Rare Event Simulation of Quantum Error-Correcting Circuits

Carolyn Mayer,[1, *] Anand Ganti,[2] Uzoma Onunkwo,[2, †]

Tzvetan Metodi,[3, ‡] Benjamin Anker,[4, 5, §] and Jacek Skryzalin[6]

[1]*Discrete Math & Optimization, Sandia National Laboratories, Albuquerque, NM, USA*
[2]*Cyber Security Initiatives, Sandia National Laboratories, Albuquerque, NM, USA*
[3]*Quantum Computer Science, Sandia National Laboratories, Albuquerque, NM, USA*
[4]*Center for Quantum Information and Control, University of New Mexico, Albuquerque, NM, USA*
[5]*Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA*
[6]*Jump Trading Group, 1 London Wall, Barbican, London, UK*

(Dated: September 2025)

We describe a practical approach for accessing the logical failure rates of quantum error-correcting (QEC) circuits under low physical (component) failure rate regimes. Standard Monte Carlo is often the de facto approach for studying the failure rates of quantum circuits. However, in the study of fault-tolerant error-correcting circuits, the ability to extend this approach to low physical failure rates is limited. In particular, the use of Monte Carlo to access circuits that are relatively large or have high correcting power becomes more difficult as we lower the input failure rates of the individual components (gates) in the circuit. For these reasons, many simulations studying the circuit model go no lower than end-to-end logical failure rates in the 10^{-6} regime.

In this report, we outline an approach that borrows from earlier work by Bravyi and Vargo [1] to the more complex circuit noise model. Earlier works studied both the capacity and phenomenological noise models [1–3], but the work is insufficient for generating similar simulations in the circuit-noise model. To the best of our knowledge, our team is the first to develop a full prescription of the rare event simulation by splitting technique for the circuit-based noise model. We have also generated promising results that are confirmed by standard Monte Carlo simulation under an accessible regime. This work shows that we can access noise in the circuit-model prescription of quantum error-correcting code to failure rates below 10^{-20} regime.

I. INTRODUCTION

In recent years, strong interest has grown in the area of quantum computing. This has been made apparent by the amount of research funding and commercial sectors focusing on advancing the science for realizing quantum computing systems. These interests are motivated by opportunities to better arrive at solutions to challenging technological problems in the areas of pharmaceutical, cybersecurity, and quantum chemistry to name a few.

However, the advancement towards quantum computing is deemed by most to be unattainable without quantum error correction and fault-tolerant systems. Many quantum computing paradigms rely on qubits, the basis of quantum information, which are quite unreliable. Current state-of-the-art systems are looking at having few hundreds to thousands qubit systems with the ability to perform thousands of primitive quantum gate operations [4–6]. Furthermore, the application of quantum gates introduce sources of decoherence for these qubits, which must be protected. Quantum error correction (QEC) provides an avenue to protect qubits against environmental noise. With many of the advances over the last decade in hardware designs for quantum systems, today's

systems are still only pushing towards 10^{-3} infidelity for quantum operations. However, many experts believe that infidelity rates in the 10^{-12}, so-called *teraquop*, range are needed for quantum systems to show real advantage over today's dominant classical computer. QEC, along with fault-tolerant designs, could bring such hardware designs to enable quantum computing in the teraquop range. Popular QEC codes, such as surface codes [7–13] and quantum LDPC codes [14–17], show promise.

Simulation software have become imperative for assessing the performance of QEC designs on different codes and their associated decoders. These simulations enable researchers to understand the logical failure rates under different noise characteristics. Many works show simulation results under varying noise constraints that range from the code capacity noise model to the phenomenological noise model and the more realistic circuit noise model. The assessment of QEC designs usually leverages Monte Carlo style of simulations, but naïve approaches can limit the ability to assess codes in the teraquop regime. In this work, we illustrate how we can assess codes using a method pioneered by Bravyi and Vargo ([1]) on the phenomenological noise model for the punctured surface code. We improve upon their work by showing how to adapt simulations for the circuit-noise model and demonstrate a simulation feature that enable result to run orders of magnitude faster with a caching scheme that is naturally amenable to rare event simulations.

