Simulation of rare events in quantum error correction

Sergey Bravyi\textsuperscript{1} and Alexander Vargo\textsuperscript{1}

\textsuperscript{1}IBM T.J. Watson Research Center, Yorktown Heights NY 10598

(Dated: May 11, 2014)

We consider the problem of calculating the logical error probability for a stabilizer quantum code subject to random Pauli errors. To access the regime of large code distances where logical errors are extremely unlikely we adopt the splitting method widely used in Monte Carlo simulations of rare events and Bennett’s acceptance ratio method for estimating the free energy difference between two canonical ensembles. To illustrate the power of these methods in the context of error correction, we calculate the logical error probability $P_L$ for the 2D surface code on a square lattice with a pair of holes for all code distances $d \leq 20$ and all error rates $p$ below the fault-tolerance threshold. Our numerical results confirm the expected exponential decay $P_L \sim \exp \left[ -\alpha(p)d \right]$ and provide a simple fitting formula for the decay rate $\alpha(p)$. Both noiseless and noisy syndrome readout circuits are considered.

PACS numbers:

I. INTRODUCTION

Quantum error correction holds the promise of extending the coherence time of quantum devices by utilizing redundant encoding of information. Like its classical counterpart, quantum error correction enables reliable storage of encoded quantum states in the presence of noise by monitoring parity check violations and applying suitable recovery operations. Furthermore, many quantum codes support a limited set of logical operations that can be applied to encoded states without exposing them to noise. Extensive theoretical work rigorously confirmed the feasibility of large-scale fault-tolerant quantum computing for a wide range of noise models [1–5].

Several families of quantum codes have been proposed as candidates for scalable fault-tolerant architectures, including concatenated codes [6–8], surface codes [9–13], surface codes with twists [14], color codes [15, 16] and Turaev-Viro codes [17]. Each of these families contains an infinite sequence of codes labeled by a code distance $d$. The number of physical qubits and elementary operations required to implement a single logical gate using a distance-$d$ code typically grows polynomially as one increases $d$, whereas the probability of a logical error $P_L$ decreases exponentially, that is, $P_L \leq d^3 \cdot \exp (-\alpha d)$. Here $\alpha$ and $\beta$ are constant coefficients depending on the chosen family of codes, the noise model, and the decoding algorithm. Furthermore, the threshold theorem asserts that $\alpha > 0$ for a sufficiently small noise strength [3–5, 10, 15]. A problem essential for estimating the overhead associated with error correction is finding the minimum code distance that achieves the desired level of noise suppression. This requires a precise knowledge of the decay rate

$$\alpha = -\lim_{d \to \infty} \frac{1}{d} \log (P_L(d)),$$

since the exponential term gives the dominant contribution to $P_L$ for large code distances. The decay rate $\alpha$ is also a natural figure of merit for comparing the performance of different decoding algorithms.

The present paper describes a new algorithm for computing the decay rate $\alpha$ in the special case of the surface code family. We report numerical results for two commonly studied noise models corresponding to noiseless and noisy syndrome readout circuits. Prior to our work, several methods have been developed for computing the decay rate $\alpha$ of the surface codes, most notably Monte Carlo simulation [18] and fault path counting [10, 20]. Monte Carlo method attempts to estimate $P_L$ by generating many random error configurations and computing the fraction of trials that resulted in a logical error. This method, however, is not a viable option in the regime of small physical error rates and/or large code distances, where logical errors are extremely unlikely.

Fault path counting method adapted to surface codes by Dennis et al. [10] provides an upper bound on $P_L$ in the form of a weighted sum over self-avoiding walks on a suitably defined lattice. This can be translated to a lower bound on the decay rate $\alpha$, although the bound is not expected to be tight. A different version of the method, proposed by Fowler [20], enables exact computation of $P_L$ in the asymptotic regime of small error rates $p$. In this regime the dominant contribution to $P_L$ comes from minimum-weight uncorrectable errors that span $[d/2]$ physical locations [4]. Accordingly, if the limit $p \to 0$ is taken for a fixed code distance $d$, one can use an asymptotic formula $P_L = A_d p^{d/2}$, where $A_d$ is a constant coefficient. For relatively small values of $d$ one can compute $A_d$ by summing the probabilities of all minimum-weight uncorrectable errors, see Ref. [20]. This method, however, is not well-suited for computing the decay rate $\alpha$ which requires taking the limit $d \to \infty$ for a fixed error rate $p$.

Here we propose a new algorithm for estimating the logical error probability $P_L$ and the decay rate $\alpha$. It enables us, for the first time, to access the regime of large code distances and moderately small error rates, which we expect to be particularly important in the context of fault-tolerance. The key ingredients of our algorithm are the splitting method and Bennett’s acceptance ratio.
method \[21\]. The splitting method is a standard tool in Monte Carlo simulation of rare events, see for instance Ref. \[22\]. To compute the quantity \( P_L(p) \) for a given physical error rate \( p \) we choose a sufficiently dense monotonic sequence of error rates \( p_1, \ldots, p_t \), where \( p_t = p \) and \( p_1 \) is chosen such that \( P_L(p_1) \) can be computed efficiently by either the Monte Carlo simulation (in which case \( p_1 \) must be sufficiently large) or by the fault path counting method (in which case \( p_1 \) must be sufficiently small). We then employ the acceptance ratio method \[21\] to estimate the quantity \( R_j = P_L(p_{j+1})/P_L(p_j) \) for each \( j = 1, \ldots, t - 1 \). This is accomplished through a Metropolis-type subroutine for sampling uncorrectable errors from a specific probability distribution. The Metropolis subroutine is the most time consuming part of our algorithm and we propose several tricks for improving its efficiency. Finally, we find \( P_L(p) \) from the obvious identity \( P_L(p) = P_L(p_t) = R_1R_2 \cdots R_{t-1}P_L(p_1) \). Once \( P_L(p) \) has been computed for several choices of the code distances \( d \), the exponential fitting yields the decay rate \( \alpha(p) \). Depending on whether the sequence \( p_1, \ldots, p_t \) is increasing or decreasing we shall use a term upward or downward splitting. To the best of our knowledge, the idea of using the splitting method in the context of error correction was originally proposed by Wang, Harrington, and Preskill \[23\], see Section IV(D) of Ref. \[23\]. The use of the acceptance ratio method in this context appears to be new.

II. SUMMARY OF RESULTS

We begin by highlighting main features of the surface code and motivating its choice as a testing ground for the proposed algorithm. The surface code achieves fault-tolerance by repeatedly measuring syndromes of 4-qubit parity check operators on a 2D grid of physical qubits \[10\]. Logical qubits are introduced by creating special defect areas in the lattice in which the pattern of syndrome measurements is altered. By choosing a suitable syndrome readout schedule one can change the location and shape of defects in a way that simulates braiding, splitting, and fusion of topological charges \[43\]. By analogy with topological quantum computation \[9\], this enables fault-tolerant implementation of some logical gates such as the CNOT \[12\ \[13\ \[24\]. The code distance \( d \) is determined by the length of the shortest loop encircling a defect or the shortest path connecting some pair of defects. The code admits efficient decoding by Edmonds’s minimum weight matching algorithm \[10\ \[25\], renormalization group methods \[26\ \[28\], or by the Markov chain Monte Carlo algorithm \[29\], and features an error threshold close to 1% for the standard depolarizing noise model \[30\].

Let us now describe our numerical results. We consider the standard surface code on a square lattice \[9\ \[11\] with a pair of defects representing one logical qubit, see Fig. 1. A defect is defined as a square block of plaquettes removed from the lattice. Logical Pauli operators \( \mathcal{Z} \) and \( \mathcal{X} \) are products of Pauli \( Z \) and \( X \) over a loop encircling one of the defects and over a path on the dual lattice connecting the two defects respectively, see Fig. 1. The dimensions of the lattice in our simulations were chosen such that the code distance is \( d = 4r \), where \( r \) is the linear size of the defects. Accordingly, each defect is separated from the external boundary of the lattice and from the other defect by distance \( 4r \), see Fig. 4. Simulations were performed only for \( r \geq 2 \) to avoid finite size effects.

We first analyze a toy error model with a noiseless syndrome readout where every qubit is subjected to independent bit-flip and phase flip-errors, each applied with probability \( p \) (qubits on the external boundary of the lattice may be treated specially, see Section VII for details). Once all errors have been generated, one computes the error syndrome and finds the most likely recovery operator consistent with the syndrome, see Sections IV,V. A logical loop-like or path-like error occurs if the recovery operator differs from the actual error by a logical operator \( \mathcal{Z} \) or \( \mathcal{X} \) (modulo stabilizers). We are interested in the logical error probability \( P_L = P_L(p, r) \) for both loop-like and path-like logical errors. In the chosen geometry the code distance \( d \) can only be a multiple of four; thus, it will be more convenient to define the decay rate \( \alpha \) such that

\[
P_L(p, r) \sim \exp[-\alpha(p)r].
\]

It differs from the decay rate defined earlier by a factor of 4. The largest error rate \( p_{th} \) such that \( \alpha(p) > 0 \) for all \( p < p_{th} \) is known as the error threshold. For the chosen model \( p_{th} \approx 10.03\% \), see \[23\]. Our simulations were performed only for error rates \( p \leq 8\% \) to avoid finite-size effects which become important for \( p \approx p_{th} \).
The numerical results are presented in Fig. 2. The logical error probability $P_L$ is shown as a function of $p$ for a few small values of $r$. This provides enough data to perform the exponential fitting $P_L \sim \exp(-\alpha p r)$ and extract the decay rate $\alpha$. The plot of $\alpha$ as a function of $p$ is shown in Fig. 2. This figure also shows the estimate of $\alpha$ based on the asymptotic low-$p$ formula $P_L \approx A_d p^{d/2}$. It is clear that the latter provides a poor approximation of $\alpha$ for large and moderately small error rates, especially in the case of path-like errors. Since $\alpha(p_{th}) = 0$, one can easily extend the function $\alpha(p)$ to the interval $8\% \leq p \leq p_{th}$ using a linear interpolation. More details on the simulation methods can be found in Section VII.

