.052328](https://doi.org/10.1103/PhysRevA.96.052328)

Suppression of effective noise in Hamiltonian simulations

Milad Marvian,^{1,2} Todd A. Brun,^{1,2,3} and Daniel A. Lidar^{1,2,3,4}

¹*Department of Electrical Engineering, University of Southern California, Los Angeles, California 90089, USA*

²*Center for Quantum Information Science & Technology,*

University of Southern California, Los Angeles, California 90089, USA

³*Department of Physics and Astronomy, University of Southern California, Los Angeles, California 90089, USA*

⁴*Department of Chemistry, University of Southern California, Los Angeles, California 90089, USA*

Simulating high-weight Hamiltonians can convert local noise on the original Hamiltonian into undesirable nonlocal noise on the simulated Hamiltonian. Here we show how starting from two-local Hamiltonian in the presence of non-Markovian noise, a desired computation can be simulated as well as protected using fast pulses, while maintaining an energy gap against the errors created in the process.

I. INTRODUCTION

The theory of quantum fault tolerance ensures that quantum computers can operate reliably in the presence of decoherence and noise [1, 2]. The quantum accuracy threshold theorem, in various incarnations [3–13], guarantees that arbitrary long, reliable quantum computation is achievable if the error rate is below a threshold. Although in principle the existence of this threshold means that scalable quantum computation is possible, in practice the value of this threshold and the required overhead are very important, as achieving them in experiments remains extremely challenging.

One approach to reduce these requirements is to use active quantum error correction in combination with other methods that provide additional robustness against instability or noise. Among such methods are holographic quantum computation [14] and topological quantum computation [15]. In [16, 17] schemes have been proposed based on encoding the information in the ground subspace of a Hamiltonian with a constant energy gap and topological properties, such as the surface code Hamiltonian, and performing the computation by adiabatically deforming the Hamiltonian. The energy gap of the Hamiltonian suppresses the thermal excitations induced by the environment. Compatible active error correction is performed frequently enough to prevent logical errors. As the energy gap protection from a Hamiltonian stabilizer code on a two-dimensional lattice cannot increase with system size [18], to sustain arbitrary long universal quantum computation on such a lattice with stabilizer codes requires active error correction.

Although this construction is appealing, the surface code Hamiltonian and also the time-dependent Hamiltonian implementing the deformation consist of up to 4-local interactions [16]. Implementing such interactions experimentally is difficult (though not impossible and various proposals exist [19–21]). One way to circumvent this is to simulate these Hamiltonians using other available resources. For example, simulation of stabilizer Hamiltonians at finite temperature is discussed in Refs. [22] and [23]. But the noise of the original system may be dramatically transformed by the simulation pro-

cedure. It is important to investigate how the simulation process converts realistic noise on the original resources to new effective noise on the simulated Hamiltonian. For example, local noise on the original Hamiltonian can be converted into nonlocal noise on the simulated Hamiltonian, and this could reduce the effectiveness of the simulated Hamiltonian in suppressing errors. This is a general concern for various simulation methods, including stroboscopic methods (see, e.g., [24, 25]) and perturbative gadgets [26–28].

The question of the effect of a Markovian noise on simulation has been partially studied [29], but despite its importance [30], only a limited amount of attention has been devoted to the effects of general noise on the simulation of high-weight Hamiltonians. In this work we show how—starting from an entangling Hamiltonian in the presence of a general local non-Markovian environment—one can generate the desired nonlocal Hamiltonian by the application of one-local unitary operators. By combining simulation with schemes for dynamical decoupling (DD) [2, 31, 32], we construct a sequence of pulses that suppresses errors while simulating the desired interactions. The relationship between the strength of the simulated Hamiltonian and the strength of effective noise is investigated. We also consider how far the transformed errors spread, based on the locality (both geometric and algebraic) of the bath Hamiltonian and the system-bath interaction.

We illustrate our construction with the Hamiltonians used in Ref. [16] for fault-tolerant quantum computation in the surface code, but it can easily be modified to simulate other, similar Hamiltonians, including the Hamiltonians used for protection of adiabatic quantum computation in Ref. [33]. In the latter, a stabilizer code that can detect the effect of the environment on the system is chosen, the system Hamiltonian is then encoded using the logical operators, while a penalty Hamiltonian is added to break the degeneracy. Again, the energy gap of the constructed Hamiltonian suppresses the rate of excitation out of the code space [34]. This scheme also requires 4-body interactions.

II. SUMMARY OF MAIN RESULTS AND ORGANIZATION OF THE PAPER

Our main contributions in this work can be summarized as follows:

- We provide a method to simulate Hamiltonians with high-weight interactions on a grid, such as surface code Hamiltonians, in a number of steps that is independent of the size of the grid. We demonstrate this with detailed construction steps for surface code Hamiltonians, using two different stroboscopic simulation techniques.
- We provide a method to design new pulse sequences that can simulate a desired Hamiltonian while pushing errors (caused by the presence of the bath) to higher orders, by combining DD and Hamiltonian simulation techniques. To do so, we use two applications of the symmetrization procedure [32, 35]. The first application shows how to protect a Hamiltonian while averaging out all the errors. The second application reduces the number of pulses needed to average out local errors.
- For local, but otherwise general, non-Markovian noise on the original Hamiltonian we show how the effective error on the simulated Hamiltonian becomes non-local. We demonstrate that this effective noise is mostly suppressed during the simulation of a surface code Hamiltonian because of an effective gap.

This paper is organized as follows. In Sec. III we introduce the resources and the goal of the simulation. In Sec. IV, we describe two different methods to simulate the desired high-weight Hamiltonian on a grid in the noise-free setting. We also discuss how to simulate a time-dependent Hamiltonian that, with the help of active error correction, can perform universal quantum computation fault-tolerantly. In Sec. V, we discuss the effects of general non-Markovian noise on the simulation. We then show how using DD techniques in the simulation procedure can reduce the strength of the effective noise. We also discuss how the locality of the bath and system-bath interaction Hamiltonian changes the spread of the noise and also the resources required for a successful simulation. In Sec. VI, we explicitly show that during the simulation an effective gap against the strongest errors created in the process is maintained. Section VII is devoted to conclusion and discussion. Additional technical details are provided in the Appendix.

III. SETUP

Assume N qubits are placed on a two-dimensional square lattice as depicted in Fig. 1. Let $d(i, j)$ denote the Euclidean distance between qubits i and j ; we choose

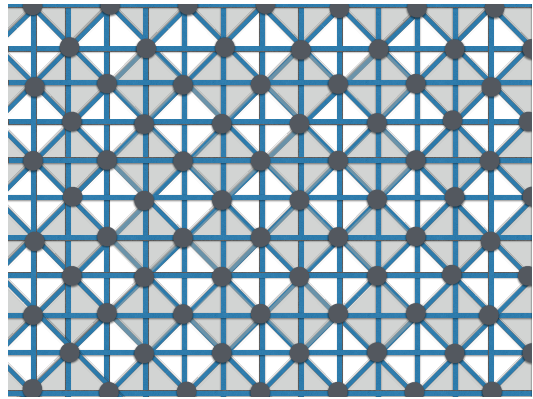


FIG. 1. Interacting qubits on a grid. Blue (gray) lines represent XX interactions between qubits (circles). There is an interaction between nearest neighbor and next-nearest neighbor qubits. For the toric Hamiltonian, the goal is to simulate $X^{\otimes 4}$ on the shaded areas and $Z^{\otimes 4}$ on the light areas.

units such that $d(i, j) = 1$ if qubits i and j are nearest neighbors. A two-body entangling Hamiltonian acts on nearest neighbor and next-nearest neighbor qubits (with $d(i, j) = \sqrt{2}$):

$$H_X = \sum_{d(i,j) \leq \sqrt{2}} X_i X_j, \quad (1)$$

where X_i denotes the Pauli σ^x matrix acting on qubit i . (An example of an alternative connectivity is provided in Appendix A.)

Let us start with the toric code Hamiltonian; later we will convert it to the surface code Hamiltonian with cuts. In this case the Hamiltonian is:

$$H_p = \sum_v A_v + \sum_p B_p, \quad (2)$$

with each vertex operator A_v corresponding to an $X^{\otimes 4}$ term acting on the spins connected to the vertex v and each plaquette operator B_p corresponding to a $Z^{\otimes 4}$ term acting on the spins in the plaquette p ; see Fig. 1 (Z denotes the Pauli σ^z matrix). This Hamiltonian is 4-local, and as the first step we show how to simulate it in the ideal case (no noise) using the 2-local Hamiltonian H_X .

Having stroboscopic simulation in mind, for now we assume that we can apply instantaneous, ideal unitary pulses on each qubit. For simplicity we assume that the time interval between pulses is fixed (more general schemes can improve the results).

IV. SIMULATION: IDEAL CASE

Assuming there is no noise, here we show how to use the commutator method to simulate the Hamiltonian H_p of Eq. (2) using the Hamiltonian H_X of Eq. (1). We set $\hbar \equiv 1$.

A. Commutator method

One way to build a 3-local Hamiltonian using 2-local Hamiltonians is to use the identity:

$$\begin{aligned}\Omega_\tau(A, B) &\equiv e^{iB\tau} e^{iA\tau} e^{-iB\tau} e^{-iA\tau} \\ &= e^{-i\tau^2([A, B])} + O(\tau^3),\end{aligned}\quad (3)$$

valid for any pair of operators A and B . This allows us to generate an effective 3-local Hamiltonian $i\delta t[H_a, H_b]$ using appropriate 2-local Hamiltonians H_a and H_b :

$$\Omega_{\delta t}(H_a, H_b) = e^{-i\delta t^2([H_a, H_b])} + O(\delta t^3). \quad (4)$$

Note that in order to neglect the higher order terms, obviously $\|H_a\|\delta t$ and $\|H_b\|\delta t$ must be small. Also, the effective Hamiltonian contains δt , and so is much weaker than the Hamiltonians we started with.