For the rest of this paper, we briefly introduce rare

---

* cdmayer@sandia.gov
† {aganti,uonunkw}@sandia.gov
‡ tsmetod@sandia.gov
§ banker@unm.edu

event simulation using the splitting method and describe the common noise models for assessing QEC and fault-tolerance in Section II. In Section III, we work out the extension of the Monte Carlo Markov Chain (MCMC) approach to the circuit noise model. The circuit noise model captures many of the effects that real quantum systems have at a level that is more representative of hardware effects than the code capacity and phenomenological noise models. In Section IV, we describe the parameters for our simulation study, which includes the case study for the popular rotated surface codes. We extensively show results from our simulations across a range of code distances and provide our insights on the agreement between the Monte Carlo and rare event simulation in the accessible regime. Finally, we conclude this work in Section VI and state the future directions for this work in studies that have qubit leakage effects and circuits with dynamic size imposed by post-selections.

II. SPLITTING METHOD

The splitting method is a Metropolis-Hastings Bayesian algorithm [18] that can much more efficiently explore a parameter space, which we adapt for the estimation of logical performance of QEC circuits under circuit-noise models. Traditionally, the assessment of logical performance of QEC circuits rely on Monte Carlo computer simulation approaches due to the many different gates involved in a QEC circuit. However, as the QEC circuit sizes increases or the physical error rates get lower, assessments using Monte Carlo simulations start becoming too time consuming. In fact, for low physical error rate (p) regimes (much below the pseudo-threshold of QEC circuits), standard Monte Carlo sampling becomes infeasible, as the number of samples needed for an estimate becomes exorbitant. For example, trying to estimate logical failure rates ($\bar{p}$) in the tera-aquop regime would rely on order of a hundred trillion independent runs. For a strictly fault-tolerant encoded circuit, $\bar{p} = \Omega\left(p^{\lceil d/2 \rceil}\right)$ as p approaches 0, where d is the distance of the $[[n, k, d]]$ code. The $[[n, k, d]]$ convention means a stabilizer code with n physical qubits, k logical qubits and minimum distance d.

To reduce the number of Monte Carlo samples needed to get an estimate, one can limit the size of gate failure sets sampled, noting that for large enough size m, the logical failure rate as a function of the physical error rate is:

$$\bar{p}(p) \approx$$
$$\sum_{k=0}^{m} \binom{G}{k} p^k (1-p)^{G-k}$$
$$\cdot \Pr(\text{logical fail}|\text{no. failing gates} = k),$$

where G is the number of quantum gates in the encoded circuit. This technique is sometimes called subset sampling [19].

For a strictly fault-tolerant circuit, the terms with $k < \lceil \frac{d}{2} \rceil$ do not contribute to the sum as they should not lead to any logical failures. For the cases with large k, i.e., $k > m$, the p^k terms become negligible that they do not contribute significantly to the sum. However, as the code distance increases, many samples are required to estimate the $\Pr(\text{fail}|\text{no. failing gates} = k)$ term. At $k = \lceil d/2 \rceil$, many of the $\binom{G}{k}$ size-k sets may not lead to failure. See Figure 1 for estimates from simulation.

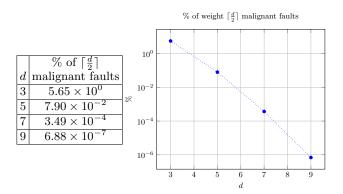|	% of $\lceil \frac{d}{2} \rceil$
d	malignant faults
3	5.65×10^0
5	7.90×10^{-2}
7	3.49×10^{-4}
9	6.88×10^{-7}

FIG. 1: Estimated percentage of weight $\lceil \frac{d}{2} \rceil$ faults leading to logical failure for a distance-d rotated surface code.