The following heuristic ansatz was used to fit the numerical data for $P_L$:

$$P_L(p, r) \approx \exp(-\alpha(p)(r - r_0))P_L(p, r_0), \quad (2)$$

$$P_L(p, r_0) = \exp[2r_0 \log(p) + x(p)],$$

$$-\alpha(p) = c + 2\log(p) + \log(1 + y(p)).$$

Here $r_0 = 2$ is the smallest defect size in our simulations, $c$ is a constant coefficient, and the functions $x(p)$, $y(p)$ are low-degree polynomials such that $y(0) = 0$. This ansatz is a slightly refined version of the exponential scaling Eq. (1) that attempts to reproduce the pre-exponential factor. We fit $x(p)$ and $y(p)$ by the second and the third degree polynomials,

$$x(p) = \sum_{n=0}^{2} x_n p^n, \quad y(p) = \sum_{n=1}^{3} y_n x^n. \quad (3)$$

The terms proportional to $\log(p)$ in Eq. (2) are chosen to reproduce the asymptotic formula $P_L \sim A_d p^{d/2}$ in the limit $p \to 0$. The coefficients $c$, $x_i$ and $y_i$ found by fitting the numerical data are listed in Table I.

Next we analyze a more realistic noise model with a noisy syndrome readout. Here the syndromes of site and plaquette stabilizers are measured by applying a sequence of CNOT gates which couple the respective code qubits with ancillary qubits collecting the syndrome information, see Section VIII. Independent depolarizing errors occur with probability $p$ on each qubit preparation, measurement, and CNOT gate. Our noise model mostly coincides with the one introduced by Fowler in Ref. [12]. To enable reliable error correction, the syndrome readout is repeated $d$ times, where $d = 4r$ is the code distance. The lattice geometry is time-independent which corresponds to the logical identity gate (fault-tolerant storage). Once all syndromes have been generated, the most likely configuration of errors consistent with the syndrome and the corresponding recovery operator on code qubits are computed, see Section VII. As before, a logical loop-like or path-like error occurs if the recovery operator differs from the actual accumulated error on code qubits by a logical operator $(\overline{Z}$ or $\overline{X}$) (modulo stabilizers). We are interested in the logical error probability $P_L = P_L(p, r)$ for both loop-like and path-like logical errors. More precisely, we

![Graph 1](image1.png)

![Graph 2](image2.png)

**FIG. 2.** (Color online) Noiseless syndrome readout. The decay rate $\alpha(p)$ in the exponential scaling $P_L \sim \exp(-\alpha(p)r)$ for loop-like and path-like logical errors computed by the downward splitting method. The dashed line shows the estimate of $\alpha(p)$ based on the asymptotic low-$p$ formula $P_L \sim (4r)^{p^{d/2}}$, that is, $\alpha(p) = -4\log 2 - 2\log(p)$.

**FIG. 3.** (Color online) Noisy syndrome readout. The decay rate $\alpha(p)$ for loop-like and path-like logical errors computed by the downward splitting method. The dashed lines show the estimate of $\alpha(p)$ based on the low-$p$ asymptotic formula $P_L \sim A_p p^{d/2}$, that is, $\alpha(p) = -c - 2\log(p)$, see the last line of Eq. (2).

**TABLE I: Coefficients in the fitting formulas Eqs. (2,3).**

|      | Noiseless syndromes | Noisy syndromes |
|------|---------------------|-----------------|
|      | Loop errors | Path errors | Loop errors | Path errors |
| $x_0$ | 3.18 | 4.25 | 8.67 | 9.88 |
| $x_1$ | 24.2 | 64.6 | 405 | 1.17 $\times 10^3$ |
| $x_2$ | 19 | -347 | 5.83 $\times 10^3$ | -7.64 $\times 10^4$ |
| $c$ | 4 log 2 | 4 log 2 | 5.86 | 7.52 |
| $y_3$ | 12.2 | 193 | 551 | 880 |
| $y_2$ | 297 | -1570 | 3.27 $\times 10^4$ | 4.69 $\times 10^3$ |
| $y_1$ | 0 | 0 | 5.72 $\times 10^7$ | 6.04 $\times 10^6$ |
define $P_L$ as the logical error probability divided by the total number of syndrome readout steps, which is 4$r$ in our simulations. Estimates of the error threshold $p_{th}$ for a noisy syndrome readout vary between 0.75% and 0.9%, see [18, 25, 31]. To avoid finite size effects, simulations were performed only for $p \leq 0.7\%$ (as before, one can use linear interpolation to cover the interval between 0.7% and $p_{th}$). Our numerical results are presented in Figs. 3-6. The fitting curves in Fig. 6 represent the ansatz Eqs. (2,3) where the coefficients $a$, $x_i$, and $y_i$ are defined in Table I. As before, the numerically computed decay rate $\alpha$ is compared with its estimate based on the low-$p$ asymptotic formula $P_L \approx A_d p^{d/2}$. The coefficient $A_d$ was found by setting $x(p) = x(0)$ and $y(p) = 0$ in Eqs. (2). We observe that the asymptotic low-$p$ formula significantly overestimates $\alpha$ for large and moderately small error rates. More details on the simulation methods can be found in Section VII.

The limiting factor in our simulations was the running time of the Metropolis subroutine which grows rapidly as one increases the error rate, see Section VII for details. Accordingly, we were able to implement the splitting method only for sufficiently small errors rates, $p \leq p^*$, where $p^* \approx p_{th}/2$. Fortunately, the cutoff error rate $p^*$ was large enough to enable Monte Carlo computation of $P_L(p)$ for all $p \geq p^*$. In the case of noisy syndrome readout, we used downward splitting to access error rates $p \leq p^*$. For noiseless syndrome readout, where asymptotic low-$p$ formulas for $P_L(p)$ are readily available, we used upward splitting to access error rates $p \geq p^*$. Accordingly, in the latter case we were able to test the correctness of the splitting method by comparing the values of $P_L(p^*)$ found independently by the splitting and the Monte Carlo methods, see Fig. 5. For noisy syndrome readout, the correctness of the splitting method is partially confirmed by the fact that the function $P_L(p)$ computed by the two methods has the same derivative on both sides of $p^*$, see Fig. 6. A diagram illustrating the applicability region of each simulation method is shown in Fig. 4.

We hope that the new simulation techniques and the fitting formulas can find applications in the estimation of the fault-tolerance overhead for quantum computing architectures based on the surface code. Significant progress in this direction has been recently made in Refs. [18, 25, 31]. We hope that our results can refine estimates made in Refs. [25, 31] by replacing the asymptotic low-$p$ formulas for the logical error probability with more accurate approximations, such as the one defined in Eqs. (2,3) and Table I. We also hope that the fitting formulas Eqs. (2,3) have applicability beyond the toy noise models considered in this paper.

In the rest of this paper we introduce the general idea of the splitting method, see Section III and specialize it to the surface code settings, see Sections IV-VIII. It should be emphasized that at present the splitting method is a heuristic algorithm. Deriving rigorous bounds on its running time and the approximation error goes beyond the scope of this paper. Nevertheless, we make first steps in this direction in Section VII (see Lemmas 1-4). Some of our results such as the maximum weight matching decoding algorithm presented in Section VII and a decoder-independent definition of correctability, see Section VI might be interesting on their own right. We conclude by discussing some open problems in Section IX.

### III. RARE EVENT SIMULATION

This section provides some necessary background on the acceptance ratio method due to Bennett [21] and the splitting method. Let $\Omega$ be a finite set of events, $\mathcal{F} \subseteq \Omega$ be a subset of failure events, and $\pi$ be a normalized probability distribution that assigns a probability $\pi(E)$ to any event $E \in \Omega$. Our goal is to calculate the overall failure probability

$$\pi(\mathcal{F}) \equiv \pi(\mathcal{F}^c) \prod_{E \in \mathcal{F}} \pi(E).$$

For any probability distributions $\pi_1, \pi_2, \ldots, \pi_t$ such that $\pi_t \equiv \pi$ and such that $\pi_1(\mathcal{F})$ is known, one can use the obvious identity

$$\pi(\mathcal{F}) = \pi_1(\mathcal{F}) \prod_{j=1}^{t-1} \pi_{j+1}(\mathcal{F})/\pi_j(\mathcal{F}).$$

The splitting method attempts to compute $\pi(\mathcal{F})$ by evaluating each ratio $\pi_{j+1}(\mathcal{F})/\pi_j(\mathcal{F})$ in Eq. (4) separately. The method is applicable whenever the distributions $\pi_1, \ldots, \pi_t$ have the following properties:

(i) For any event $E \in \Omega$ the probability $\pi_j(E)$ can be computed efficiently.

(ii) There exists an efficient randomized algorithm $M_j$
that generates samples $E \in \mathcal{F}$ drawn from the conditional probability distribution $\pi_j(E|\mathcal{F}) = \pi_j(E)/\pi_j(\mathcal{F})$.

(iii) The conditional distributions $\pi_j(E|\mathcal{F})$ and $\pi_{j+1}(E|\mathcal{F})$ have a non-negligible overlap, as quantified below.