The same method can be used iteratively to increase the locality of the effective Hamiltonian. To generate a 4-local Hamiltonian we notice that by negating H_b or H_a , the term $[H_a, H_b]$ is also negated. So we repeat the procedure with H_a , $-H_b$, and a new Hamiltonian H_c :

$$\begin{aligned}U &= e^{iH_c\delta t}\Omega_{\delta t}(H_a, -H_b)e^{-iH_c\delta t}\Omega_{\delta t}(H_a, H_b) \\ &= e^{-i\delta t^3([H_a, H_b], H_c) + [H_a, H_b], H_b]} + O(\delta t^4).\end{aligned}\quad (5)$$

Considering δt as the duration that the simulated Hamiltonian is implemented, the term $\delta t^2([H_a, H_b], H_c]$ is the desired 4-local Hamiltonian. The extra term, $\delta t^2([H_a, H_b], H_b]$, is the result of the third-order error from Eq. (4) which becomes relevant now:

$$\begin{aligned}\Omega_{\delta t}(H_a, H_b) &= e^{-i\delta t^2([H_a, H_b]) - i\delta t^3([H_a, H_b], H_a + H_b)} + O(\delta t^4),\end{aligned}\quad (6)$$

where the third-order error is not negated by the replacement $H_b \mapsto -H_b$; the term $[H_a, H_b], H_b]$ remains. Using the operator identity

$$ue^A u^\dagger = e^{uAu^\dagger} = \begin{cases} e^A & [u, A] = 0 \\ e^{-A} & \{u, A\} = 0 \end{cases} \quad (7)$$

valid for unitary u and arbitrary A , one can apply a pulse u , itself generated by a 2-local Hamiltonian, that commutes with the desired Hamiltonian and anticommutes with the extra term to eliminate it:

$$UuUu^\dagger = e^{-2i\delta t^3([H_a, H_b], H_c)} + O(\delta t^4). \quad (8)$$

This process takes time $20\delta t$. This type of simulation is stroboscopic, so we obtain the desired effective Hamiltonian at a specific time (here at $t = 20\delta t$).

In the above discussion we used three different Hamiltonians $H_{a,b,c}$. For the actual simulation, we will assume that H_X is always on, and then apply pulses (not generated by H_X) to generate the effective Hamiltonians $H_{\{a,b,c\}}$.

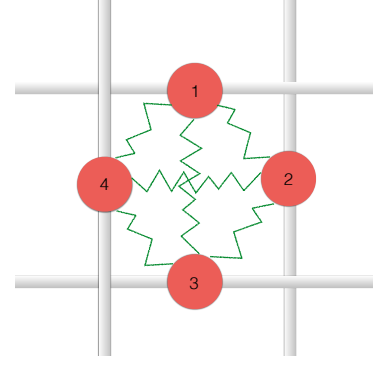


FIG. 2. Depiction of $H_x = \sum_{1 \leq i < j \leq 4} X_i X_j$ for four qubits in a plaquette.

B. Simulating one plaquette operator

Let us now show how to use the Hamiltonian

$$H_x = \sum_{1 \leq i < j \leq 4} X_i X_j \quad (9)$$

(not to be confused with H_X) to simulate the Hamiltonian $X_1 X_2 X_3 X_4$ on the qubits of a plaquette (see Fig. 2).

We will repeatedly use some basic identities, which are listed next for convenience, as they will be used throughout the remainder of this work:

$$\begin{aligned}WXW &= Z, & WYW &= -Y, & WZW &= X, \\ SX S^\dagger &= Y, & SY S^\dagger &= -X, & SZ S^\dagger &= Z,\end{aligned}\quad (10)$$

where W is the Hadamard gate and S is the phase gate.

Let the system evolve with the Hamiltonian H_x for a time $2\delta t$, and apply $W_1 W_2, u = Z_1 Z_2$ and $W_1 W_2 u$ pulses at times $t = 0, \delta t$ and $2\delta t$ respectively, where W_i is a Hadamard pulse on the i -th qubit:

$$W_1 W_2 (u e^{-iH_x \delta t} u) e^{-iH_x \delta t} W_1 W_2 = e^{-i2(Z_1 Z_2 + X_3 X_4) \delta t}. \quad (11)$$

The resulting Hamiltonian is $H_a = 2(Z_1 Z_2 + X_3 X_4)$.

We generate $H_b = 2(Y_1 X_4 + X_2 X_3)$ by first applying the inverse phase gate S_1^\dagger to the first qubit, then applying $u = Z_1 Z_4$ at time δt , and finally $S_1 u$ at time $2\delta t$:

$$S_1 (u e^{-iH_x \delta t} u) e^{-iH_x \delta t} S_1^\dagger = e^{-i2(Y_1 X_4 + X_2 X_3) \delta t}. \quad (12)$$

We also generate $H_c = 2(Y_2 X_3 + X_1 X_4)$ by applying pulses $S_2^\dagger, u = Z_1 Z_4$, and $S_2 u$ at times $0, \delta t$ and $2\delta t$:

$$S_2 (u e^{-iH_x \delta t} u) e^{-iH_x \delta t} S_2^\dagger = e^{-i2(X_1 X_4 + Y_2 X_3) \delta t}. \quad (13)$$

Generating any of $-H_{a,b,c}$ is almost the same as generating $H_{a,b,c}$, but with a few additional pulses. For example, for H_a we notice that:

$$(X_1 Z_3) e^{-i2(Z_1 Z_2 + X_3 X_4) \delta t} (X_1 Z_3) = e^{i2(Z_1 Z_2 + X_3 X_4) \delta t}. \quad (14)$$

In the same way one can generate $-H_b$ by conjugating H_b with X_1Z_2 pulses, and $-H_c$ by conjugating H_c with Z_1Z_2 pulses.

Having all the ingredients needed for Eq. (5), the effective Hamiltonian at order δt^3 contains the desired 4-body Hamiltonian with some extra terms that can be removed with only one extra step. To see this, note that:

$$[[H_a, H_b], H_c] = 32(X_1X_2X_3X_4 + Y_1Y_2X_3X_4), \quad (15a)$$

$$[[H_a, H_b], H_b] = 64(Z_1Z_2 + X_1Y_2X_3X_4). \quad (15b)$$

One then follows Eq. (8) with an extra pulse $u = Z_1Y_2$, which commutes with the desired term ($X_1X_2X_3X_4$) and anticommutes with all the other terms. The effective Hamiltonian becomes $64\delta t^2 X_1X_2X_3X_4$, as desired.

Therefore, in $N_X = 40$ steps we can convert the always-on Hamiltonian H_x (after a time $N_X\delta t$) to $64\delta t^2 X_1X_2X_3X_4$, with errors of order δt^4 , which is the same as saying that the original Hamiltonian is stroboscopically converted to a 4-local Hamiltonian at the price of making it $64\delta t^2/N_X$ times weaker.

C. Simulating all the plaquette operators on the grid

We assume that simultaneous pulses on all the qubits of the grid can be applied in parallel. Using this we can simulate all the plaquette operators on the grid. To do so, we start with the Hamiltonian H_X and repeat the steps provided for one plaquette operator, while ensuring that no unwanted terms are created in the process.

We start by creating patterns similar to H_x by letting H_X evolve and apply Z pulses on the qubits that are colored in blue (light gray) in Fig. 3(a) at times δt and $2\delta t$, but leave alone the qubits that are colored in black (dark gray). (More precisely, we conjugate by Z pulses all the qubits of a plaquette if the plaquette is in both an odd row and an odd column of the grid, counting from the top left corner.) Every pair of neighboring blue (light gray) and black (dark gray) qubits is thus decoupled, since Z anticommutes with XX , while every pair of neighboring blue (light gray) qubits is unaffected, since ZZ commutes with XX . This results in an effective Hamiltonian, called H_{temp} , shown in Fig. 3(b) by red lines as the surviving XX interactions (the plaquettes in an odd row and an odd column, or plaquettes in an even row and an even column of the grid.) Now we use H_{temp} and apply pulses similar to Sec. IV B on the labelled qubits in Fig. 3(c) in parallel to generate plaquette operators in half of the grid, as shown in Fig. 3(d). The steps of this procedure, and the corresponding components $H_{\{a,b,c\}}$ are presented in Figs. 4 and 5.

As discussed in Sec. IV B, by applying Z_1Y_2 (but now in parallel on the whole grid) one can remove all the extra unwanted terms, and all that remains are the plaquette operators shown in Fig. 3(d).

Up to now, the effective Hamiltonian only contains half of the plaquette operators needed. The other half can be

generated by repeating the same procedure, but shifting the locations of the pulses by one qubit. In this case plaquette operators in the shaded area of Fig. 3(d) are generated. The combined process gives all the plaquette operators needed:

$$e^{-ih\delta t^3 \sum_v A_v} + O(\delta t^4), \quad (16)$$

where $h\delta t^2$ is strength of the effective Hamiltonian, and h accounts for various numerical factors (such as $64/N_X$ in Sec. IV B).

D. Simulating all the vertex operators on the grid

Noticing, using Eq. (7), that

$$W_1W_2W_3W_4e^{-iX_1X_2X_3X_4\delta t}W_1W_2W_3W_4 = e^{-iZ_1Z_2Z_3Z_4\delta t}, \quad (17)$$

the process to create the vertex operators on the grid is similar to the process for plaquette operators. One can just shift the location of pulses and then conjugate the whole effective Hamiltonian with Hadamard pulses on all the qubits of the grid.

E. Strength of the effective Hamiltonian

By simulating the plaquette operators and the vertex operators consecutively, we can effectively simulate the toric code Hamiltonian in $N_{\text{sim}} = 320$ steps,¹ and so in time $N_{\text{sim}}\delta t$, independent of size of the grid.

The generated 4-body operator on the grid is $2^9\delta t^3 H_p$. Taking the strength of H_X as 1, the strength of each $H_{\{a,b,c\}}$ is 4. Inserting these into the commutators, $[[H_a, H_b], H_c]$ has strength 4^4 . Another factor 2 is added when we remove the error term. Thus $4^2 * 2\delta t^3 = 2^9\delta t^3$. That is, we use H_X to generate the toric code Hamiltonian while making it $2^9\delta t^3/320\delta t = 1.6\delta t^2$ times weaker. As expected, decreasing δt makes the ratio smaller.

F. Boundaries and the surface code

The toric code is defined on a torus, without boundaries. We are also interested in the surface code, which has boundaries [36]. To simulate the boundaries for the surface code Hamiltonian, whether outer boundaries or the inner cuts representing qubits, two other procedures are also needed. The first is generating holes (either Z -cut or X -cut), which is straightforward. For example,

¹ $10 * 2$ comes from the commutator method, a factor of 4 comes from generating each of $H_{\{a,b,c\}}$ from the original Hamiltonian H_X . A factor of 2 is needed to generate the other half of the plaquettes, and another factor of 2 is needed to generate the vertex operators. Thus $N_{\text{sim}} = 10 * 2 * 4 * 2 * 2 = 320$.