Thus, estimating the probability of logical failure to sufficient accuracy given that $k = \lceil d/2 \rceil$ quantum gates failed becomes computationally challenging. Our experience is that for large code distance, and without any guarantees of strict fault tolerance, there is a tendency for the estimator to fail on this step because it assumes that $\Pr(\text{fail}|\text{no. failing gates} = k) = 0$, so as not to be in an infinite loop cycle estimating what it assumes is zero. For instance, considering a case of 10,000 faulty quantum gates and a code distance of $d = 13$, there are potentially about 2×10^{24} samples to search for failing sets of gates of size $= \lceil 13/2 \rceil = 7$, which may number in the thousands only. That is, it would take roughly 2×10^{20} samples to get a single failing event on average, which is generally not attainable in today's classical computing systems.

The splitting method of [1] offers an alternative, although it was limited to the *code capacity* and the *phenomenological* error models (see Figure 2 for descriptions). We summarize that approach below before extending to the more complicated *circuit* model in Section III.

Consider a monotonic sequence of parameterized physical failure rates, $p \in \{p_i\}_{i=1}^t$ and denote the corresponding output failure rates by $\{\bar{p}_i\}_{i=1}^t$. If the sequence of physical failure rates is decreasing and $\bar{p}_i$ is small, it may be easier to estimate $\bar{p}_1$ and the sequence of values $\left\{\frac{\bar{p}_{i+1}}{\bar{p}_i}\right\}_{i=1}^{t-1}$ than it is to directly estimate the target output logical failure rate $\bar{p}_t$. This initial output failure rate, $\bar{p}_i$, can be estimated from a Monte Carlo simulation at the parameterized input rate of p_1 chosen to be

---

Noise Models

Code capacity: Errors can occur on data qubits. Syndrome extraction is treated as an error-free process.

Phenomenological: Errors can occur on data qubits. Syndrome extraction is treated as a black box process that allows errors on the measurements.

Circuit: Syndrome extraction is modeled via a quantum circuit with one and two qubit gates. Every gate is modeled as an ideal gate composed with a Pauli noise channel. Gates fail independently, although this can be relaxed by generalizing the sampling routine. Errors can be propagated between data and ancilla qubits during syndrome extraction, e.g., through couplings with CNOT, CPHASE, or ISWAP gates.

FIG. 2: Descriptions of three common error models.

relatively large so as that Monte Carlo can estimate in a timely manner. In general, this splitting approach can be extended to also simulate cases where the target failure rate, $\overline{p}_t$, we wish to estimate is larger than the initial output logical rate, $\overline{p}_1$. However, the latter is often not the scenario most researchers and engineers are interested in; they often desire estimates that are closer to the the teraquop regime and thus the target logical rate is much smaller than the initial failure rate estimates that Monte Carlo can achieve.

For a surface code that relies on a decoding graph, the authors prescribed a way to choose the sequence interval points for the rare event simulation using the heuristic

$$p_{i+1} = p_i \cdot 2^{\pm 1/\sqrt{w_i}}, \tag{1}$$

where $w_i = \max(d/2, p_i \cdot |\{\text{edges in decoding graph}\}|)$. The $\pm$ sign in the exponent of Equation 1 defines the direction the sequence will go as a function of where the target failure rate is with respect to the initial output failure rate. For most users, the intent would be to use rare event simulation to estimate the lower target logical failure rate, so the sequence becomes: $p_{i+1} = p_i \cdot 2^{-1/\sqrt{w_i}}$. The intuition behind choosing w_i is to start around the expected number of gates that will cause logical failures. Thus, for any quantum circuit encoded in an error-correcting code (not just the surface codes), we can adapt the splitting method so that $w_i = \max(d/2, p_i \cdot G)$, where G is the number of faulty gates in the circuit as defined before. We later illustrate in Section V that the heuristic for picking the sequence $\left\{\frac{p_{i+1}}{p_i}\right\}_{i=1}^{t-1}$ is really a suggestion for getting roughly the same ratios between sequence intervals, but other convenient choices would still lead to the same estimate of the target logical failure rate desired.