Given any function $f : \Omega \to \mathbb{R}$, we shall use the shorthand notation

$$\langle f \rangle_j = \frac{1}{\pi_j(\mathcal{F})} \sum_{E \in \mathcal{F}} \pi_j(E) f(E) \equiv \sum_{E \in \mathcal{F}} \pi_j(E|\mathcal{F}) f(E).$$

Then one can easily check that

$$\frac{\pi_{j+1}(\mathcal{F})}{\pi_j(\mathcal{F})} = C \frac{\langle g(C \pi_j/\pi_{j+1}) \rangle_j}{\langle g(C^{-1}\pi_j/\pi_{j+1}) \rangle_{j+1}}$$

for any constant $C > 0$ and any function $g : \mathbb{R} \to \mathbb{R}$ satisfying the “detailed balance” condition

$$g(x) = x^{-1} g(x^{-1}).$$

The expectation values needed to compute the righthand side Eq. (6) can be approximated by calling the algorithm $\mathcal{M}_j$ to generate sufficiently many samples $E_1, \ldots, E_N \in \mathcal{F}$.
\( \mathcal{F} \) drawn from the distribution \( \pi_j(E|\mathcal{F}) \) and using an estimate

\[
\langle g(C^{\pm 1}\pi_j/E) \rangle_j \approx \frac{1}{N} \sum_{a=1}^{N} g(C^{\pm 1}\pi_j(E_a)/\pi_j(1(E_a))).
\]  

As was shown in [21], for a fixed number of samples \( N \) the statistical error in Eq. 8 is minimized if one chooses

\[
g(x) = \frac{1}{1 + x}
\]

while the constant \( C \) satisfies

\[
\langle g(C\pi_j/E) \rangle_j = \langle g(C^{-1}\pi_j/E) \rangle_{j+1}.
\]  

In practice, both sides of Eq. (10) are replaced by their \( N \)-sample approximations (as defined in Eq. (6)); the resulting equation can then be solved for \( C \) to obtain the optimal value. This choice of \( g(x) \) and \( C \) guarantees that the ratio \( \pi_{j+1}(\mathcal{F})/\pi_j(\mathcal{F}) \) is estimated with a relative error \( \sigma_j \), where

\[
\sigma_j^2 = \frac{2}{N} \left[ \left( \sum_{E \in \mathcal{F}} \frac{2\pi_j(E|\mathcal{F})\pi_{j+1}(E|\mathcal{F})}{\pi_j(E|\mathcal{F}) + \pi_{j+1}(E|\mathcal{F})} \right)^{-1} - 1 \right],
\]

see Ref. [21]. Simple algebra shows that

\[
\sigma_j^2 \leq \frac{2}{N(1 - \|\pi_j(\cdot|\mathcal{F}) - \pi_j(\cdot|\mathcal{F})\|_1)},
\]

where \( \|p - q\|_1 \equiv (1/2) \sum_i |p_i - q_i| \) is the total variation distance. In particular, if \( \|\pi_j(\cdot|\mathcal{F}) - \pi_{j+1}(\cdot|\mathcal{F})\|_1 \leq 1 - \delta \) for all \( j = 1, \ldots, t - 1 \) then the overall failure probability \( \pi(\mathcal{F}) \) is estimated with a relative error

\[
\sigma \sim \sqrt{\frac{t}{\sqrt{\delta N}}}.
\]

(In fact, since different terms in Eq. 4 are computed independently, one may expect that the statistical errors accumulate in a random walk fashion).

**IV. SURFACE CODES**

We consider \( n \) physical qubits located on the edges of a square lattice \( \Sigma \) with open boundary conditions, possibly containing one or several defects, see Fig. 1 for an example. A defect is defined as a square block of plaquettes removed from the lattice. The sets of sites, edges, and plaquettes of the lattice are denoted \( \Sigma_0, \Sigma_1, \) and \( \Sigma_2 \) respectively. Given a subset of qubits \( E \subseteq \Sigma_1 \), let \( X(E) = \prod_{e \in E} X_e \) and \( Z(E) = \prod_{e \in E} Z_e \), where \( X_e \) and \( Z_e \) are the Pauli operators \( \sigma_x \) and \( \sigma_z \) acting on the qubit \( e \). For any site \( u \in \Sigma_0 \) and any plaquette \( f \in \Sigma_2 \) let \( A_u \subseteq \Sigma_1 \) be the set of edges incident to \( u \) and \( B_f \subseteq \Sigma_1 \) be the set of edges lying on the boundary of \( f \). Pauli operators \( X(A_u) \) and \( Z(B_f) \) are called stabilizers of the surface code. The stabilizers pairwise commute and have a common invariant subspace

\[
\mathcal{L} = \{ \psi \in (\mathbb{C}^2)^{\otimes n} : X(A_u)\psi = Z(B_f)\psi = \psi \ \forall u, f \}.
\]

The subspace \( \mathcal{L} \) is used to encode logical qubits. It is well-known that \( \dim(\mathcal{L}) = 2^{b_1} \), where \( b_1 \) is the first Betti number of the lattice. For open boundary conditions \( b_1 \) coincides with the number of defects. Below we only consider single and double defect geometries, \( b_1 = 1, 2 \). Given a Pauli error \( P \), the syndrome of \( P \) is defined as the set of all stabilizers anti-commuting with \( P \).

To describe the relationship between single-qubit errors and the corresponding syndromes it will be convenient to introduce a decoding graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \). There will be one decoding graph for each type of Pauli errors (bit-flip and phase-flip errors). Given a subset of edges \( E \subseteq \mathcal{E} \), let \( \partial E \subseteq \mathcal{V} \) be the set of vertices \( u \in \mathcal{V} \) having odd number of incident edges from \( E \).

Let us start with phase errors. In this case, the decoding graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) is identical to the physical lattice, that is, \( \mathcal{V} = \Sigma_0 \) and \( \mathcal{E} = \Sigma_1 \). By construction, a phase-flip error on any edge \( e \in \mathcal{G} \) creates a pair of syndromes at the two end-points of \( e \). More generally, given a subset of edges \( E \subseteq \mathcal{E} \), the syndrome of the Pauli error \( Z(E) \) coincides with the set \( \partial E \).

Let us now consider bit-flip errors. In this case the decoding graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) loosely coincides with the dual of the physical lattice. More precisely, let \( T \subseteq \Sigma_1 \) be the set of all qubits lying on the boundary of the lattice (either the boundary of some defect or the external boundary). We choose \( \mathcal{V} = \Sigma_2 \cup T \) and \( \mathcal{E} = \Sigma_1 \). A bit-flip error on any edge \( e \in \Sigma_1 \setminus T \) creates a pair of syndromes at the two plaquettes \( f, f' \) adjacent to \( e \). In the decoding graph \( f, f' \) are the two end-points of \( e \). A bit-flip error on a boundary edge \( e \in T \), however, creates a syndrome only at one plaquette \( f \) adjacent to \( e \). On the decoding graph \( e \) is a “hanging edge” connecting vertex \( f \) with a degree-1 vertex labeled by \( e \) itself. More generally, given a subset of edges \( E \subseteq \mathcal{E} \), the syndrome of a Pauli error \( X(E) \) is \( (\partial E) \setminus T \). To deal with both types of errors on the same basis, we set \( T = \emptyset \) in the case of phase-flip errors.

Let \( \psi \in \mathcal{L} \) be a logical state and \( P \) be an unknown Pauli error. We shall deal with phase-flip and bit-flip errors independently, so without loss of generality we can assume that \( P = Z(E) \) or \( P = X(E) \) for some subset of edges \( E \subseteq \mathcal{E} \) in the corresponding decoding graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \). Following Ref. [10] we shall refer to the subset \( E \) as an error chain. Error correction is a two-stage process that consists of a syndrome readout followed by a recovery step. A syndrome readout takes as input the corrupted state \( P \psi \) and performs a non-destructive eigenvalue measurement of each stabilizer. It determines the syndrome \( S = \partial E \setminus T \). We shall first consider the case when the syndrome readout circuit contains no errors. Noisy syndrome readout is discussed in Section [VIII]. A recovery step is determined by a decoding algorithm that takes as input the measured syndrome \( S \subseteq \mathcal{V} \setminus T \) and re-
Here \( \Gamma^Z \) is a closed loop on the primal lattice \( \Sigma \) encircling the (left-most) defect and \( \Gamma^X \) is a path on the lattice dual to \( \Sigma \) connecting the defect with the external boundary (for the double-defect geometry \( \Gamma^X \) connects the two defects with each other). One can easily check that \( \overline{X}, \overline{Z} \) commute with all stabilizers and anti-commute with each other. The decoding graph will be equipped with a logical chain \( \Gamma \subseteq \mathcal{E} = \Sigma_1 \) such that \( \Gamma = \Gamma^X \) for phase-flip errors and \( \Gamma = \Gamma^Z \) for bit-flip errors. Error correction is successful iff \( P'P \) commutes with the logical operators \( \overline{X}, \overline{Z} \). Equivalently, \( R \oplus \mathcal{E} \) must have even overlap with \( \Gamma \). The decoding graphs used to generated the data shown on Fig. 5 for \( r = 2 \) and the corresponding logical chains \( \Gamma \) are shown on Fig. 7.

V. DECODING ALGORITHMS

Here we discuss algorithms for choosing a recovery chain. This material is mostly based on Ref. [30].