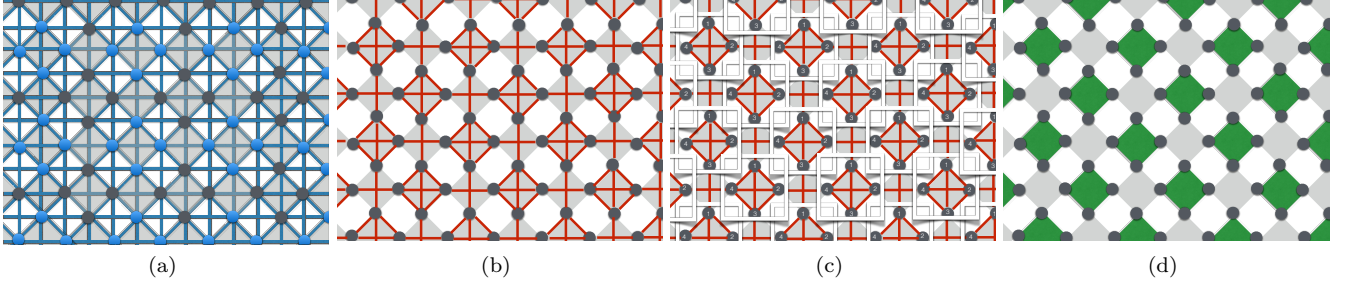


FIG. 3. Generating plaquette operators. (a) The location of Z pulses is shown in blue (light gray). (b) The remaining XX interactions after (a) are shown in red (dark gray). Their sum defines H_{temp} . (c) Labelling of qubits according to Fig. 2. (d) The result after applying pulses as is Sec. IV B to (c). This achieves the first goal of generating plaquette operators in the green (shaded) areas.

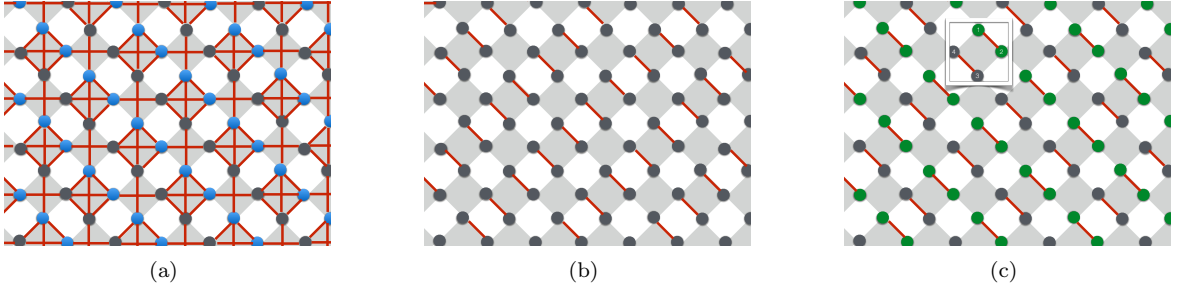


FIG. 4. Generating H_a on the grid. (a) Evolving under H_{temp} [Fig. 3(b)], we apply Z pulses on the blue (light gray) qubits at times δt and $2\delta t$. (b) The effective Hamiltonian at time $2\delta t$. (c) H_a is generated by conjugating the Hamiltonian in Fig. 4(b) with Hadamard pulses. The effective Hamiltonian becomes patterns of $Z_1 Z_2 + X_3 X_4$, depicted as green (light gray) coupled pairs (ZZ) and black (dark gray) coupled pairs (XX).

wherever it is necessary to create a hole one can choose H'_c such that it commutes with $[H_a, H_b]$. To accomplish this, rather than applying S_2 in Eq. (13), if we apply the Hadamard pulse W_2 , then $H'_c = Z_2 X_3 + X_1 X_4$, which commutes with $[H_a, H_b]$, and so the effective Hamiltonian is zero to order δt^3 . Thus, simply by using a Hadamard pulse rather than phase gate on a specific qubit we can generate a hole.

The second procedure needed is the simulation of 3-body interactions. One approach is to simulate them using Eq. (6) at order δt^2 , while eliminating errors of order δt^3 . Using the same pattern as Sec. IV C, starting from H_x in Fig. 2 we can choose $H_a = Z_1 X_2 + X_3 X_4$ and $H_b = Y_1 X_4 + X_2 X_3$. This results in an effective Hamiltonian $X_1 X_2 X_4 \delta t^2 + (Y_1 X_4 + Z_1 X_2) \delta t^3 + O(\delta t^4)$. By decoupling using one extra pulse X_1 , we can remove the terms proportional to δt^3 . Also, if in Fig. 2 the third qubit and the terms it involves are missing (these are edges of the Hamiltonian, corresponding to outer boundaries), then again we can generate $H_a = Z_1 X_2$ and $H_b = Y_1 X_4$, and then apply X_1 with the same final result.

To balance the different strength of the generated terms (boundaries are proportional to δt^2 while the rest of the construction is proportional to δt^3), the proportion of time slices implemented from each of these terms (in Trotterization of the evolutions) can be chosen to be δt .

Another approach is to implement these boundaries together with the other terms, but with flipping the signs of some of the Hamiltonians so that all but a fraction δt of the 3-body terms cancel out. (In our example, we could use Z_1 pulses: $Z_1 e^{-iX_1 X_2 X_4 \delta t^2} Z_1 = e^{iX_1 X_2 X_4 \delta t^2}$.)

An alternative approach to avoid the complicated scheduling described above is to apply pulses such as $e^{-iY\delta t}$ to implement single-body Hamiltonians. For example, starting from H_x in Fig. 2 we can repeat the procedure in Sec. IV B to generate $H_a = Z_1 X_2 + X_3 X_4$ and $H_b = Y_1 X_4 + X_2 X_3$, but now generate $H_c = X_1 X_4 + X_3 X_4 + X_1 X_3 + Y_2$. If so, the effective Hamiltonian to order δt^3 (after applying $Z_1 Y_2$ pulses to remove the extra terms, as mentioned in Sec. IV B) would become $X_1 X_2 X_4$ as desired. To generate such an H_c we can follow this procedure:

$$Z_2 e^{-iH_x \delta t} Z_2 e^{-iH_x \delta t} e^{-i2Y_2 \delta t} = e^{-i2(X_1 X_4 + X_3 X_4 + X_1 X_3 + Y_2) \delta t}. \quad (18)$$

G. $\pi/4$ -Conjugation Method

An alternative method for simulating high-weight Hamiltonians is based on the $\pi/4$ -conjugation identity (see, e.g., [24]). This identity for Pauli operators A, B

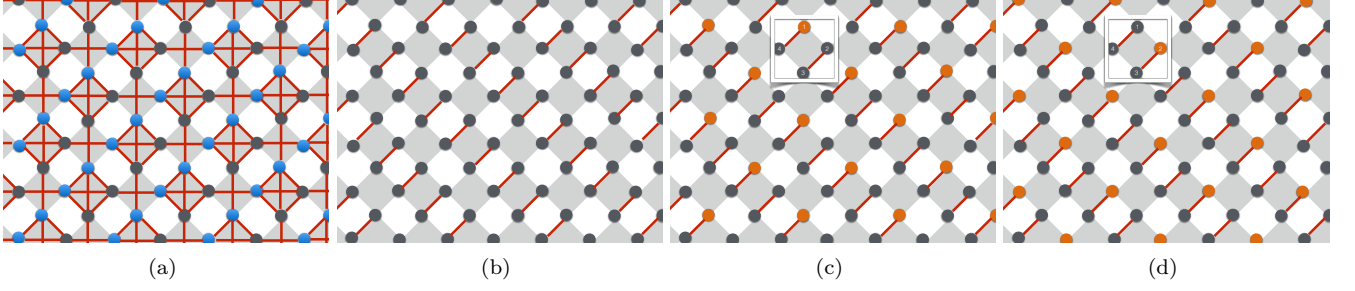


FIG. 5. Generating H_b and H_c on the grid. (a) Evolving under H_{temp} [Fig. 3(b)], we apply Z pulses on the blue (light gray) qubits at times δt and $2\delta t$. (b) The effective Hamiltonian at time $2\delta t$. (c) H_b is generated by conjugating the Hamiltonian in Fig. 5(b) with S pulses on the orange (light gray) qubits. The effective Hamiltonian becomes patterns of $Y_1X_4 + X_2X_3$, depicted as orange-black (light gray-dark gray) coupled pairs (YX) and black (dark gray) coupled pairs (XX). (d) H_c is generated by conjugating the Hamiltonian in Fig. 5(b) with S pulses on the orange (light gray) qubits. Now the effective Hamiltonian becomes patterns of $X_1X_4 + Y_2X_3$, depicted as black (dark gray) coupled pairs (XX) and orange-black (light gray-dark gray) coupled pairs (YX).

with $\{A, B\} = 0$ is

$$e^{-i\pi/4A} e^{i\theta B} e^{i\pi/4A} = e^{i\theta(AB)}, \quad (19)$$

which is exact. Obviously if $[A, B] = 0$ then we have

$$e^{-i\pi/4A} e^{i\theta B} e^{i\pi/4A} = e^{i\theta B}. \quad (20)$$

Assume the entangling Hamiltonian is $cH_x = c \sum_{i < j} X_i X_j$ with c being a constant representing the strength of the entangling Hamiltonian:

$$e^{-i\frac{\pi}{4c}(cH_x)} e^{-i\theta Y_1} e^{i\frac{\pi}{4c}(cH_x)} = e^{i\theta Z_1 X_2 X_3 X_4}. \quad (21)$$

Using this we can show that:

$$\begin{aligned} W_1 [e^{-i\frac{\pi}{4c}(cH_x)} e^{i\Delta t Y_1} (Y_1 e^{-i\frac{\pi}{4c}(cH_x)} Y_1)] W_1 \\ = i e^{-i\Delta t X_1 X_2 X_3 X_4}. \end{aligned} \quad (22)$$

So we apply three fast pulses while having H_x on for a length of time $2\pi/4c$. It is important to notice that in this method the simulation time, here $2\pi/4c$, is independent of Δt , the time over which we wish to simulate the evolution.

To simulate plaquette operators we again can conjugate this with Hadamard pulses:

$$\begin{aligned} W_1 W_2 W_3 W_4 e^{-i\Delta t X_1 X_2 X_3 X_4} W_1 W_2 W_3 W_4 \\ = e^{-i\Delta t Z_1 Z_2 Z_3 Z_4}. \end{aligned} \quad (23)$$

Implementation of the Hamiltonian on the grid can be done similarly to Sec. IV C, with slight modifications.

Using this procedure there is no error in simulation (as the identity is exact) and it takes a time proportional to π/c independent of Δt .

In contrast to the commutator method, where using the entangling Hamiltonian for a short time produces a weak effective Hamiltonian, here the entangling Hamiltonian has to evolve for the fixed time $\pi/4c$ to produce the desired effective Hamiltonian. For an experimental setup that can only generate weak entangling Hamiltonian (small c), this method requires a large time which

may make the implementation more prone to noise. For experiments with access to strong entangling Hamiltonians compared to the single-body gates and measurements, this method is beneficial as it is exact and the simulation time is independent of the time over which we wish to simulate a Hamiltonian. See [29] for a comparison of the $\pi/4$ -conjugation method to the commutator method in a Markovian environment.