A. Notation

To describe the splitting method and prescribe the algorithm, we introduce a few notations which closely follows [1]. Let:

- Ω: sample space of all possible *events*, i.e., the powerset of all possible (gate, fault) tuples in the QEC-encoded circuit;

- $\mathcal{F}$: set of all failing (malignant) events, which are all the elements of Ω that individually lead to a logical failure in the QEC-encoded circuit;

- $\pi_i(E)$: probability of event $E \in \Omega$ at step i in the sequence of simulated p_i values;

- $\pi_{i|\mathcal{F}}$ is the conditional probability distribution over failing events defined as:

$$\pi_{i|\mathcal{F}}(E) := \begin{cases} \frac{\pi_i(E)}{\pi_i(\mathcal{F})} & \text{if } E \in \mathcal{F} \\ 0 & \text{if } E \notin \mathcal{F} \end{cases}$$

- $\mathbb{E}_{i|\mathcal{F}}(\cdot)$: expectation with respect to p_i conditioned on the failure set, $\mathcal{F}$.

B. Ratio Estimation

For some function $g : \mathbb{R}^+ \to \mathbb{R}^+$ such that $g(x) = x^{-1}g(x^{-1})$ and any constant $C > 0$,

$$\frac{\overline{p}_{i+1}}{\overline{p}_i} = \frac{\pi_{i+1}(\mathcal{F})}{\pi_i(\mathcal{F})} = C \frac{\mathbb{E}_{i|\mathcal{F}}\left[g\left(C\frac{\pi_i}{\pi_{i+1}}\right)\right]}{\mathbb{E}_{i+1|\mathcal{F}}\left[g\left(C^{-1}\frac{\pi_{i+1}}{\pi_i}\right)\right]}. \tag{2}$$

For a fixed number of samples, N, the function g minimizing the statistical error is $g(x) = \frac{1}{1+x}$ [20]. The expectations on the numerator and denominator will be approximated with empirical averages:

$$\mathbb{E}_{i|\mathcal{F}}\left[g\left(C\frac{\pi_i}{\pi_{i+1}}\right)\right] \approx \mathbb{E}_{i|\mathcal{F}}^{\text{est}}\left[g\left(C\frac{\pi_i}{\pi_{i+1}}\right)\right]$$

$$:= \frac{1}{N}\sum_{j=1}^{N} g\left(C\frac{\pi_i(E_j)}{\pi_{i+1}(E_j)}\right), \tag{3}$$

where the $E_1, \ldots, E_N \in \mathcal{F}$ are drawn with the probability distribution $\pi_{i|\mathcal{F}}$, found using a Metropolis routine to generate a reversible ergodic Markov chain (see II C).

Using the empirical estimates of the expectations, our goal is to find C such that

$$\mathbb{E}_{i|\mathcal{F}}^{\text{est}}\left[g\left(C\frac{\pi_i}{\pi_{i+1}}\right)\right] = \mathbb{E}_{i+1|\mathcal{F}}^{\text{est}}\left[g\left(C^{-1}\frac{\pi_{i+1}}{\pi_i}\right)\right]. \tag{4}$$

The C value that results in the above equality gives an estimate of the desired ratio $\frac{\overline{p}_{i+1}}{\overline{p}_i}$. This is apparent from Equation 2 because the ratio of expectations becomes unity and one is left with $\frac{\overline{p}_{i+1}}{\overline{p}_i}$ on the left-hand side and C on the right-hand side.