Let \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) be the decoding graph defined above. We assume that every edge \( e \in \mathcal{E} \) has some specified error rate \( 0 \leq p(e) \leq 1/2 \). We shall only consider noise models such that errors on different edges of \( \mathcal{G} \) are independent. An error chain \( E \subseteq \mathcal{E} \) then appears with a probability

\[
\pi(E) = \prod_{e \in E} p(e) \prod_{e \in \mathcal{E} \setminus E} (1 - p(e)).
\]

Introduce edge weights \( \phi(e) \geq 0 \) such that

\[
\exp\left[-\phi(e)\right] = \frac{p(e)}{1 - p(e)}.
\]

Then \( \pi(E) = c \cdot \exp\left[-\phi(E)\right] \), where

\[
\phi(E) = \sum_{e \in E} \phi(e),
\]

and \( c \) is a constant coefficient independent of \( E \). We shall refer to \( \phi(E) \) as a weight of \( E \). Following Refs. [10, 30] we shall choose a recovery chain \( R \) as the most likely error chain consistent with the observed syndrome \( S \). Equivalently, a candidate recovery chain \( R \) must obey \( (\partial R) \cap T = S \) and \( \phi(R) \leq \phi(R') \) for any chain \( R' \subseteq \mathcal{E} \) satisfying \( (\partial R') \cap T = S \). Thus decoding amounts to solving one of the following problems.

**Problem 1.** Given a graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) with non-negative edge weights \( \phi(e) \) and a subset of vertices \( S \subseteq \mathcal{V} \). Find a minimum weight chain \( E \subseteq \mathcal{E} \) satisfying \( \partial E = S \).

**Problem 2.** Given a graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) with non-negative edge weights \( \phi(e) \) and disjoint sets \( S, T \subseteq \mathcal{V} \). Find a minimum weight chain \( E \subseteq \mathcal{E} \) satisfying \( (\partial E) \cap T = S \).

In combinatorial optimization, Problem 1 is known under the name minimum weight \( T \)-join (in our case \( T = S \)), see Ref. [54]. It can be solved in time \( O(|\mathcal{V}|^3) \) for any graph \( \mathcal{G} \) and any real edge weights by a reduction to
the minimum weight perfect matching problem and solving the latter using Edmonds’s blossom algorithm \cite{edmonds1965} or its subsequent improvements, see Ref. \cite{kolmogorov2009}. Our implementation of the decoder utilized the Blossom V library due to Kolmogorov \cite{kolmogorov2009}.

Let us now show that Problem 2 is equivalent to the maximum weight matching problem. Indeed, for any pair of vertices $u, v \in V \setminus T$ let $D_\phi(u, v)$ be the weighted distance between $u$ and $v$, that is, the minimum weight of a path in $G$ that connects $u$ and $v$. Likewise, let $D_\phi(u, T)$ be the weighted distance between $u$ and the subset $T$. One can easily check that any minimum weight path that connect pairs of vertices in $S$ and, possibly, minimum weight paths that connect a vertex in $S$ with a vertex in $T$. Hence Problem 2 is equivalent to finding a (non-perfect) matching $M$ of vertices in the complete graph $K_{|S|}$ with edge weights $D_\phi(u, v)$ that minimizes the objective function

$$f(M) = \sum_{\text{matched}} D_\phi(u, v) + \sum_{\text{unmatched}} D_\phi(u, T).$$

Here the first sum runs over all pairs of vertices $u, v \in S$ which are matched to each other in $M$, while the second sum runs over unmatched vertices $u \in S$. To minimize $f(M)$ define modified edge weights

$$\eta(u, v) = D_\phi(u, T) + D_\phi(v, T) - D_\phi(u, v).$$

The objective function $f(M)$ can now be rewritten as

$$f(M) = c - \sum_{(u, v) \in M} \eta(u, v),$$

where $c = \sum_{u \in S} D_\phi(u, T)$ is a constant that does not depend on the choice of $M$. Thus Problem 2 is reduced to finding a maximum weight matching $M$ in the complete graph $K_{|S|}$ with edge weights $\eta(u, v)$. A maximum weight matching can be found in time $O(|S|^3)$ using a slightly different version of Edmonds’s blossom algorithm \cite{edmonds1965}. Our implementation of the decoder was based on the library LEMON \cite{lemon} which realizes a maximum weight matching algorithm due to Gabov \cite{gavov1973}. It is worth pointing out that the optimal matching $M$ can be always chosen such that pairs of vertices with $\eta(u, v) \leq 0$ are unmatched. Such negative-weight edges can be safely removed from the graph before calling the matching algorithm. Although the reduction outlined above is elementary, to the best of our knowledge it has not been used before in the context of error correction.

VI. CORRECTABILITY

In this section we define correctable and uncorrectable error chains and prove some technical lemmas needed for the analysis of the splitting method.

Let $C_{\min}(S)$ and $C_{\min}(S, T)$ be the sets of minimum weight chains $E$ satisfying $\partial E = S$ and $(\partial E) \cap T = S$ respectively, see Problems 1, 2. Clearly, $C_{\min}(S) = C_{\min}(S, \emptyset)$. Below we consider minimum weight decoders (MWD), that is, algorithms choosing a recovery chain $R$ from the set $C_{\min}(S, T)$ according to some specified rule. Define the parity $\epsilon$ of an error chain $E \subseteq E$ as

$$\epsilon(E) = |E \cap \Gamma| \pmod{2},$$

where $\Gamma \subseteq E$ is the logical chain on the decoding graph. We will say that $E$ is odd (even) if $\epsilon(E) = 1 \,(= 0)$. Recall that error correction is successful if the chosen recovery chain has the same syndrome and the same parity as the actual error chain, see Section V.

Let $E \subseteq E$ be some fixed error chain. Deciding whether a MWD corrects $E$ is straightforward when all chains in $C_{\min}(S, T)$ have the same parity. Indeed, in this case the correctability condition $\epsilon(R) = \epsilon(E)$ is satisfied or not simultaneously for all $R \in C_{\min}(S, T)$.

In general, however, the set $C_{\min}(S, T)$ may contain both even and odd chains, see Fig. 8 for a simple example. In this case we will say that $S$ is a degenerate syndrome. Deciding whether an error chain with a degenerate syndrome is corrected by a given MWD requires a detailed knowledge of the rule used for choosing a recovery chain. As was pointed out by Stace and Barrett \cite{stace2019}, some decoding rules can reduce the logical error probability by breaking ties in favor of chains having the largest entropy, see Fig. 8. This demonstrates that the logical error probability $P_L$ is not well-defined unless the decoder’s behavior for all degenerate syndromes is specified.

![FIG. 8: Example of a degenerate syndrome. One can easily check that any error chain $E$ satisfying $\partial E = \{1, 2, 3, 4\}$ has length at least 12. There is a unique even length-12 chain composed of paths $\{1, 2\}$ and $\{3, 4\}$. There are $\binom{6}{3}^2 = 400$ odd length-12 chains composed of paths $\{1, 4\}$ and $\{2, 3\}$. If the actual error chain has minimum length (which is true in the limit of small error rates), a decoding rule that favors the pairing $\{1, 4\}, \{2, 3\}$ is 400 times more likely to correct the error.

To address this difficulty we opted to work with two different notions of correctability — decoder-specific and decoder-independent. For the purposes of numerical simulations, an error chain $E$ is called correctable if a recovery chain $R$ chosen by our implementation of the decoder satisfies $\epsilon(R) = \epsilon(E)$. Otherwise, $E$ is called uncorrectable. Accordingly, the logical error probability $P_L$
computed in our simulations applies only to the one specific MWD implemented as described in Section V. From a practical perspective, this is a natural approach, since an experimental realization of error correction must be based upon some specific decoder. Furthermore, we expect that $P_L$ does not vary too much for different MWDs because degenerate syndromes are relatively rare. The decoder-specific notion of correctability was implicitly used in most of the previous work on the subject.

From the theoretical perspective, it is more natural to work with a decoder-independent notion of correctability. The definition given below is used in the rest of this section where we make first steps towards a rigorous justification of the splitting method.

**Definition 1.** Let $E \subseteq \mathcal{E}$ be an error chain with a syndrome $S = (\partial E) \setminus T$. We say that $E$ is correctable iff $\epsilon(E) = \epsilon(R)$ for all $R \in C_{\text{min}}(S, T)$. Otherwise, $E$ is called uncorrectable.

According to this definition, an error chain is correctable iff any MWD corrects it. Note that this requirement is much stronger than the decoder-specific correctability discussed above. Below we prove that the strong version of correctability can be tested efficiently in certain special cases. The following lemmas are applicable to the decoding graphs used in our simulations for noiseless syndrome readout, see Fig. 7. Let us first assume that $T = \emptyset$.

**Lemma 1.** Suppose the decoding graph $\mathcal{G}$ is planar with the maximum vertex degree $O(1)$. Suppose that the logical chain $\Gamma$ is a path on the dual graph. Then correctability can be tested in time $O(|\mathcal{V}|^3)$.

**Proof.** Let $E$ be any error chain and $S = \partial E$ be its syndrome. Choose any recovery chain $R_0 \in C_{\text{min}}(S)$. It can be constructed in time $O(|\mathcal{V}|^3)$, see Section V. If $R_0 \oplus E$ has odd parity then $E$ is uncorrectable and we are done. Below we assume that $R_0 \oplus E$ has even parity. Then $E$ is correctable iff all chains in $C_{\text{min}}(S)$ have the same parity.