H. Applying the scheme to fault-tolerant holonomic quantum computation in surface codes

Using adiabatic deformation of the surface code Hamiltonian in combination with active error correction it is possible to achieve fault-tolerant universal quantum computation [16]. In such a scheme, a gapped Hamiltonian protects the ground space from the deleterious effects of thermal noise, and the gates are applied by slowly deforming the Hamiltonian, such that the desired gate is applied to the ground state of the new Hamiltonian (holonomic quantum computation [14]). Explicit constructions to implement all the gates needed for universal quantum computation were proposed in [16] for such deformations.

The Hamiltonians needed for these schemes can be made geometrically local [16], but in practice it is challenging to implement them as they consist of up to 4-local interactions. Using the methods we proposed above, it is straightforward to simulate these Hamiltonians as well. For the purpose of illustration we choose a few examples from [16] and explain the needed procedure, but the constructions for the remaining needed interactions are similar.

As the first example, we note that the creation of a $|+\rangle$ state for an X -cut double qubit (and of a $|0\rangle$ state for a Z -cut double qubit) is done by just turning off two stabilizer terms (Sec IV.A in [16]). As explained in Sec. IV F, this can be done by changing a pulse such that the $O(\delta t^3)$

effective Hamiltonian vanishes.

We also need the ability to enlarge a hole. The proposed time-dependent Hamiltonian implementing the deformation is of the form:

$$H(t) = -J \left(\left(1 - \frac{t}{t_1}\right) B_2 + \frac{t}{t_1} X_1 + \sum_{p \neq 1,2} B_p + \sum_v A_v \right), \quad (24)$$

for $t \in [0, t_1]$ (see Eq. (37) in [16], with a specific choice of the monotonic function).

To simulate the time-dependent Hamiltonian, we can use Trotter-Suzuki expansions (see, e.g., [37]), and simulate each piece by applying the appropriate pulses. To do so, we define $\Delta t = \frac{t_1}{N_{tr}}$ and approximate the evolution generated by the time-dependent Hamiltonian with ordered evolutions consisting of several time independent Hamiltonians:

$$\prod_{m=0}^{N_{tr}} e^{-i((1-\frac{m}{N_{tr}})B_2 + \sum_{p \neq 1,2} B_p)J\Delta t} e^{-i\frac{m}{N_{tr}}X_1J\Delta t} e^{-iJ\sum_v A_v\Delta t} \quad (25)$$

To simulate these terms, one can generate the weakest 4-body interaction in this expansion using the building blocks we constructed in Sec. IV C, setting $\frac{J\Delta t}{N_{tr}} = h\delta t^3$:

$$\prod_{m=0}^{N_{tr}} (e^{-iB_2h\delta t^3})^{N_{tr}-m} (e^{-i\sum_{p \neq 1,2} B_ph\delta t^3})^{N_{tr}} (e^{-iX_1h\delta t^3})^m (e^{-ih\sum_v A_v\delta t^3})^{N_{tr}}. \quad (26)$$

Each 4-body term in this expansion can be simulated as described in Sec. IV C. We notice that B_2 can be constructed in parallel with the rest of the plaquette operators. Also, $e^{-iX_1h\delta t^3}$ is just a single qubit rotation applied using a pulse.

The error in the Trotterization procedure scales as Δt^2 , and is independent of the system size (the only non-commuting term is X_1 ; see [37]).

The same procedure works for the other Hamiltonians needed, such as another Hamiltonian needed to enlarge a hole (Eq. (39) in [16]), or the Hamiltonian needed for moving logical qubits (Eq. (47) in [16]).

The \bar{Z} and \bar{X} logical operators of the surface code are just strings of Z and X operators that connect appropriate boundaries (or holes). Thus any product thereof can be formed by applying parallel Z and X pulses (in practice one may avoid applying them and instead keep track of updates in software).

V. EFFECT OF NOISE

So far we have considered only the noise-free case. In this section we study the effect of noise on the simulation, and propose methods to suppress the effective noise of this process.

Again we use the entangling Hamiltonian H_X [Eq. (1)], but now we assume the presence of a general non-Markovian bath with Hamiltonian H_B , interacting with the system via the Hamiltonian H_{SB} . For now we assume that this interaction Hamiltonian is 1-local:

$$H_B = I \otimes B_0, \quad (27)$$

$$H_{SB} = \sum_i \sum_{\alpha \in \{x,y,z\}} \sigma_i^\alpha \otimes B_i^\alpha, \quad (28)$$

where i labels the the system qubit (when it is clear from the context, we will combine this index and the Pauli operator index into a single index: $H_{SB} = \sum_\beta \sigma_\beta \otimes B_\beta$.) The total Hamiltonian is:

$$H = H_X + H_B + \lambda H_{SB}, \quad (29)$$

with the dimensionless parameter λ being the strength of the system-bath coupling.

As the effective 4-body interactions constructed in previous sections appear in order δt^3 , our strategy is to choose a series of DD pulses that commute with the Hamiltonian H_X but suppress the errors. This use of DD pulses that commute with the Hamiltonian is an example of the general idea of using the stabilizer generators of a quantum error correction (or detection) code to perform a decouple-while-compute operation, which is possible since the Hamiltonian is a sum of the logical elements of the same code [38–40]. We will show how to protect the Hamiltonian H_X against the noise at least up to order δt^3 , so that the simulated Hamiltonian is not overwhelmed by the noise. To simulate the desired interactions, we combine the pulses designed for simulation with these DD pulses.

Note that DD schemes are effective when the timescale of the bath is long compared to the timescales of the pulses, and in fact DD can be shown to fail in the Markovian limit [41]. Therefore the method proposed here is only effective for non-Markovian environments.

We begin by first finding pulses that can suppress the errors in a general non-Markovian environment, and then discuss how further assumptions on the locality of the bath and the system-bath interaction allow for more efficient schemes.

To analyze the effects of DD pulses, and the corresponding effective Hamiltonian, it is handy to use the Magnus expansion (see Appendix B.) Here we choose the simplest DD schemes to illustrate the main ideas, but there is much room to use more sophisticated DD schemes (see, e.g., Ref. [2] for a review).

A. Protecting against noise at first and second order in δt

To remove the first order noise terms, we conjugate the Hamiltonian H_X with pulses $I^{\otimes N}, X^{\otimes N}, Y^{\otimes N}, Z^{\otimes N}$ in 4 steps. These pulses all commute with the Hamiltonian but remove the noise from the effective Hamiltonian to first order, via symmetrization [35, 42].

Then, using the fact that all the even-order terms of the Magnus expansion vanish for time-symmetric Hamiltonians [12], we simply append another 4 steps, conjugating with the same 4 pulses but in reverse order. This removes the first and second order terms of the noise from the effective Hamiltonian:

$$\begin{aligned} U_{sec} &= (I^{\otimes N} e^{-iH\delta t} I^{\otimes N})(X^{\otimes N} e^{-iH\delta t} X^{\otimes N}) \\ &\quad (Y^{\otimes N} e^{-iH\delta t} Y^{\otimes N})(Z^{\otimes N} e^{-iH\delta t} Z^{\otimes N}) \\ &\quad (Z^{\otimes N} e^{-iH\delta t} Z^{\otimes N})(Y^{\otimes N} e^{-iH\delta t} Y^{\otimes N}) \\ &\quad (X^{\otimes N} e^{-iH\delta t} X^{\otimes N})(I^{\otimes N} e^{-iH\delta t} I^{\otimes N}) \\ &= e^{-i(H_X+H_B)8\delta t+O(\delta t^3)}. \end{aligned} \quad (30)$$

Note that the middle $Z^{\otimes N}$ terms cancel, and that we can simplify this sequence to:

$$\begin{aligned} U_{sec} &= e^{-iH\delta t} X^{\otimes N} e^{-iH\delta t} Z^{\otimes N} e^{-iH\delta t} X^{\otimes N} e^{-2iH\delta t} \\ &\quad X^{\otimes N} e^{-iH\delta t} X^{\otimes N} e^{-iH\delta t} X^{\otimes N} e^{-iH\delta t} \\ &= e^{-i(H_X+H_B)8\delta t+O(\delta t^3)}. \end{aligned} \quad (31)$$

At this point, having evolved for $8\delta t$ (independent of the grid size), we have protected the H_X Hamiltonian from the effect of noise to first and second order in δt .

B. Third order error terms

First using a lemma we describe a general approach to protect a Hamiltonian while averaging out all other errors at a fixed order of time. It can be understood as an application of the symmetrization schemes proposed in [32, 35].

Lemma 1. (Symmetrization lemma: Protecting interaction) *Let P_1, \dots, P_n be commuting Pauli operators, and let \mathcal{P} be the group generated by these Pauli operators. Denote the set of all Pauli operators that commute with all the P_i s by $\mathcal{N}(\mathcal{P})$ (the normalizer of \mathcal{P}). Then for any Pauli operator s*

$$\forall s \notin \mathcal{P} : \sum_{g \in \mathcal{N}(\mathcal{P})} g s g^\dagger = 0, \quad (32)$$

$$\forall s \in \mathcal{P} : \sum_{g \in \mathcal{N}(\mathcal{P})} g s g^\dagger = s |\mathcal{N}(\mathcal{P})|. \quad (33)$$

As a special case, if \mathcal{P} only includes the identity operator we recover the usual Pauli twirling lemma (see, e.g., [43].)

Proof. See Appendix C. \square

Using this lemma for the particular plaquette operator discussed in section IV B, choosing $P_1 = X_1 X_2, P_2 = X_2 X_3, P_3 = X_3 X_4$ we need $4^4/2^3 = 32$ pulses to cancel out all the third-order errors. (Again, by symmetrization and doubling the number of pulses we get can get rid of fourth-order error as well.)

But to protect the whole grid, without additional assumptions about the locality of the bath part of the interaction Hamiltonians (the $\{B_\alpha\}$ operators), the number of pulses needed to cancel the terms at third order may grow with the grid size. The reason is that at higher orders, multiqubit error terms begin to appear without a locality assumption. To average out these terms by DD methods, the number of pulses must grow with the number of qubits (see Appendix D).