C. Reversible Markov Chain

The goal of the Metropolis routine is to produce for each i, a reversible (which implies *aperiodicity*) and irreducible, hence ergodic, Markov Chain (RIMC) whose stationary distribution is $\pi_{i|\mathcal{F}}$. This enables us to estimate the empirical averages on the right-hand side of Equation 3. The Metropolis routine initializes a failing set $E \in \mathcal{F}$ and uses Bennett's acceptance method [20] to produce a RIMC as follows. At each iteration:

1. Select a (gate, fault) tuple e (i.e., a set of cardinality 1 in Ω) uniformly at random with respect to the unit-cardinality sets, and let E' be the symmetric difference of E, e i.e., $E \oplus e$. For the surface code and the phenomenological noise model, e can be thought of as an edge in the decoder graph (defined in Section III); this is not the case for circuit noise models, as will be seen in Section III.

2. Compute $q = \min\left\{1, \frac{\Pr(E')}{\Pr(E)}\right\}$ and perform a Bernoulli trial with success probability q. This is the acceptance step.

3. If the Bernoulli trial resulted in success and $E' \in \mathcal{F}$, output E'. Otherwise, output E.

The routine results in a Markov Chain that satisfies the detailed balance equation:

$$\frac{\Pr(E')}{\Pr(E)} = \frac{\Pr(E'|E)}{\Pr(E|E')},$$

and the forward chain, $E \to E'$, has the same probability as with the backward chain, $E' \to E$. Furthermore the chain is irreducible since one can start with a failing set E and arrive at another E' using the elements of $E \oplus E'$. So the Markov chain is a RIMC and its probability distribution will converge to $\pi_{i|\mathcal{F}}$.

III. EXTENSION TO CIRCUITS

When using the circuit noise model, (gate, fault) pairs can introduce errors to the data qubits and the syndrome measurements. There exist a class of graph-based decoders such as the *minimum weight perfect matching* (MWPM) and *union-find* (UF) for surface codes that can be described on a weighted graph that we call the decoder graph. The vertices of the decoder graph are the X (Z) checks and the edges represent gate-faults or errors that result in the incident checks changing values. In the phenomenological noise model, selecting errors simply corresponds to selecting edges in the decoder graph. In the circuit noise model multiple gate-faults map to the same syndrome (edges). Due to this nature of circuit noise, we modify the state space of the MC in the original protocol in [1] from sets of edges to gate-faults. When doing so, we must modify the Metropolis routine from

selecting edges in the decoding graph to selecting physical (gate, fault) pairs. Our Metropolis routine selection protocol results in a reversible MC. This is because as we show below the MC transition probabilities satisfies the detailed balance equations.

Starting with a set E of (gate, fault) pairs leading to logical failure, we select a (gate, fault) pair (g, f) uniformly at random in the circuit. If g is not in any (gate, fault) pair in E regardless of the fault, then set $E' = E \cup \{(g, f)\}$. Otherwise, $(g, h) \in E$ for some fault h. In the latter case, set $E' = (E \cup \{(g, f)\}) \setminus \{(g, h)\}$. Note that if the faults f and h are the same in this latter case, then $|E'| = |E| - 1$, where the (g, f) has been dropped. The selection process ensures E' is physically viable as opposed to a symmetric difference which would allow a single gate with two different faults. For $E' \in \mathcal{F}$, the acceptance probability, $\Pr(E'|E)$, is chosen as follows:

- **Case 1**: g is not in any gate fault pair in E. Set the acceptance probability to

$$\frac{\Pr(g)}{1 - \Pr(g)} \cdot \Pr_g(f),$$

  where $\Pr(g)$ is the probability of gate g failing and $\Pr_g(f)$ is the probability of the fault f occurring given that g failed.

- **Case 2**: There exist a fault h such that $(g, h) \in E$.
  1. If $f = h$, set the acceptance probability to 1.
  2. If $f \neq h$, set the acceptance probability to $\Pr_g(f)$.