A chain $C \subseteq \mathcal{E}$ is called a cycle iff $\partial C = 0$. Let $C_{\text{odd}}$ be the set of all cycles with odd parity. Define a quantity

$$\delta = \min_{C \in C_{\text{odd}}} \phi(R_0 \oplus C) - \phi(R_0).$$

The minimality of $R_0$ implies that $\delta \geq 0$. We claim that $E$ is uncorrectable iff $\delta = 0$. Indeed, suppose $E$ is uncorrectable. Then there must exist a chain $R \in C_{\text{min}}(S)$ such that $R_0$ and $R$ have different parity. Therefore $C = R_0 \oplus R$ is an odd cycle and $\phi(R_0 \oplus C) = \phi(R) = \phi(R_0)$, that is, $\delta = 0$. Conversely, assume that $\delta = 0$. Then $\phi(R_0 \oplus C) = \phi(R_0)$ for some odd cycle $C$ and thus $R = R_0 \oplus C \in C_{\text{min}}(S)$ has parity different from $R_0$. Therefore $E$ is uncorrectable.

Thus, it suffices to show that the quantity $\delta$ can be computed in time $O(|\mathcal{V}|^3)$. Define new weights

$$w(e) = \begin{cases} \phi(e) & \text{if } e \notin R_0, \\ -\phi(e) & \text{if } e \in R_0. \end{cases}$$

Then $\delta$ coincides with the minimum $w$-weight of an odd cycle, $\delta = \min_{C \in C_{\text{odd}}} w(C)$. Let us show that $\delta$ can be expressed as the ground state energy of the Ising model defined on a genus-1 graph (i.e. a graph embeddable into a torus). This ground state energy can be computed in time $O(|\mathcal{V}|^3)$ using algorithms of Refs. [41–44]. Indeed, since $\mathcal{G}$ is planar, it has a well-defined set of faces $\mathcal{V}^*$ such that every edge $e \in \mathcal{E}$ has exactly two adjacent faces $f, g \in \mathcal{V}^*$. We will write $e = (f, g)$. For each face $f \in \mathcal{V}^*$ introduce Ising spin $\sigma_f \in \{-1, 1\}$. Given a spin configuration $\sigma = \{\sigma_f\}_{f \in \mathcal{V}^*}$, define a chain $C(\sigma) \subseteq \mathcal{E}$ such that $C(\sigma)$ includes all edges $e = (f, g)$ with $\sigma_f \sigma_g = -1$. The standard relationship between cycles in a planar graph and cuts in the dual graph implies that $C(\sigma)$ is a cycle for any choice of $\sigma$ and that any cycle $C \subseteq \mathcal{E}$ has form $C(\sigma)$ for some spin configuration $\sigma$.

By assumption, $\Gamma$ is a path on the dual graph connecting some pair of faces $f^\prime, f^\prime\prime \in \mathcal{V}^*$. Simple algebra shows that a cycle $C(\sigma)$ has odd parity iff $\sigma_{f^\prime} \sigma_{f^\prime\prime} = -1$. We conclude that $\delta = \min_{\sigma} H(\sigma), \quad H(\sigma) = \text{Ising-like Hamiltonian on the dual graph defined as}$

$$H(\sigma) = J(1 + \sigma_{f^\prime} \sigma_{f^\prime\prime}) + \sum_{e = (f, g) \in \mathcal{E}} w(e)(1 - \sigma_f \sigma_g)/2$$

Here $J \gg 1$ is chosen large enough to guarantee that $\sigma_{f^\prime} \sigma_{f^\prime\prime} = -1$ for any ground state $\sigma$. To make the interaction $\sigma_{f^\prime} \sigma_{f^\prime\prime}$ compatible with the graph structure, the edge $(f^\prime, f^\prime\prime)$ must be added to the dual graph $\mathcal{G}^\prime$. Although the resulting graph is not planar, it is embeddable into a torus. Indeed, one can first embed the primal graph $\mathcal{G}$ into a sphere, create a pair of holes in the faces $f^\prime, f^\prime\prime$, connect the two holes by a tube, and finally draw the edge $(f^\prime, f^\prime\prime)$ on the tube’s surface. The algorithms of Refs. [41–44] enable exact computation of the partition function $Z(\beta) = \sum_\sigma \exp[-\beta H(\sigma)]$ in time $O(|\mathcal{V}|^3)$ for any value of the inverse temperature $\beta$ assuming that the graph of spin-spin interactions has constant genus (see, for instance, Theorem 1 of Ref. [44]). The two cases $\delta = 0$ and $\delta > 0$ correspond to $\lim_{\beta \to \infty} Z(\beta) = 1$ or $\lim_{\beta \to -\infty} Z(\beta) = 0$ which can be distinguished by computing $Z(\beta)$ for large enough $\beta$.

Suppose now that $T$ is non-empty. Recall that a subset of edges $C \subseteq \mathcal{E}$ is called a cut iff one can partition vertices of the graph into two disjoint sets, $\mathcal{V} = \mathcal{V}_0 \cup \mathcal{V}_1$, such that $C$ coincides with the set of edges connecting $\mathcal{V}_0$ and $\mathcal{V}_1$.

**Lemma 2.** Suppose the decoding graph $\mathcal{G}$ is arbitrary and the logical chain $\Gamma$ is a cut of $\mathcal{G}$. Then correctability can be tested in time $O(|\mathcal{V}|^3)$.

**Proof.** Let $E$ be any error chain and $S = (\partial E) \setminus T$ be its syndrome. Choose any recovery chain $R_0 \in C_{\text{min}}(S, T)$. It can be constructed in time $O(|\mathcal{V}|^3)$, see Section V. If $R_0 \oplus E$ has odd parity then $E$ is uncorrectable and we are done. Below we assume that $R_0 \oplus E$ has even parity. Then $E$ is correctable iff all chains in $C_{\text{min}}(S, T)$ have the same parity.
Without loss of generality, no edge \( e \) having both endpoints in \( T \) belongs to \( \Gamma \). Indeed, such an edge \( e \) does not participate in the constraint \( (\partial R) \setminus T = S \). If \( e \) has positive weight \( \phi(e) \), no chain in \( C_{\text{min}}(S,T) \) contains \( e \) and we can safely remove \( e \) from \( \Gamma \) without changing the parity of any \( R \in C_{\text{min}}(S,T) \). If \( e \) has zero weight, \( C_{\text{min}}(S,T) \) contains both odd and even chains and the error is uncorrectable.

By assumption, \( V = V_0 \cup V_1 \), \( V_0 \cap V_1 = \emptyset \), and \( \Gamma \) is the set of edges connecting \( V_0 \) and \( V_1 \). Below we shall often use the obvious fact that for any chain \( R \subseteq \mathcal{E} \) the overlap \( [R \cap \Gamma] \) is even (odd) if \( (\partial R) \cap V_0 \) is even (odd).

Consider two cases.

Case 1: \( T \cap V_0 = \emptyset \). Then all chains \( E \) satisfying \( (\partial E) \setminus T = S \) are either even or odd depending on whether \( |S \cap V_0| \) is even or odd. Hence \( E \) is correctable.

Case 2: \( T \cap V_0 \neq \emptyset \). Define a new graph \( \mathcal{G} = (V, \mathcal{E}) \) obtained from \( \mathcal{G} \) by collapsing all vertices of \( T \cap V_0 \) into a single vertex \( t_0 \). More precisely, we replace every edge \( (u, v) \) with \( u \in T \cap V_0, v \notin T \) by an edge \( (t_0, v) \) without changing its weight. We can regard \( \Gamma \) as a subset of edges of \( \mathcal{G} \). Define \( S' = S, S'' = S \cup t_0 \), and \( T = T \cap V_1 \). Let \( w'_{\text{min}} \) and \( w''_{\text{min}} \) be minimum weights of chains \( R', R'' \in \mathcal{E} \) satisfying \( (\partial R') \setminus T = S' \) and \( (\partial R'') \setminus T = S'' \) respectively. Note that \( w'_{\text{min}} \) and \( w''_{\text{min}} \) can be computed in time \( O(|V|^3) \), see Section \( \text{III} \). Furthermore, any chain \( R' \) as above has even (odd) parity iff \( |S \cap V_0| \) is even (odd). On the other hand, any chain \( R'' \) as above has even (odd) parity iff \( S \cap V_0 \) is odd (even). Hence \( w'_{\text{min}} = w''_{\text{min}} \) if \( C_{\text{min}}(S,T) \) contains two chains with a different parity. We conclude that \( E \) is uncorrectable if \( w'_{\text{min}} = w''_{\text{min}} \).

Our implementation of the splitting method involves a Metropolis-type subroutine for sampling uncorrectable error chains from the chosen probability distribution. The following two lemmas are needed to prove that the corresponding Markov process is ergodic. Let us first assume that \( T = \emptyset \).

Lemma 3. Suppose the decoding graph \( \mathcal{G} \) is a square lattice with one smooth defect and \( \Gamma \) is a path on the dual lattice that connects the defect to the external boundary, see Fig. \( \text{II} \). Let \( E, E' \) be any uncorrectable chains. Then one can transform \( E \) to \( E' \) by adding and removing single edges such that all intermediate chains are uncorrectable.

Proof. Let \( \Omega \) be the set of all edges lying on the boundary of the defect. Since \( \Omega \) has trivial syndrome, \( \partial \Omega = \emptyset \), and odd parity, \( \Omega \) is uncorrectable by Definition 1. It suffices to prove the lemma for the special case \( E' = \Omega \). We will need the following simple observation.

Proposition 1. Let \( S \subseteq V \) be a subset of vertices and \( R \in C_{\text{min}}(S) \) be any chain. Choose any edge \( e = (u,v) \in R \) and define \( S' = S \cup \{u,v\} \). Then \( R \setminus e \in C_{\text{min}}(S') \).