Later we will consider the case where the terms $\{B_\alpha\}$ are geometrically local, but here we show that even without this assumption it is possible to protect the Hamiltonian against the dominant noise terms appearing at third order in δt using a few pulses. To see this, we note that at order δt^3 , the terms that are first order in the coupling strength λ have a specific form: they are nested commutators of $\sigma_i^\alpha \otimes B_i^\alpha$ with two H_X terms:

$$\begin{aligned} &[[H_X, \sigma_i^\alpha \otimes B_i^\alpha], H_X] \\ &= [[X_i \otimes \sum_{\langle\langle i, i' \rangle\rangle} X_{i'}, \sigma_i^\alpha \otimes B_i^\alpha], H_X] \\ &= [[X_i, \sigma_i^\alpha] \otimes \sum_{\langle\langle i, i' \rangle\rangle} X_{i'} \otimes B_i^\alpha, H_X] \\ &= [[X_i, \sigma_i^\alpha], X_i] \otimes \left(\sum_{\langle\langle i, i' \rangle\rangle} X_{i'} \right)^2 \otimes B_i^\alpha. \end{aligned} \quad (34)$$

These terms are either $Y_i, Z_i, Y_i X_{i'} X_{i''}$ or $Z_i X_{i'} X_{i''}$, and are thus decoupled away by conjugating the evolution in Eq. (31) with $I^{\otimes N}$ and $X^{\otimes N}$ pulses:

$$\begin{aligned} &(I^{\otimes N} U_{sec} I^{\otimes N})(X^{\otimes N} U_{sec} X^{\otimes N}) \\ &= e^{-i(H_X+H_B)16\delta t+O(\lambda^2 \delta t^3)+O(\delta t^4)} \end{aligned} \quad (35)$$

In fact, these two pulses not only remove the λ^1 terms at order δt^3 , but at any order of δt . This is expected, as the first-order terms in λ are the result of the system Hamiltonian spreading the noise across more qubits. As the system Hamiltonian terms are all local and commuting, the noise to first order in λ can only have the form of either $Y_i X \dots X$ or $Z_i X \dots X$, and so conjugating with the proposed pulses can remove them.

Assuming that the system is weakly coupled to the environment (small λ), the dominant terms in the expansions are the lower powers of λ , so using just two additional pulses can cancel out all errors to first order in λ .

To sum up, we obtain this DD protection for H_X on the grid by evolving for time $16\delta t$, without any assumptions about geometric locality of the interaction Hamiltonian. With the noise removed by this DD procedure to order δt^3 , we can use our earlier constructions to produce the effective Hamiltonian without its being overwhelmed by noise.

C. Strength of the simulated Hamiltonian vs. effective noise

The effective noise process on the system depends on the particular schemes used for simulation and for DD. The ratio between this effective noise strength and the strength of the simulated Hamiltonian is important. If this ratio is small enough, the energy gap protection of the simulated Hamiltonian can be effective against the effective noise. Here we give estimates of the strength of each of these terms, but we also expect that for specific systems with more knowledge about the form of the noise, better bounds can be achieved.

We assume δt is the time interval between pulses. This is most likely determined by the physical limitations of the experiment. As described earlier, we choose $N_{\text{DD}}N_{\text{sim}}$ pulses to simulate a building block of the desired Hamiltonian at order δt^3 , with the effective noise at second order in λ and third order in δt (Fig. 6).

We first find a bound on the effective noise strength in each DD interval, and then add the effect of simulation on it. Following [12], we define the effective noise strength as:

$$\eta = \|U(N_{\text{DD}}\delta t) - U_{\text{ideal}}(\delta t)\|, \quad (36)$$

with U being the unitary generated by $H_X + H_B + H_{SB}$ with the DD pulses applied in between. The operator

$$U_{\text{ideal}}(t) \equiv e^{-i(H_B + H_X)N_{\text{DD}}t} \quad (37)$$

is the ideal case, with H_{SB} absent. We denote the cumulative unitary generated by all pulses at each time (not including the Hamiltonian terms) by $U_{\text{pulse}}(t)$. We assume that the pulses all commute with the system Hamiltonian H_X and also assume that the product of all pulses in one cycle gives identity: $U_{\text{pulse}}(N_{\text{DD}}\delta t) = I$. Moving to the interaction picture defined by the pulses we have:

$$\eta = \|\tilde{U}(N_{\text{DD}}\delta t) - U_{\text{ideal}}(\delta t)\|, \quad (38)$$

with

$$\tilde{U}(N_{\text{DD}}\delta t) = \text{Texp} \left[-i \int_0^{N_{\text{DD}}\delta t} H_X + H_B + \tilde{H}_{SB}(t) dt \right], \quad (39)$$

and

$$\tilde{H}_{SB}(t) = U_{\text{pulse}}^\dagger(t) H_{SB} U_{\text{pulse}}(t). \quad (40)$$

The Magnus expansion can be used to approximate $\tilde{U}(N_{\text{DD}}\delta t)$ with an effective Hamiltonian H_{eff} up to arbitrary order. Denoting $\tilde{U}(N_{\text{DD}}\delta t) = \exp(-iH_{\text{eff}}N_{\text{DD}}\delta t)$, the Magnus expansion gives each term in H_{eff} (see Appendix B):

$$H_{\text{eff}} = \sum_{k=0} H_{\text{eff}}^{(k)}. \quad (41)$$

Using this we have an upper bound on η as

$$\eta \leq N_{\text{DD}}\delta t \left\| \sum_{k=0} H_{\text{eff}}^{(k)} - (H_B + H_X) \right\|. \quad (42)$$

Assuming that the pulses achieve complete decoupling for first and second order, we have $H_{\text{eff}}^{(0)} = H_B + H_X$ and $H_{\text{eff}}^{(1)} = 0$. So we have

$$\eta \leq N_{\text{DD}}\delta t \left(\|H_{\text{eff}}^{(2)}\| + \left\| \sum_{k \geq 3} H_{\text{eff}}^{(k)} \right\| \right) \quad (43)$$

Now we can use the bounds on Magnus terms derived in [12], but a slight modification is needed. First is the addition of the H_X Hamiltonian, and the other is taking into account the effect of the two extra pulses we used to remove the first order terms in λ .

For the first term in Eq. (43), adding the terms corresponding to λ^2, λ^3 (adding Eqs. (112) and (113) in [12]: changing β to $\|H_B + H_X\|$ and replacing ϵ with $\|H_B + H_X + \lambda H_{SB}\|$), we have:

$$\|H_{\text{eff}}^{(2)}\| \leq (N\delta t)^2 \lambda^2 \|H_{SB}\|^2 (c_0 \lambda \|H_{SB}\| + c_1 \|H_B + H_X\|), \quad (44)$$

where c_0 and c_1 are some small constants. For the second term we have (using Eqs. (119) and (125) in [12]):

$$\begin{aligned} \left\| \sum_{k \geq 3} H_{\text{eff}}^{(k)} \right\| &\leq \\ &(N\delta t)^3 \lambda \|H_{SB}\| (\lambda \|H_{SB}\| + \|H_B + H_X\|)^3 \\ &\times (c_2 + c_3 (\lambda \|H_{SB}\| + \|H_B + H_X\|) N_{\text{DD}}\delta t), \end{aligned} \quad (45)$$

where c_2 and c_3 are some small constants. Denoting this upper bound by η' (the right hand side of Eq. (43)), the upper bound on the effective error strength of the combined DD and simulation procedure becomes $N_{\text{sim}}\eta'$ plus the error we get from the simulation. The strongest of these errors is at order δt^4 , and a bound on these terms is $c(N_{\text{sim}}N_{\text{DD}}\delta t)^4 \|H_X\|$. (This comes from counting the number of terms in the Magnus expansion and considering the locality of H_X , so rather than having $\|H_X\|^4$, we have a constant times $\|H_X\|$.) Similarly, higher order error terms of the simulation can also be included up to any desired accuracy.

The strength of the Hamiltonian itself, after the pulses, becomes $N_{\text{DD}}H_X$, and so after the simulation the desired Hamiltonian shows up at order $(N_{\text{DD}}\delta t)^3$. Increasing N_{DD} makes the simulated Hamiltonian stronger, but the error associated with DD can become worse, as can be seen from the bound.

D. Locality and strength of noise

Local noise on the original Hamiltonian can act as effective nonlocal noise on the simulated Hamiltonian. This is true for the DD process, and also for the simulation process. Here, starting from the Hamiltonian in Eq. (29), we investigate how the locality and strength of

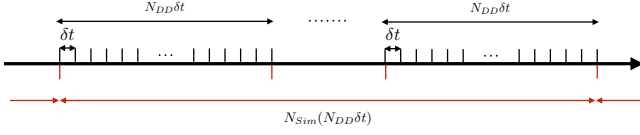


FIG. 6. Timing of the pulses: combining a DD scheme with N_{DD} pulses and a simulation with N_{sim} steps.

the effective noise changes after applying pulses. For generality, we assume that the system Hamiltonian is k -local and the system bath interaction, H_{SB} , consists of l -local terms on the system (for our construction $k = 2, l = 1$).

Using the Magnus expansion it is evident that the terms in δt^m are in the form of $m - 1$ nested commutators (basically, products of m Hamiltonians).

1. Growth of nonlocality of noise in higher order terms

First, let us assume that the bath part of the interaction Hamiltonian can be arbitrarily nonlocal: $\forall i \neq j : [B_\alpha^i, B_\beta^j] \neq 0$. Conjugating a segment of the evolution with u_a pulses, is equivalent to having a evolution with an effective Hamiltonian $H_i = H_{a,X} + H_B + \lambda H_{a,SB}$, where:

$$u_a e^{-iH\delta t} u_a^\dagger = e^{-i(H_{a,X} + H_B + \lambda H_{a,SB})}. \quad (46)$$

The pulses do not change the locality of the system Hamiltonian or the interaction Hamiltonian (i.e., the locality of $H_{a,X}$ and $H_{a,SB}$ is the same as that H_X and H_{SB}). The effective Hamiltonian at order δt^m would be the result of $m - 1$ nested commutators of these H_a segments. At this order in δt , for any q , the terms of order λ^q are at most $[ql + (m - q)(k - 1)]$ -local. The most non-local terms are ml -local, and they appear at order λ^m (which is quite weak). We can have a $[m(k - 1) + 1]$ -local term at order λ^0 , which we use for the simulation.

In the case of DD, if we choose the $\{u_a\}$ pulses to commute with the system Hamiltonian we get $H_a = H_X + H_B + \lambda H_{a,SB}$. In this case, all the H_B and H_X terms commute with each other and the λ^0 term vanishes.

2. The case of our construction

For the case where the system Hamiltonian is 2-local and the interaction Hamiltonian is 1-local ($k = 2, l = 1$), errors of order λ^q for $\forall q \geq 1$ are $[ql + (m - q)(k - 1)] = m$ -local, and at order λ^0 the simulated Hamiltonian is $(m + 1)$ -local. So at order δt^3 ($m = 3$) we have 3-local noise and the 4-body simulated Hamiltonian. (See Appendix E for an estimate of the number of error terms.)

In our construction, we use $N_{sim}N_{DD}$ pulses to simulate hH_p at order δt^3 with errors also at order δt^3 , with the errors of leading order λ^2 . (These terms are at most 3-local, and there are at most $N_{sim} \times N_{DD}^3 \times (3N)^2$ such

terms.) The leading order terms in the simulation error are of order δt^4 and of order λ^0 . (These terms are at most 5-local, and there are $O(N_{sim}^4)$ such terms.)