In each case, the detailed balance equation is satisfied:

- **Case 1**: g is not in any gate fault pair in E, $E' = E \cup \{(g, f)\}$.

$$\begin{aligned}
&\frac{\Pr(E')}{\Pr(E)} \\
&= \frac{\prod_{(i,j)\in E'}(\Pr(i)\Pr_i(j))\prod_{(i,j)\notin E'}(1-\Pr(i))}{\prod_{(i,j)\in E}(\Pr(i)\Pr_i(j))\prod_{(i,j)\notin E}(1-\Pr(i))} \\
&= \frac{\Pr(g)\Pr_g(f)}{1-\Pr(g)} \\
&= \frac{\frac{\Pr(g)}{1-\Pr(g)}\cdot\Pr_g(f)}{1} \\
&= \frac{\Pr(E'|E)}{\Pr(E|E')}.
\end{aligned}$$

- **Case 2**: There exist a fault h such that $(g, h) \in E$.
  1. If $f = h$, $E' = E \setminus \{(g, f)\}$ or equivalently $E = E' \cup \{(g, f)\}$. This case is the reciprocal of Case 1.
  2. If $f \neq h$:
  $$\frac{\Pr(E')}{\Pr(E)} = \frac{\Pr_g(f)}{\Pr_g(h)} = \frac{\Pr(E'|E)}{\Pr(E|E')}.$$

Note that as stated, these acceptance probabilities apply to modifying E by a single fault to get E'.

IV. SIMULATION SETUP

Here, we describe the setup for the simulation studies used in our analysis of the rare event error model. Our simulation studies were conducted with the $[[d^2, 1, d]]$ rotated surface code. This code is widely studied [7–10] for its many advantages towards a scalable quantum computer, namely locality, simplicity of the syndrome extraction, and the existence of efficient decoder families. In our studies, we use the efficient MWPM decoder but note that other efficient decoders like the *Union-Find* decoder [21, 22] can be used as well. While our studies were for the rotated surface code under the MWPM decoder, the MCMC piece of our simulation has no real dependence on decoder selection and, as such, extends to other code and decoder configurations of interest.

The smallest error-correcting member of the medial surface code can be described as: This code consists of
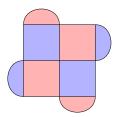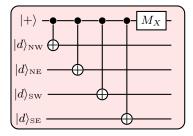
FIG. 3: A distance-3 medial surface code, with X checks and Z checks. Physical data qubits are at the corners. Each color face is a check with qubits incident.

X checks and Z checks as represented by the checkered pattern of red and blue. Figure 3 is the $[[9, 1, 3]]$ member of the family of medial surface code because it has 9 data qubits distributed at the corners of the squares and it can fix any $\frac{d-1}{2} = 1$ error. The other members of this medial surface code follow the same pattern with the grid that is required to yield a d^2 layout of data qubits, where d is the code's distance. In the circuit realization of this code, there is a need to follow certain routine for correctness and fault tolerance of syndrome extraction. In particular, we choose a syndrome extraction routine that does not permit for single faults to spread maliciously as shown in Figures 4 and 5.

The circuits above are repeated at each corresponding site for the X and Z syndrome bits, then repeated over $r = 2d$ rounds of syndrome extraction for fault-tolerance to the distance of the given code. There are a number of simulation software for studying the rotated surface code, including Stim [23]. In our studies, we perform simulation and evaluation of syndrome extraction in the following way:

1. For a $[[d^2, 1, d]]$ surface code, we perform noisy syndrome extractions for $2d$ rounds.

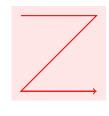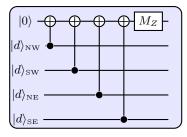2. We perform error correction only within the first d rounds.

FIG. 4: CNOT couplings for measuring a X-type stabilizer generator in the standard gate basis along with the direction of coupling operations on an X-type plaquette of a surface code patch. The subscript of NW, SW, NE, and SE on the data qubit represent the northwest, southwest, northeast, and southeast locations of the data with respect to the X ancilla in the grid layout.
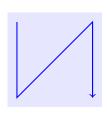

FIG. 5: CNOT couplings for measuring a Z-type stabilizer generator in the standard gate basis along with the direction of coupling operations on an Z-type plaquette of a surface code patch. The subscript of NW, SW, NE, and SE on the data qubit represent the northwest, southwest, northeast, and southeast locations of the data with respect to the Z ancilla in the grid layout.