Proof. Obviously, \( \partial (R \setminus e) = \partial (R \oplus e) = \partial R \oplus e = S' \). Thus, \( \partial (R \setminus e) = S' \) and \( \phi(R \setminus e) = \phi(R) - \phi(e) \). Suppose \( R \setminus e \) is not in \( C_{\text{min}}(S') \). Then there exists \( R' \) such that \( \partial (R') = S' \) and \( \phi(R') < \phi(R) - \phi(e) \). Define \( R'' = R' \oplus e \). Then \( \partial R'' = S \) and \( \phi(R'') \leq \phi(R') + \phi(e) < \phi(R) \) which contradicts the minimality of \( R \). Thus \( R \setminus e \in C_{\text{min}}(S') \).

Let \( S = \partial E \) be the syndrome of \( E \). Since \( E \) is uncorrectable, there must exist a recovery chain \( R \in C_{\text{min}}(S) \) such that \( E \oplus R = L_1 \cup \ldots \cup L_m \) is a disjoint union of loops and the number of odd loops among \( L_1, \ldots, L_m \) is odd. Choose any edge \( e \in R \) and define \( E_1 = E \oplus e, R_1 = R \oplus e \). The proposition above implies that \( R_1 \in C_{\text{min}}(\partial E_1) \), that is, \( R_1 \) is a minimum weight recovery chain for \( E_1 \). Furthermore, since \( E \oplus R = E_1 \oplus R_1 \), the new error chain \( E_1 \) is uncorrectable. By repeating this argument one can construct a sequence of uncorrectable error chains \( E_0 = E, E_1, \ldots, E_p \) such that \( E_{i+1} \) is obtained from \( E_i \) by adding (modulo two) any edge from the recovery chain \( R_i \) corresponding to \( E_i \). This process stops as soon as the recovery chain corresponding to \( E_p \) is empty. At this step \( E_p = L_1 \cup \ldots \cup L_m \) is a disjoint union of loops. We shall deal with the loops \( L_m \) one by one such that each even loop is transformed into an empty chain (contracted) while each odd loop is transformed into \( \Omega \). These transformations can be implemented such that at every step a loop is modified by adding (modulo two) a boundary of some plaquette which requires adding or removing at most three edges. At each step the syndrome consists of at most two vertices and the recovery chain consists either of a single edge or a pair of adjacent edges lying on the boundary of some plaquette. Since the number of odd loops is odd, the final error chain coincides with \( \Omega \).

Suppose now that \( T \neq \emptyset \).

Lemma 4. Suppose the decoding graph \( \mathcal{G} \) is a square lattice with two rough defects and \( \Gamma \) is a loop on the dual lattice encircling one of the defects, see Fig. \( \text{IV} \). Let \( E, E' \) be any uncorrectable chains. Then one can transform \( E \) to \( E' \) by adding and removing single edges such that all intermediate chains are uncorrectable.

Proof. As explained in the proof of Lemma 2, one can collapse all vertices \( u \in T \) on the boundary of each defect into a single vertex. Thus we can assume that \( T = \{t', t''\} \). Let \( \Omega \) be a path connecting \( t' \) and \( t'' \). Clearly, \( \Omega \) is uncorrectable. It suffices to prove the lemma for the case \( E' = \Omega \). Repeating the same steps as in the proof of Lemma 3 one can transform \( E \) to a disjoint union of loops and paths connecting \( t' \) and \( t'' \), such that the number of paths is odd. Applying a sequence of plaquette transformations as in the proof of Lemma 3 one can contract each loop and transform each path into \( \Omega \).

VII. IMPLEMENTATION OF THE SPLITTING METHOD

Here we specialize the splitting method described in Section \( \text{III} \) to the minimum weight decoding problem. Let \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) be the decoding graph corresponding to
logical loop-like or path-like errors, see Section [V]. For simplicity we shall first discuss the case when all edges have the same error probability, \( \pi(E) = p^{|E|}(1-p)^{n-|E|} \), where \( n = |\mathcal{E}| \) is the total number of edges.

We define an event as an arbitrary error chain \( E \subseteq \mathcal{E} \). Accordingly, \( \Omega \) coincides with the set of all subsets of \( \mathcal{E} \). The subset of failure events \( \mathcal{F} \subseteq \Omega \) consists of all uncorrectable chains such that \( \pi(F) \) is the probability of a logical loop-like or path-like error. We shall first focus on the decoder-specific definition of correctability, see Section [V]. Testing a membership \( E \in \mathcal{F} \) thus requires solving Problem 1 or 2.

We choose the family of distributions \( \pi_1, \ldots, \pi_t \) as
\[
\pi_j(E) = p_j^{|E|}(1-p_j)^{n-|E|}, \quad j = 1, \ldots, t
\]
for a monotonic sequence of error rates \( p_1, \ldots, p_t \) with \( p_1 = p \). The following heuristic choice of the splitting sequence was found to provide a reasonable tradeoff between the statistical error and the number of splitting steps:
\[
p_{j+1} = p_j + 2^{1/\sqrt{\tau_j}}, \quad w_j = \max(d/2, p_j n).
\]
Here \( d \) is the code distance and the two signs correspond to upward and downward splitting. To motivate this choice we note that the quantity \( w_j \) provides a rough estimate of the average number of edges in a random chain \( E \in \mathcal{F} \) drawn from the distribution \( \pi_j(E|F) \). Since errors on different edges are independent and \( p_j \ll 1 \), one should expect that the random variable \( |E| \) is concentrated near its mean with the standard deviation \( O(\sqrt{\tau_j}) \). Therefore one can use a bound
\[
\left( \frac{p_{j+1}}{p_j} \right)^{-O(\sqrt{\tau_j})} \leq \frac{\pi_{j+1}(E|F)}{\pi_j(E|F)} \leq \left( \frac{p_{j+1}}{p_j} \right)^{O(\sqrt{\tau_j})}
\]
for all ‘typical’ error chains \( E \). Here, for concreteness, we consider upward splitting, that is, \( p_j < p_{j+1} \). Then Eq. (17) implies
\[
c^{-1}\pi_j(E|F) \leq \pi_{j+1}(E|F) \leq c\pi_j(E|F)
\]
for some constant \( c = O(1) \). Thus \( \|\pi_j(\cdot|F) - \pi_{j+1}(\cdot|F)\|_1 \leq 1 - 1/c \) and the ratio \( \pi_{j+1}(F)/\pi_j(F) \) is estimated with an error \( \sigma_j \leq \sqrt{2cN^{-1}} = O(N^{-1/2}) \), see Eq. (12).

In order to sample a chain \( E \in \mathcal{F} \) from the conditional distribution \( \pi(E|F) = \pi(E)/\pi(F) \) we used a Metropolis-type subroutine. A single Metropolis step takes as input a chain \( E \in \mathcal{F} \) and outputs a new chain \( E' \in \mathcal{F} \) which differs from \( E \) on at most one edge as described below.

1. Select an edge \( e \in \mathcal{E} \) at random from the uniform distribution. Set \( E' = E \oplus e \).
2. Compute \( q = \min[1, \pi(E')/\pi(E)] \) and generate a random bit \( b = 0, 1 \) such that \( \Pr(b = 1) = q \).
3. If \( b = 0 \) then stop and output \( E \).
4. If \( b = 1 \) and \( E' \in \mathcal{F} \) then output \( E' \). Otherwise output \( E \).

For any chains \( E, E' \in \mathcal{F} \) let \( P(E, E') \) be the probability that the Metropolis step outputs \( E' \) if called on the input \( E \). One can easily check that \( P \) obeys a detailed balance condition
\[
\pi(E)P(E, E') = \pi(E')P(E', E) \quad \text{for all } E, E' \in \mathcal{F}.
\]
Thus the Metropolis step defines a reversible Markov process \( \mathcal{M} \) such that states of \( \mathcal{M} \) are uncorrectable error chains, \( P(E, E') \) is the transition probability from \( E \) to \( E' \), and \( \pi(E|F) \) is a steady distribution of \( \mathcal{M} \). A similar Markov process \( \mathcal{M}_j \) is constructed for each distribution \( \pi_j \) in the splitting sequence.

The full Metropolis subroutines involves \( M \gg 1 \) Metropolis steps starting from some fixed initial uncorrectable chain \( E_0 \). The latter was chosen as a loop encircling a defect (for loop-like errors) or a path connecting the two defects (for path-like errors). The sequence of uncorrectable error chains \( E_0, E_1, \ldots, E_M \) generated by the Metropolis subroutine was used to estimate the expectations values in Eq. (8). In order for Eq. (8) to hold, the number of Metropolis steps must satisfy \( M \gg N\tau_j \), where \( \tau_j \) is the mixing time of the Markov process \( \mathcal{M}_j \). Since in practice the mixing time is unknown, the number of steps \( M \) was chosen by checking two heuristic conditions: (i) statistical fluctuations of the righthand side of Eq. (8) are smaller than the desired precision, (ii) doubling \( M \) does not change the righthand side of Eq. (8) by more than the desired precision. Our goal was to compute the logical error probability \( \pi(F) \) with a relative error about 50%. Note that \( \pi(F) \) changes by almost 20 orders of magnitude in our simulations. A relative error 50% is thus good enough for all practical purposes. Accordingly, we aimed at estimating the expectation values in Eq. (8) with a relative error around 0.5/t, where \( t \) is the number of splitting steps. The required number of Metropolis flips (non-trivial steps) varied in the range \( 10^5 \) to \( 10^7 \) depending on the geometry, lattice dimensions, and the error rate.