3. Geometrically local bath

It is reasonable to assume that the dynamics of the bath and the way it interacts with the system can be characterized by some local Hamiltonians. Assuming locality for the bath part of the interaction Hamiltonian and also the bath Hamiltonian, one can show that the stronger noise terms (lower powers of λ) are more geometrically local. For example this means that the 3-local errors showing up in δt^3 are more concentrated in a geometrically local region of the lattice, rather than showing up in any three locations. The surface code Hamiltonian (and topological codes in general) are expected to perform better if the errors are more geometrically correlated.

Now we assume locality for the bath part of the interaction Hamiltonian. We assume that the terms B_i^α and B_j^β from the interaction Hamiltonian commute with each other if the corresponding system qubits are at least distance r from each other.

$$\forall i, j \quad d(i, j) \geq r \Rightarrow [B_\alpha^i, B_\beta^j] = 0. \quad (47)$$

Again, looking at the nested commutators (we will add the effect of the bath Hamiltonian later), we see that errors of order λ consist of a single term from H_{SB} commuted with one or more terms from the system Hamiltonian. This error term therefore can be $[l + (m - 1)(k - 1)]$ -local, but all the qubits it acts on are neighbors. In general, $[ql + (m - q)(k - 1)]$ -local errors in λ^q can appear with the qubits separated by a distance of at most $(q - 1)r$, and the geometrically local terms coming from the system Hamiltonian can only grow from this base of qubits.

We need to also include the effects of the bath Hamiltonian. H_B does not itself increase the non-locality of the interaction Hamiltonian terms, but it can connect terms in different local regions and so affect the spread of error. One can separately assume geometric locality for the bath Hamiltonian and repeat the argument above to bound the spread of errors. Namely, we may assume that $H_B = \sum h_B$, where each term can have a nontrivial effect on the bath part of H_{SB} corresponding to qubits separated by a distance of at most r' :

$$\begin{aligned} \forall h_B, d(i, j) \geq r' : \\ \Rightarrow [B_\alpha^i, h_B] = 0 \text{ or } [B_\alpha^j, h_B] = 0. \end{aligned} \quad (48)$$

If there are b copies of H_B in the m nested commutators, at order λ^d this results results in at most $[dl + (m - d - b)(k - 1)]$ -local term that can spread up to a distance of $br' + (d - 2)r$ with $d \geq 2$.

E. DD pulses for a local bath

Suppose we want to protect N qubits on a line from any possible multiqubit error. The usual twirling lemma would suggest applying all 4^N possible pulses. One can ask: what if each error term is supported on at most l neighboring qubits? Here we discuss another symmetrization lemma to lower the number of pulses to just 4^l multiqubit pulses, independent of N (benefiting from the parallelism in applying pulses).

The construction of the pulses is as follows: list all 4^l possible Pauli operators on the first l qubits. Now extend each pulse by periodically repeating the Pauli operator on each successive set of l qubits such that each of the pulses has the same Pauli on qubits i and $i + l$.

A special case is 1-local noise on N qubits ($l = 1$, so the period is one). As we saw we can average out all the errors by considering just four pulses: $I^{\otimes N}, X^{\otimes N}, Y^{\otimes N}, Z^{\otimes N}$.

Clearly we can similarly generalize this construction to higher dimensional lattices:

Lemma 2. (Symmetrization lemma: Local noise) *Assume that the support of each error term on a D -dimensional regular lattice is contained in a hypercube of size l^D qubits. Then with 4^{l^D} pulses, independent of the size of the lattice, all the error terms can be averaged out.*

Applying this result to our 2-dimensional square lattice, we only need to construct the sequences by considering all possible Pauli operators on the qubits in squares of length l (there are 4^{l^2} corresponding pulses), and then cover the lattice with parallel use of these patterns.

For example, for $r = 1$ (bath operators from the interaction Hamiltonian, B_i^α and B_j^β , commute unless qubits i and j are nearest neighbors) all the errors of order δt^3 are strings of 3 Pauli operators. We can define a square of 3×3 qubits, and apply all Pauli operators that commute with all the XX interactions in that square (8 independent interaction of such type). At most $4^9/2^8$ steps are needed (all the possible Pauli's operators on 9 qubits that commute with 8 independent Pauli terms).

Therefore, assuming locality for the bath, all error terms of order δt^3 can be removed with a number of pulses independent of the size of the grid. Of course this is a worst case analysis (assuming that any kind of three-Pauli error can be generated), but when needed one can work out the details of the generated errors and find the pulses to remove them, resulting in much shorter sequences.

VI. ERROR SUPPRESSION

Now we analyze the effect of the simulated energy gap on the noise. For simplicity, rather than adding the simulation of the desired computation to this picture, we just

repeat the process of simulating $hH_p\delta t^3$ a total of k times (a quantum memory). Each repetition of the simulation represents a time step $\Delta t = \delta t^3$. We recall that the combination of DD and simulation described earlier generates the following effective Hamiltonian in one cycle:

$$e^{-i[hH_p + H_B/\delta t^2 - gV(\delta t)]\Delta t}, \quad (49)$$

where h is the strength of the simulated Hamiltonian and $V(\delta t) = \sum_{a=0}^{\infty} V_a \delta t^a$ is the effective error Hamiltonian. Each term V_a can be decomposed according to the location of the system part: $V_a = \sum_{i,a} V_i^a$, and g is the overall scale [normalizing $V(\delta t)$]. The ratio between h and g quantifies the energy penalty as we will see later.

We can think of the effective Hamiltonian as a constant Hamiltonian $hH_p + H_B/\delta t^2 - gV(\delta t)$, that is on for time $k\Delta t$. The goal is to see how much error suppression we get from this process, assuming that we start in the ground subspace. To do so we bound the quantity:

$$\|U(k\Delta t)P - U_{\text{ideal}}(\Delta t)P\|, \quad (50)$$

where P is the projector onto the codespace, and $U(t)$ is the evolution generated by $H(t) = hH_p(t) + H_B/\delta t^2 + V(t)$. This difference bounds how much the noisy evolution can make the state deviate from the ideal evolution represented by $U_{\text{ideal}}(t) \equiv e^{-i(hH_p + H_B/\delta t^2)kt}$.

To bound this difference, we start by moving to the interaction picture defined by the base Hamiltonian $hH_p + H_B/\delta t^2$. We denote the evolution corresponding to this base Hamiltonian by $U_P(t)$. Moving to the interaction picture with respect to the base Hamiltonian we have:

$$V_I(t) = U_P^\dagger(t)V(t)U_P(t), \quad (51)$$

$$U_I(t) = U_P^\dagger(t)U(t), \quad (52)$$

where

$$i\dot{U}_I(t) = V_I(t)U_I(t). \quad (53)$$

Integrating this we have

$$U_I^\dagger(k\Delta t) = I + i \int_0^{k\Delta t} U_I^\dagger(t)V_I(t)dt. \quad (54)$$

We note that

$$\begin{aligned} & \|U(k\Delta t)P - U_P(k\Delta t)P\| \\ &= \|(I - U_I^\dagger(k\Delta t))P\| \\ &= \left\| \int_0^{k\Delta t} \frac{dU_I^\dagger(t)}{dt} dt P \right\| \\ &= \left\| \int_0^{k\Delta t} U_I^\dagger(t)V_I(t)dt P \right\|. \end{aligned} \quad (55)$$

Following [44], we can define:

$$F(t) = \int_0^t V_I(\tau)d\tau P. \quad (56)$$

Integrating by parts, Eq. (55) becomes:

$$\begin{aligned}
& \|U(k\Delta t)P - U_P(k\Delta t)P\| \\
&= \left\| U_I^\dagger(k\Delta t)F(k\Delta t) - i \int_0^{k\Delta t} U_I^\dagger(\tau)V_I(\tau)F(\tau)d\tau \right\| \\
&\leq \|F(k\Delta t)\| + \left\| \int_0^{k\Delta t} U^\dagger(\tau)V(\tau)U_P(\tau)F(\tau)d\tau \right\| \\
&\leq \|F(k\Delta t)\| + \int_0^{k\Delta t} \|V(\tau)\| \|F(\tau)\| d\tau, \quad (57)
\end{aligned}$$

where we used the triangle inequality and the unitary invariance of the operator norm.

$F(t)$ quantifies the averaging out of the interaction Hamiltonian by the rotations induced by the penalty Hamiltonian (in the interaction picture), and the reduction of this term implies that the total evolution becomes closer to the ideal evolution (the difference between the actual and ideal evolutions goes to zero if $\|F\|$ goes to zero [44, 45]).

To bound this term, we can evaluate:

$$F(t') = g \int_0^{t'} U_P^\dagger(\tau) \sum_{i,a} V_i^a \delta t^a U_P(\tau) P d\tau. \quad (58)$$

We denote by c_i^a the number of stabilizer generators in the simulated Hamiltonian H_p that anticommute with the error V_i^a . Using this, and also the fact that $H_p P = \epsilon_0 P$, we have:

$$\begin{aligned}
F(t') &= g \int_0^{t'} e^{+i(H_B/\delta t^2)\tau} \\
&\quad \times \sum_{i,a} V_i^a \delta t^a e^{-i(H_B/\delta t^2)\tau} e^{-i2h\epsilon_0 c_i^a \tau} P d\tau. \quad (59)
\end{aligned}$$

Breaking this sum into two parts depending on whether c_i^j is zero or nonzero, and then integrating by parts, we get:

$$\begin{aligned}
& \|F(k\Delta t)\| \leq \\
& \frac{g}{h} \sum_{\substack{i,a \\ c_i^a \neq 0}} \frac{1}{c_i^a(2\epsilon_0)} (2 \|V_i^a \delta t^a\| + k\Delta t \| [V_i^a, H_B] \delta t^{a-2} \|) \\
& + gk\Delta t \left\| \sum_{\substack{i,a \\ c_i^a = 0}} V_i^a \delta t^a \right\|. \quad (60)
\end{aligned}$$

It is important to note that the first term, with the detectable errors (with $c_i^a \geq 1$), is suppressed by a factor of g/h . So for large values of h , we get error suppression, and the total evolution becomes closer to the one with no error. This error suppression is similar to the error suppression we get from an ideal implementation of H_p (rather than its simulation). This becomes clearer when we perform a similar calculation for the Hamiltonian H_p with energy penalty E_p , in the presence of a local system-bath interaction Hamiltonian λH_{SB} for a duration of T_0 .

Similar to the simulation case, using Eq. (57), we can bound

$$\begin{aligned}
& \|e^{-i(E_p H_p + H_B + \lambda H_{SB})T_0} P - e^{-i(E_p H_p + H_B)T_0} P\| \\
& \leq \|F_{\text{ideal}}(T_0)\| + \int_0^{T_0} \|\lambda H_{SB}\| \|F_{\text{ideal}}(\tau)\| d\tau, \quad (61)
\end{aligned}$$

where we have defined $F_{\text{ideal}}(t)$ as:

$$\begin{aligned}
F_{\text{ideal}}(t) &= \int_0^t e^{+i(E_p H_p + H_B)\tau} \\
&\quad \times (\lambda H_{SB}) e^{-i(E_p H_p + H_B)\tau} P d\tau. \quad (62)
\end{aligned}$$

Decomposing the interaction Hamiltonian according to the location of the system part, $H_{SB} = \sum_i h_{SB}^i$, we get:

$$\begin{aligned}
& \|F_{\text{ideal}}(t)\| \leq \\
& \frac{\lambda}{E_p} \sum_i \frac{1}{e^{i(2\epsilon_0)}} (2 \|h_{SB}^i\| + t \| [h_{SB}^i, H_B] \|), \quad (63)
\end{aligned}$$

where e^i denotes the number of stabilizer generators in the ideal H_p that anticommute with the error term h_{SB}^i . In this case, by construction, we always have $e^i \neq 0$.

This shows that the bound in Eq. (61) actually contains a suppression factor of λ/E_p . Clearly, in the simulated case, g/h plays the role of λ/E_p .

In contrast to the ideal case where $\forall i : e^i \neq 0$, in the simulation not all the terms of the effective error are suppressed by the effective Hamiltonian. For the chosen H_p , the only errors that commute with all the stabilizers (and so have $c_i^a = 0$) are (I) loops of X around A_v operators and loops of Z around B_p operators; and (II) logical operators, i.e., chains connecting boundaries. These errors are not suppressed by this mechanism, as can be seen from the second term in the bound.

Fortunately, the type I errors are not (too) destructive, as they are just product of the stabilizers of the code. So, these terms in the effective error do not cause logical errors. The only effect they can have is changing the strength of each simulated stabilizer slightly (changing h to $h \pm \delta h$). The first error of this type is 4-local and so occurs at order $O(\delta t^4)$. Comparing to the Hamiltonian itself, which occurs at order $O(\delta t^3)$, it is at least a factor of order δt weaker. Also, the effect of these errors is expected to average out, as the sign of the errors changes, and so the effective δh should be small.

The situation is different for the logical errors. If the distance of the code is d , these errors happen in $O(\delta t^d)$ or higher, which is small for large d . While this error is small, to have an arbitrary long computation it is necessary to correct possible errors before they accumulate into logical errors. This is done by active error correction.

VII. SUMMARY AND CONCLUSIONS

High-weight Hamiltonians are frequently used in designing quantum algorithms, especially when the goal is

to provide protection against noise. Implementing such high-weight Hamiltonians is experimentally challenging. One approach is to simulate such interactions using resources that are easier to implement. However, this simulation procedure itself can spread the noise and convert it to some effective geometrically correlated noise on the simulated system.

In this work we proposed combining techniques from dynamical decoupling and quantum simulation to simulate high-weight Hamiltonians such that the simulated Hamiltonians are stronger than the new effective noise. The ratio of the strength between the simulated Hamiltonian and the effective noise depends on the strength of the original noise on the resources used and also on the specific type of DD and simulation techniques applied. The spread of the effective noise on the simulated system depends on the locality of the bath and the system-bath interaction Hamiltonians. The reasonable assumption that these Hamiltonians are geometrically local, guarantees that the dominant terms in the effective noise Hamiltonian spread in a geometrically local region. Topological codes are expected to perform well in the presence of these types of geometrically correlated errors. For this reason we chose the surface code Hamiltonian and showed how to simulate the time dependent deformation of this Hamiltonian to perform universal quantum computation. Our analysis provided the details specific to this Hamiltonian, but the method is general. Similar to any other scheme performing universal quantum computation on a 2D grid, active error correction is necessary to guarantee fault tolerance. But as we showed explicitly, an energy gap is maintained during the simulation against the strongest errors generated in the process. The presence of this energy gap reduces the number of cycles of active error correction necessary during the simulation of the computation.

We expect that the methods proposed in this work can also be used to simulate the Hamiltonians that use subsystem codes to reduce the non-locality, either for surface code Hamiltonians [46] or general Hamiltonian-based quantum computation [45].

An interesting problem for future work is to consider other types of resources and simulation methods. One example is to consider simulation using perturbative gadgets and investigating methods to reduce the effect that local noise on a Hamiltonian has on the effective Hamiltonian in the low energy spectrum.

In all the constructions in this work, we only used the simplest form of DD and simulation techniques. One can expect to gain performance improvements by using more complex DD pulses and simulation techniques. It then becomes more important to consider the effect of the imperfection and noise on the pulse sequences and their timing.

More generally, it is interesting to design methods that are natively optimized to generate the largest ratio between the strength of the simulated system Hamiltonian and the strength of the effective noise.

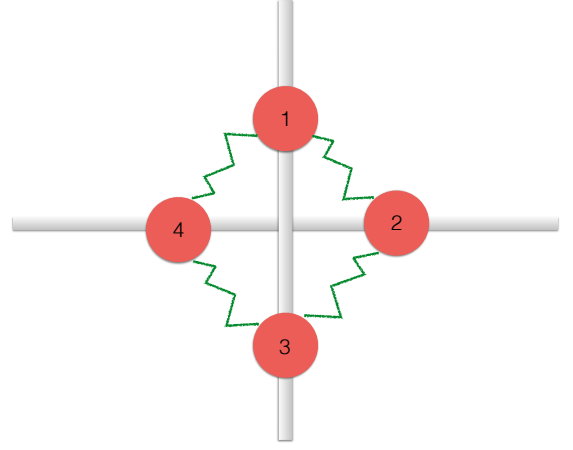


FIG. 7. Labeling of qubits connected to a vertex

ACKNOWLEDGMENTS

The authors acknowledge useful conversations with Yi-Cong Zheng. This work was supported under ARO MURI Grant Nos. W911NF-11-1-0268 and W911NF-15-1-0582; by NSF Grant No. CCF-1421078; and by an IBM Einstein Fellowship at the Institute for Advanced Study.

Appendix A: Alternative Connectivity

An alternative connectivity for the Hamiltonian on the square grid is when there is only nearest-neighbor coupling. In this case the Hamiltonian would become:

$$H_X = \sum_{d(i,j)=1} X_i X_j. \quad (\text{A1})$$

Here we show how to use the Hamiltonian $H_x = (X_1 + X_3)(X_2 + X_4)$ to simulate the Hamiltonian $X_1 X_2 X_3 X_4$ on qubits connected to a vertex, for an effective time Δt . (See Figure 7.) We denote the Hadamard gate by W , and the Phase gate by S .

Now,

$$(S_1 S_2) e^{-i H_X \delta t} (S_1 S_2) = e^{-i (Y_1 + X_3)(Y_2 + X_4) \delta t} \quad (\text{A2})$$

generates $H_a = (Y_1 + X_3)(Y_2 + X_4)$. Likewise,

$$(W_1 S_2) e^{-i H_X \delta t} (W_1 S_2) = e^{-i (Z_1 + X_3)(Y_2 + X_4) \delta t} \quad (\text{A3})$$

generates $H_b = (Z_1 + X_3)(Y_2 + X_4)$, and

$$(W_2) e^{-i H_X \delta t} (W_2) = e^{-i (X_1 + X_3)(Z_2 + X_4) \delta t} \quad (\text{A4})$$

generates $H_c = (X_1 + X_3)(Z_2 + X_4)$.

All of these Hamiltonians can be negated using $X_2 Z_4$ pulses. Using the commutator method given in Eq. (5), we will have the the desired Hamiltonian $8X_1 X_2 X_3 X_4$ plus some extra terms. Again, by doubling the number of pulses and conjugating the new ones by $Z_1 Y_2$, we can get

rid of all the extra terms, and so the desired Hamiltonian becomes

$$e^{-i16X_1X_2X_3X_4\delta t^3} + O(\delta t^4).$$

So using $N_X = 20$ pulses, and applying the Hamiltonian H_x for a time $N_X\delta t$ we can simulate $16X_1X_2X_3X_4$ at order δt^3 with errors of order δt^4 .

Generating this Hamiltonian on a grid can be done using the methods from Sec. IV with slight modifications.

Appendix B: Magnus expansion for piecewise constant Hamiltonian

A good reference introducing the Magnus expansion is [12]. Here are a few low-order terms for piecewise constant Hamiltonians [47]:

$$e^{-iH_n\delta t} \dots e^{-iH_2\delta t} e^{-iH_1\delta t} = e^{-iT_n H_{\text{eff}}(T_n)}. \quad (\text{B1})$$

Here $T_n = n\delta t$ and $H_{\text{eff}} = \sum_{k=0}^{\infty} H_{\text{eff}}^{(k)}$, with

$$H_{\text{eff}}^{(0)}(T_n) = \frac{\delta t}{T_n} \sum_{k=1}^n H_k, \quad (\text{B2})$$

$$H_{\text{eff}}^{(1)}(T_n) = -\frac{i(\delta t)^2}{2T_n} \sum_{l=2}^n \sum_{k=1}^{l-1} [H_l, H_k], \quad (\text{B3})$$

$$H_{\text{eff}}^{(2)}(T_n) = -\frac{(\delta t)^3}{6T_n} \left\{ \sum_{m=3}^n \sum_{l=2}^{m-1} \sum_{k=1}^{l-1} [H_m, [H_l, H_k] + [H_m, H_l], H_k] + \frac{1}{2} \sum_{l=2}^n \sum_{k=1}^{l-1} [H_l, [H_l, H_k] + [H_l, H_k], H_k] \right\}. \quad (\text{B4})$$

Appendix C: Proof of lemma 1

Lemma 1. (Symmetrization lemma: Protecting interaction) *Let P_1, \dots, P_n be commuting Pauli operators, and let \mathcal{P} be the group generated by these Pauli operators. Denote the set of all Pauli operators that commute with all the P_i s by $\mathcal{N}(\mathcal{P})$ (the normalizer of \mathcal{P}). Then for any Pauli operator s ,*

$$\forall s \notin \mathcal{P} : \sum_{g \in \mathcal{N}(\mathcal{P})} g s g^\dagger = 0, \quad (\text{C1})$$

$$\forall s \in \mathcal{P} : \sum_{g \in \mathcal{N}(\mathcal{P})} g s g^\dagger = s |\mathcal{N}(\mathcal{P})|. \quad (\text{C2})$$

Proof. The $s \in \mathcal{P}$ case is trivial. Assume $s \notin \mathcal{P}$ and $s \notin \mathcal{N}(\mathcal{P})$, which means that s anticommutes with at least one element of \mathcal{P} and of $\mathcal{N}(\mathcal{P})$. Denoting a set of independent generators of \mathcal{P} by $\{P_i\}_{i=1}^m$, there exists a (non-identity) P_* in this set such that $\{P_*, s\} = 0$. Obviously $\forall g \in \mathcal{N}(\mathcal{P}) \Rightarrow P_* g \in \mathcal{N}(\mathcal{P})$. From this we conclude

that g and $P_* g$ are two distinct elements of $\mathcal{N}(\mathcal{P})$ such that one commutes with s , and the other one anticommutes with it.² Going through all the elements of $\mathcal{N}(\mathcal{P})$ (and using $P_*^2 = I$), half the elements commute with s while the other half anticommute with s , and so the result holds in this case.