Note that some Metropolis steps may require testing a membership \( E' \in \mathcal{F} \), which in turn requires solution of Problem 1 or Problem 2, see Section [V]. As the full Metropolis subroutine may involve millions of steps, a natural question is whether the solution of Problems 1,2 obtained at some Metropolis step \( j \) can be ‘recycled’ and used at the next step \( j + 1 \). Suppose the corresponding error chains differ on some edge \( e \), \( E_{j+1} = E_j \oplus e \). To perform the standard reduction from Problems 1,2 to the minimum (maximum) weight matching problem one has to construct a family of minimum weight paths on the decoding graph connecting any pair of syndrome vertices, see Section [V]. Our implementation of the Metropolis subroutine recycles minimum weight paths found for the syndrome \( S_j = \partial E_j \setminus T \) and uses them to construct minimum weight paths for the syndrome \( S_{j+1} = \partial E_{j+1} \setminus T \). This yields a significant speedup since the syndromes \( S_j \) and
distribution. Therefore, in these special cases the conditional
Metropolis step can be implemented in time
particular, for the decoding graphs shown on Fig. 7 the
independent notion of correctability, see Section V. In
the minimum weight decoding problem with a decoder-
sponding decoding graphs are shown on Fig. 7.
In the case of path-like logical errors simulations were
performed for the double-defect geometry. The corre-
like errors does not depend on the number of defects, as
as separation between defects is sufficiently large.
In the case of loop-like logical errors simulations were
performed for the double-defect geometry. The corre-
sponding decoding graphs are shown on Fig. 7.

VIII. NOISY SYNDROME READOUT

Here we describe the construction of the decoding
graph and the implementation of the splitting method
in the case when the syndrome readout circuit itself may
introduce errors. The material of this section is mostly
based on Refs. [10, 12, 45].

We begin by defining the noise model and the syn-
drome readout circuit. Our set of elementary operations
includes CNOT gates, single-qubit measurements in the
X- or Z-basis, and preparation of single-qubit ancillary
states $|0\rangle$ or $|+\rangle$. The syndrome readout circuit consists of a sequence of rounds, where at each round any qubit
can participate in one elementary operation or remain
idle. Each elementary operation can fail with a probabil-
ity $p$ that we call an 
error rate. More precisely, our error
model, borrowed from [12], is defined as follows.

- A noisy X or Z measurement is the ideal measure-
ment in which the outcome is flipped with proba-
bility $p$.

- A noisy $|0\rangle$ or $|+\rangle$ ancilla preparation returns the
  correct state with probability $1 - p$ and the orthog-
  onal state $|1\rangle$ or $|-\rangle$ with probability $p$.

- A noisy CNOT gate is the ideal CNOT gate fol-
  lowed by one of 16 two-qubit Pauli operators $P$.
  We apply $P = I$ with probability $1 - p$ and each
  individual $P \neq I$ with probability $p/15$.

- If a qubit remains idle during some round, it is
  acted upon by X, Y or Z error with probability $p/3$ each (“memory error”).

Following Refs [10, 12], we measure eigenvalues of
site and plaquette stabilizers using the quantum circuit
shown in Fig. 9. Measuring a single stabilizer requires one
ancillary qubit and six rounds. In the case of truncated
stabilizers located near the boundary of defects, some
CNOT gates in the circuit of Fig. 9 are skipped and the
corresponding ancillary qubits remain idle. The ancillary
qubits are located at the centers of plaquettes and at sites
of the physical lattice. The syndrome readout is repeated
periodically in time until enough syndrome data is col-
lected to enable reliable error correction, see below. We
assume that the rounds are scheduled such that a new set
of syndrome data from every stabilizer arrives at each in-
teger time step $t$ (accordingly, each round takes 1/6 units
of time). For simplicity, we assume that defects are not
added, changed, or annihilated during the collection of
syndrome data; that is, we are only examining storage of
information in time.

The error correction protocol is tailored to the chosen
noise model and the syndrome readout circuit [12]. As
before, the key ingredient in the protocol is a decoding
graph $G = (V, E)$ and we again need two independent
decoding graphs for dealing with phase-flip and bit-flip
events separately. For concreteness, below we focus on
bit-flip errors. The vertices of $G$ can be partitioned into
two disjoint subsets, $V = V \cup T$. Each vertex in $V \subseteq V$ is
a pair $u = (p, t)$ that represents the space-time location
of a syndrome bit measured at time step $t$ at a plaquette
$p$. Let $S \subseteq V$ be the set of all vertices $(p, t)$ such that
the syndrome bits measured at the plaquette $p$ at time steps
$t$ and $t + 1$ are different. We shall refer to $S$ as a relative
syndrome. Clearly, $S = \emptyset$ in the absence of errors. Sup-
pose now that the syndrome readout circuit contains a
single error event, that is, any non-identity Pauli operator
applied as an error in any single elementary operation
or idle step. It can be shown that a relative syndrome
caused by any single error event in the circuit consists of
at most two vertices, see [12]. We connect a pair of
vertices $u, v \in V$ by an edge iff the relative syndrome
$S = \{u, v\}$ can be created by a single error event.
error events located near the boundary of a defect create a relative syndrome at a single vertex \( u \). Such an error event is represented on the decoding graph by a “hanging edge” attached to \( u \). Each vertex \( u \in V \) is attached to at most one hanging edge: the other endpoint of this hanging edge has degree one and is an element of \( T \subseteq V \). The set \( T \) solely consists of the hanging edge endpoints that are not in \( V \).

The decoding graph corresponding to the surface code lattice of Fig. 1 for bit-flip errors is shown on Fig. 10. To avoid clutter, we represent a hanging edge attached to some vertex \( u \in V \) by a solid circle centered at \( u \). A typical vertex of the decoding graph has 12 incident edges, see Fig. 11 for detail. Generally, memory errors are represented by edges oriented along the \( x \) or \( y \) axes, while measurement and initialization errors lead to edges oriented along the \( t \) axis. Any type of edge can be observed due to the two-qubit errors (which occur after each of the rounds of CNOT gates), and many of the diagonal edges can only be observed in this way. Two edges of the same orientation may be created by different errors depending on their proximity to defects. The decoding graph describing phase-flip errors is constructed in a similar fashion (hanging edges on the phase-flip decoding graph are not needed for the defects examined here).

By construction, every edge \( e \) of the decoding graph represents some set of error events \( \Omega_e \) in the syndrome readout circuit (those that create relative syndromes at the endpoints of \( e \)). The sets \( \Omega_e \) corresponding to different edges are disjoint. Define a prior \( p(e) \) as the probability of observing an odd number of error events in \( \Omega_e \) upon execution of the circuit (since errors add up modulo two, only the parity of the number of errors matters). As argued in [12], knowledge of the priors significantly improves decoding success probability. Edges of the decoding graph with large prior probabilities can be regarded as more noisy and should be preferred over less-noisy edges when choosing a recovery chain. The time-like edges and the diagonal edges are the most and the least noisy respectively. We estimated the priors \( p(e) \) by summing up the probabilities of all error events in the set \( \Omega_e \). The priors are represented by a color scale on Fig. 10.

Any combination of error events in the circuit can be represented by an error chain \( E \subseteq \mathcal{E} \) in the decoding graph such that \( e \in E \) iff the set \( \Omega_e \) contains an odd number of the error events that occurred when the circuit was run. Given an edge \( e \in \mathcal{E} \), let \( \Pi(e) \subseteq \Sigma_1 \) be the ‘projection’ of \( e \) onto the 2D surface code lattice. More precisely, if \( e = (u, u') \) for some vertices \( u = (p, t) \) and \( u' = (p', t') \) then \( \Pi(e) \) consists of the edges making up a minimum weight path between \( p \) and \( p' \) for \( p \neq p' \) and \( \Pi(e) = \emptyset \) otherwise. Note that when \( p \neq p' \), \( \Pi(e) \) contains one edge except in some of the cases where \( e \) is a diagonal edge. Given any chain \( E = \{e_1, e_2, \ldots, e_m\} \subseteq \mathcal{E} \) on the decoding graph, the corresponding accumulated 2D error chain on the surface code qubits is

\[
\Pi(E) = \Pi(e_1) \oplus \Pi(e_2) \oplus \ldots \oplus \Pi(e_m) \subseteq \Sigma_1.
\]

Our goal is to use the syndrome information to correct the error chain \( \Pi(E) \). By construction, \( (\partial E) \cap T = S \), where \( S \subseteq V \) is the relative syndrome and \( T \subseteq V \) is defined above. Thus finding the most likely error chain consistent with a given relative syndrome \( S \) is equivalent to solving Problem 1 or 2, see Section [V]. Let \( R \in C_{\text{min}}(S, T) \) be a minimum weight recovery chain constructed by the decoder. The recovery operator corresponding to \( R \) is determined by the 2D projection \( \Pi(R) \). An error chain \( E \) is called correctable iff \( \Pi(R) \oplus \Pi(E) \) has even overlap with the relevant logical chain \( \Gamma \), see Section [VI]. Equivalently, \( E \) is correctable iff \( R \oplus E \) has even overlap with a 3D logical chain \( \Gamma \subseteq \mathcal{E} \) that includes all edges \( e \in \mathcal{E} \) such that \( \Pi(e) \cap \Gamma \) is not empty. In the case of bit-flip errors, \( \Gamma \) is the set of all hanging edges located on the boundary of the left defect tube, see Fig. 10. For phase-flip errors, \( \Gamma \) can be visualized as a ‘membrane’ connecting the defect tube to the external spatial boundary of the lattice (not shown). Let \( \Omega \) be the set of all error chains and \( \mathcal{F} \subseteq \Omega \) be the set of uncorrectable chains.