For the case that $s \notin \mathcal{P}$ and $s \in \mathcal{N}(\mathcal{P})$: elements of $\mathcal{N}(\mathcal{P})$ can be generated using $\{P_i\}$ and $n - m$ extra pairs of conjugate generators, where n is the number of qubits. Let us call these pairs (\hat{X}_j, \hat{Z}_j) , where $1 \leq j \leq n - m$. Note that $s \notin \mathcal{P}$ and $s \in \mathcal{N}(\mathcal{P})$ means that s contains at least one element from these pairs, let us say an element from the j_* -th pair. Now we notice that $\forall g \in \mathcal{N}(\mathcal{P}) \Rightarrow \{g, g\hat{X}_{j_*}, g\hat{Z}_{j_*}, g\hat{X}_{j_*}\hat{Z}_{j_*}\} \in \mathcal{N}(\mathcal{P})$. All these elements are distinct, and two of them commute with s while two anticommute. Therefore, again the elements of $\mathcal{N}(\mathcal{P})$ can be partitioned into half commuting and half anticommuting, which completes the proof. \square

Appendix D: The number of pulses to average out multiqubit errors grows with the number of qubits

A simple example showing that the number of pulses to average out multi-qubit errors grows with the number of qubits is the following set of two-qubit errors:

$$\{Y_i Y_j : 1 \leq i < j \leq N\} \quad (\text{D1})$$

A simple lower bound on the number of pulses needed to average out all of these errors can be derived by noticing that each Y_i has to have a distinct pattern of commutation/anticommutation with the pulses: if Y_i and Y_j have the same pattern, then all the pulses will commute with $Y_i Y_j$, so applying pulses will leave this term unchanged. From this we conclude that number of distinct patterns has to be greater than N , which means that the number of pulses needed to average out all the error terms has to increase with the number of qubits. If $|P|$ is the number of pulses, this means that $2^{|P|} \geq N$, and so $|P| \geq \log_2 N$. For combinatorial approaches using $O(N)$ pulses to achieve first-order decoupling see Refs. [48, 49] and chapter 15 of Ref. [2].

Appendix E: Estimating the Number of Error Terms

We provide a worst-case estimate of the number of error terms that occur at order δt^m while having the simulated Hamiltonian appear at order δt^{m+1} .

Using our construction, we first apply N_{DD} pulses to protect the system Hamiltonian and push the errors to

² There are two cases depending on whether s commutes or anticommutes with g . If $\{s, g\} = 0, \{s, P_*\} = 0$ then $P_* g s = -P_* s g = s P_* g$, i.e., $[s, P_* g] = 0$. Likewise, $[s, g] = 0, \{s, P_*\} = 0 \Rightarrow \{s, P_* g\} = 0$.

order δt^m . At this order there will be roughly N_{DD}^m terms of $m-1$ nested commutators, with each term of the form $H_i = H_X + H_B + \lambda H_{i,SB}$. At order λ^0 there is no error, as all terms of H_X and H_B commute with each other. At order λ^1 there are $3N$ terms (N being the number of qubits, and $3N$ being the number of terms), multiplied by a constant depending on the connectivity degree of H_X , as the entangling Hamiltonian can only expand an existing string of errors locally to neighboring qubits. (The same argument is true for higher powers of λ .) Thus, at order λ^1 the total number of terms is of order $N_{\text{DD}}^m \times 3N$.

Notice the growth with the number of qubits. At order λ^q there are $N_{\text{DD}}^m (3N)^q$ terms. (Recall that we can remove all the errors of order λ^1 by doubling the number of pulses.)

Therefore, to simulate the desired nonlocal Hamiltonian at order $\delta t^{m'}$, we apply N_{sim} of these sequences of pulses consecutively. The number of error terms generated by the simulation process at order $\delta t^{m'+1}$ is $N_{\text{sim}}^{m'}$, which as we saw earlier can be $[m'(k-1)+1]$ -local at order λ^0 . Also, the number of error terms resulting from DD will be multiplied by N_{sim} .

-
- [1] P. W. Shor, [Proceedings of 37th Conference on Foundations of Computer Science](#), 56 (1996).
 - [2] D. A. Lidar and T. A. Brun, *Quantum error correction* (Cambridge University Press, 2013).
 - [3] D. Aharonov and M. Ben-Or, [SIAM Journal on Computing](#) **38**, 1207 (2008).
 - [4] A. Y. Kitaev, [Russian Mathematical Surveys](#) **52**, 1191 (1997).
 - [5] E. Knill, R. Laflamme, and W. H. Zurek, *Science* **279**, 342 (1998).
 - [6] D. Gottesman, *Chaos, Solitons & Fractals* **10**, 1749 (1999).
 - [7] E. Knill, *Nature* **434**, 39 (2005).
 - [8] P. Aliferis, D. Gottesman, and J. Preskill, [Quantum Info. Comput.](#) **6**, 97 (2006).
 - [9] B. W. Reichardt, in *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part I*, ICALP'06 (Springer-Verlag, Berlin, Heidelberg, 2006) pp. 50–61.
 - [10] D. Aharonov, A. Kitaev, and J. Preskill, *Phys. Rev. Lett.* **96**, 050504 (2006).
 - [11] H. K. Ng and J. Preskill, *Phys. Rev. A* **79**, 032318 (2009).
 - [12] H. K. Ng, D. A. Lidar, and J. Preskill, *Phys. Rev. A* **84**, 012305 (2011).
 - [13] D. Gottesman, [Quantum Info. Comput.](#) **14**, 1338 (2014).
 - [14] P. Zanardi and M. Rasetti, *Phys. Lett. A* **264**, 94 (1999).
 - [15] A. Y. Kitaev, *Annals of Physics* **303**, 2 (2003).
 - [16] Y.-C. Zheng and T. A. Brun, *Phys. Rev. A* **91**, 022302 (2015).
 - [17] C. Cesare, A. J. Landahl, D. Bacon, S. T. Flammia, and A. Neels, *Phys. Rev. A* **92**, 012336 (2015).
 - [18] S. Bravyi and B. Terhal, *New Journal of Physics* **11**, 043029 (2009).
 - [19] A. Mizel and D. A. Lidar, *Phys. Rev. Lett.* **92**, 077903 (2004).
 - [20] N. Chancellor, S. Zohren, and P. A. Warburton, [npj Quantum Information](#) **3**, 21 (2017).
 - [21] S. Puri, C. K. Andersen, A. L. Grimsmo, and A. Blais, [arXiv:1609.07117](#) (2016).
 - [22] C. Herdman, K. C. Young, V. Scarola, M. Sarovar, and K. Whaley, *Phys. Rev. Lett.* **104**, 230501 (2010).
 - [23] K. C. Young, M. Sarovar, J. Aytac, C. Herdman, and K. B. Whaley, *Journal of Physics B: Atomic, Molecular and Optical Physics* **45**, 154012 (2012).
 - [24] D. A. Lidar, D. Bacon, J. Kempe, and K. B. Whaley, *Phys. Rev. A* **63**, 022307 (2001).
 - [25] J. L. Dodd, M. A. Nielsen, M. J. Bremner, and R. T. Thew, *Phys. Rev. A* **65**, 040301 (2002).
 - [26] S. P. Jordan and E. Farhi, *Phys. Rev. A* **77**, 062329 (2008).
 - [27] Y. Cao and D. Nagaj, [Quantum Info. Comput.](#) **15**, 1197 (2015).
 - [28] Y. Subaşı and C. Jarzynski, *Phys. Rev. A* **94**, 012342 (2016).
 - [29] W. Dür, M. J. Bremner, and H. J. Briegel, *Phys. Rev. A* **78**, 052325 (2008).
 - [30] J. I. Cirac and P. Zoller, *Nature Physics* **8**, 264 (2012).
 - [31] L. Viola and S. Lloyd, *Phys. Rev. A* **58**, 2733 (1998).
 - [32] L. Viola, E. Knill, and S. Lloyd, *Phys. Rev. Lett.* **85**, 3520 (2000).
 - [33] S. P. Jordan, E. Farhi, and P. W. Shor, *Phys. Rev. A* **74**, 052322 (2006).
 - [34] M. Marvian and D. A. Lidar, *Phys. Rev. A* **95**, 032302 (2017).
 - [35] P. Zanardi, *Phys. Lett. A* **258**, 77 (1999).
 - [36] S. B. Bravyi and A. Y. Kitaev, [arXiv:quant-ph/9811052](#) (1998).
 - [37] D. Poulin, A. Qarry, R. Somma, and F. Verstraete, *Phys. Rev. Lett.* **106**, 170501 (2011).
 - [38] G. A. Paz-Silva and D. A. Lidar, *Sci. Rep.* **3** (2013).
 - [39] K. Khodjasteh and D. A. Lidar, *Phys. Rev. A* **78**, 012355 (2008).
 - [40] J. R. West, D. A. Lidar, B. H. Fong, and M. F. Gyure, *Phys. Rev. Lett.* **105**, 230503 (2010).
 - [41] M. Marvian, T. A. Brun, and D. A. Lidar, in preparation.
 - [42] L. Viola, S. Lloyd, and E. Knill, *Phys. Rev. Lett.* **83**, 4888 (1999).
 - [43] M. M. Wilde, *Quantum information theory* (Cambridge University Press, 2013).
 - [44] A. D. Bookatz, E. Farhi, and L. Zhou, *Phys. Rev. A* **92**, 022317 (2015).
 - [45] M. Marvian and D. A. Lidar, *Phys. Rev. Lett.* **118**, 030504 (2017).
 - [46] S. Bravyi, G. Duclos-Cianci, D. Poulin, and M. Suchara, [Quantum Info. Comput.](#) **13**, 963 (2013).
 - [47] L. F. Santos and L. Viola, *New Journal of Physics* **10**, 083009 (2008).
 - [48] P. Wocjan, M. Rötteler, D. Janzing, and T. Beth, *Phys. Rev. A* **65**, 042309 (2002).
 - [49] P. Wocjan, *Phys. Rev. A* **73**, 062317 (2006).