Apart from the different definition and interpretation of the decoding graph, the implementation of the splitting method is exactly the same as described in Section [VII]. Our simulations were performed for a phenomenological noise model where errors on different edges of the decoding graph occur independently with probabilities \( p(e) \). In order to evaluate the quantity \( P_L(p) \) for a given probability \( p \), we used a family of distributions \( \pi_1, \ldots, \pi_t \) defined as

\[
\pi_j(E) = \prod_{e \in E} p_j(e) \prod_{e \in \mathcal{E} \setminus E} (1 - p_j(e)),
\]

where \( p_j(e) \) are the priors computed for a monotone decreasing sequence of error rates \( p_1, \ldots, p_t \) such that \( p_t = p \). The sequence \( p_1, \ldots, p_t \) is defined by the heuristic rule Eq. 17, where \( w_2 = \sum e \in E p_j(e) \). Simulations were performed only for defects with linear size \( r = 2, 3, 4 \), partly due to the growing running time of the Metropolis subroutine and partly due to computer memory limitations (for the double defect geometry with \( r = 4 \) the lookup table of minimum weight paths on the decoding graph takes about 4GB of RAM). For path-like logical errors the decoding graph shown in Fig. 10 is a 3D lattice with a pair of vertical defect tubes. The decoding graph corresponding to loop-like logical errors is similar to Fig. 11 but there is only one vertical defect tube. By combining the splitting method and the Monte Carlo data we were able to compute parameters of the fitting formula Eqs. 24 which we expect to be valid for larger code distances. The decay rate \( \alpha(p) \) in the exponential scaling \( P_L \sim \exp[-\alpha(p)r] \) is shown in Fig. 3.

IX. CONCLUSION AND OPEN PROBLEMS

We proposed a new algorithm for estimating the logical error probability of the surface code in the regime of large
code distances and moderately small error rates. Numerical results are presented for two commonly studied error models corresponding to noiseless and noisy syndrome extraction. Our results demonstrate that the asymptotic formulas for the logical error probability $P_L(p)$ valid in the limit $p \to 0$ tend to underestimate $P_L(p)$ for finite error rates. A more accurate fitting formula for $P_L(p)$ is proposed.

Our work certainly leaves many important questions unanswered. First, one may ask whether our simulation techniques can be extended to non-trivial logical gates, such as the CNOT gate, or more complicated logical circuits such as the topological state distillation \cite{10}. Each of these circuits can be visualized as a network of defect tubes embedded into a 3D space-time \cite{11}. We anticipate that the logical error probability $P_L$ associated with a large network of tubes can be estimated by decomposing the network into small tiles that consist of single isolated tube segments or parallel pairs of such segments. The techniques presented in this paper are applicable to each individual tile. Therefore one can get a rough estimate of $P_L$ by summing up logical error probabilities associated with each tile.

From the theoretical perspective, it is desirable to derive rigorous bounds on the running time and the approximation error of the algorithm. This, in turn, requires upper bounds on the mixing time of the Metropolis subroutine described in Section VII. We conjecture that the mixing time scales as $p^{-\Omega(d)}$ for a general distance-$d$ surface code with multiple defects in the limit $p \to 0$.

The intuition behind this conjecture is that minimum-weight uncorrectable error chains that are localized on the boundary of different defects cannot be connected by a sequence of local Metropolis steps without passing through intermediate high-weight uncorrectable error chains. However, if the lattice contains a single defect (in the case of loop-like errors) or a pair of defects (in the case of path-like errors), see Fig. \ref{fig:circuits}, it is plausible that the mixing time is a sub-exponential function of $d$.

One can also explore possible generalizations of our algorithm to different noise models, such as the true circuit-based noise model, see Ref. \cite{12}, and different stabilizer codes. Finally, we expect that our fitting formula for the logical error probability can be refined by taking into account the pre-exponential factor depending on $d$ as was proposed in Ref. \cite{13}.

Acknowledgments

We would like to thank Charles Bennett, Martin Suchara, and Rajan Vadalakadathu for helpful discussions. This work was supported in part by IARPA QCS program under contract number D11PC20167 and by the DARPA QuEST program under contract number HR0011-09-C-0047. Computational resources for this work were provided by IBM Blue Gene Watson supercomputer center.

\begin{thebibliography}{99}

[1] P. W. Shor, In Proc. 37th FOCS p. 56 (1996).
[2] E. Knill, arXiv:quant-ph/0404104 (2004).
[3] P. Aliferis, D. Gottesman, and J. Preskill, Quant. Inf. Comput. 6, 97 (2006).
[4] D. Aharonov, A. Kitaev, and J. Preskill, Phys. Rev. Lett. 96, 050504 (2006).
[5] H. K. Ng and J. Preskill, Phys. Rev. A 79, 032318 (2009).
[6] D. Aharonov and M. Ben-Or, in Proc. 29th Ann. ACM Symp. on Theory of Computing (ACM, 1997), pp. 176–188.
[7] A. Y. Kitaev, Russian Mathematical Surveys 52, 1191 (1997).
[8] E. Knill, R. Laflamme, and W. H. Zurek, Science 279, 342 (1998).
[9] A. Y. Kitaev, Annals of Physics 303, 2 (2003).
[10] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. 43, 4452 (2002).
[11] S. Bravyi and A. Y. Kitaev, ArXiv quant-ph/9811052 (1998).
[12] A. G. Fowler, A. M. Stephens, and P. Groszkowski, Phys. Rev. A 80, 052312 (2009).
[13] R. Raussendorf and J. Harrington, Phys. Rev. Lett. 98, 190504 (2007).
[14] H. Bombin, Phys. Rev. Lett. 105, 030403 (2010).
[15] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. 97 (2006).
[16] A. Landahl, J. Anderson, and P. Rice, arXiv:1108.5738 (2011).
[17] R. Koenig, G. Kuperberg, and B. W. Reichardt, Ann. of Phys. 325, 2707 (2010).
[18] R. Raussendorf, J. Harrington, and K. Goyal, New J. Phys. 9, 199 (2007).
[19] A. G. Fowler, Phys. Rev. Lett. 109, 180502 (2012).
[20] A. G. Fowler, Physical Review A 87, 040301 (2013).
[21] C. H. Bennett, J. Comp. Phys. 22, 245 (1976).
[22] G. Rubino and B. Tuffin, Rare event simulation using Monte Carlo methods (Wiley, 2009).
[23] C. Wang, J. Harrington, and J. Preskill, Ann. of Phys. 303, 31 (2003).
[24] H. Bombin and M. A. Martin-Delgado, J. Phys. A: Math. Theor. 42, 095302 (2009).
[25] A. G. Fowler, A. C. Whiteside, and L. C. Hollenberg, Phys. Rev. Lett. 108, 180501 (2012).
[26] G. Duclos-Cianci and D. Poulin, Phys. Rev. Lett. 104, 050504 (2010).
[27] G. Duclos-Cianci and D. Poulin, Information Theory Workshop (ITW), 2010 IEEE pp. 1–5 (2010).
[28] S. Bravyi and J. Haah, arXiv:1112.3232 (2011).
[29] A. Hutter, J. R. Wootton, and D. Loss, arXiv preprint arXiv:1302.2669 (2013).
[30] D. S. Wang, A. G. Fowler, and L. C. Hollenberg, Phys. Rev. A 83, 020302(R) (2011).
[31] A. G. Fowler, Phys. Rev. A 87, 062320 (2013).
[32] J. Ghosh, A. G. Fowler, and M. R. Geller, Phys. Rev. A 86, 062318 (2012).
[33] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Phys. Rev. A 86, 032324 (2012).
[34] B. Korte and J. Vygen, Combinatorial Optimization

\end{thebibliography}
(Springer-Verlag, 2010).

[35] J. Edmonds, Canadian J. of Mathematics 17, 449 (1965).

[36] V. Kolmogorov, Mathematical Programming Computation 1, 43 (2009).

[37] J. Edmonds, Journal of Research of the National Bureau of Standards B 69, 125 (1965).

[38] Library for Efficient Modelling and Optimization in Networks, http://lemon.cs.elte.hu/trac/lemon

[39] H. N. Gabow, J. of the ACM (JACM) 23, 221 (1976).

[40] T. M. Stace and S. D. Barrett, Phys. Rev. A 81, 022317 (2010).

[41] R. Zecchina, Physica A: Statistical Mechanics and its Applications 302, 100 (2001).

[42] L. G. Valiant, SIAM J. Comput. 31, 1229 (2002).

[43] D. Cimasoni and N. Reshetikhin, Comm. Math. Phys. 275, 187 (2007).

[44] S. Bravyi, Contemp. Math. 482, 179 (2009).

[45] A. G. Fowler, A. C. Whiteside, A. L. McInnes, and A. Rabbani, Physical Review X 2, 041003 (2012).

[46] A. G. Fowler and S. J. Devitt, arXiv:1209.0510 (2012).

[47] Any error of weight smaller than $d/2$ must be correctable by definition of the code distance.

[48] Note that a defect can carry a coherent superposition of different topological charges, as opposed to point-like anyons that must have a definite topological charge.
FIG. 10: (Color online) Noisy syndrome readout. The decoding graph corresponding to the surface code lattice shown in Fig. [1] Each vertex of the graph represents a space-time location of a syndrome measurement. Any elementary error in the syndrome readout circuit (memory, preparation, measurement, or CNOT error) is associated with some edge of the graph. The overall probability of elementary errors associated with a given edge $e$ determines the effective error rate of $e$. Red (blue) color stands for the largest (smallest) effective error rate. Some edges located near the boundary of the defects have only one end-point. To avoid clutter, such edges are represented by solid circles. The corresponding surface code has two smooth defects of linear size $r = 2$, separation $s = 5$, and buffer length $b = 3$. The number of syndrome readout rounds is $t = 3$. The actual simulations were performed for $s = b = t = 4r$ for $r = 2, 3, 4$.

FIG. 11: A fragment of the decoding graph indicated by the black rectangle on Fig. [10]