This approach simulates logical idle and differs from "logical T1" simulation setup of Stim. The choice of applying error correction only on the first d rounds is motivated by the need to be strictly fault-tolerant. Intuitively, having a buffer of extra d rounds enables the user to distinguish between data errors and measurement errors. The syndrome information embedded in the latter d rounds can then be used for correction after an extra d rounds of syndrome extraction.

In our circuit noise model, every quantum gate, including preparations and measurement, is assumed to have a failure rate parameterized by p, the physical gate failure rates. For our simulation results, unless noted, all gates fail with probability p. It is possible to extend to the so-called `Pauli+` noise model, where gates are able to cause leakage on the input qubit(s); we do not include the `Pauli+` noise model in our study.

We perform simulations in the accessible regime of standard Monte Carlo, and rare event simulations in the accessible regime and well-beyond the accessible regime of the Monte Carlo runs. The results from these scenarios

enable us to assess the correctness of the rare event simulation technique in the regions of overlap, but then shows us results for the overall logical idle simulations, across all $2d$ rounds of syndrome extraction in the teraquop range. The largest runtime cost for these simulations tend to be for the calls to the decoder.

V. RESULTS

We present results highlighting the strengths of the rare event simulation techniques along with empirical evidence towards correctness by comparing to simple Monte Carlo simulations in accessible study regimes. We also present results to demonstrate the properties of our rare event simulation under symmetric and asymmetric circuit noise conditions, as well as under different conditions for choosing the interval points for the parameterized physical failing rates, p. The setup for the simulation results presented here follow our descriptions in Section IV.

We start by presenting the simulation results for the rotated surface code in Figure 6 under symmetric circuit noise conditions, that is, all quantum gates in the syndrome extraction circuits have the same failure probability p. Figure 6 shows the logical Z error rate $(p_{\overline{Z}})$ computed using rare event simulation compared to using Monte Carlo simulations with $p \in [10^{-4}, 10^{-3}]$. These simulations are also the basis for the results shown in Figure 8 and Figure 9. The rare event simulation can still be relied on to generate valid results at lower p as shown in Figure 7. However, this would be practically impossible to generate using a standard Monte Carlo approach. For the data points that were generated for the Monte Carlo in Figures 6 and 7, the results show agreement with the rare event simulation.

An interesting phenomenon with rare event simulation is that as the size of the code increases, the number of p values in the heuristically-generated sequence (Equation 1) for a fixed interval of interest for p_{phys} also increases. While this potentially increases the runtime, it is actually generally negligible in comparison to the Monte Carlo setup time for the initial result at p_0. Moreover, it turns out that strict adherence to the heuristically-generated sequence is not necessary. Figures 10 and 11 show comparisons of $p_{\overline{Z}}$ estimates using different sequences of p for the distance-5 rotated surface code, and 12 shows estimates for the distance-11 rotated surface code. For distance-5, using different sequences of p yields similar estimates. For distance-11, there is some difference between the estimates from different splitting schemes but we suspect that this is the effect of not yet reaching the steady state distribution in the MCMC approach.

A breakdown of the runtimes for our implementation of the rare event simulation for odd distance surface codes is shown in Figure 8. As the distance increases, the initial setup, which includes a standard Monte Carlo simula-
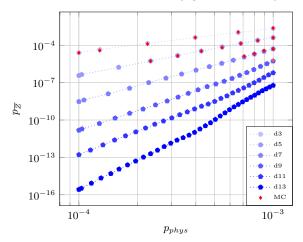

FIG. 6: Logical Z error rate for a rotated surface code for $p_{phys} \in [10^{-4}, 10^{-3}]$. Estimation using Rare Event Simulation. The red diamonds show estimates using raw Monte Carlo (MC) simulation with the number of runs needed to reach at least 1,000 failures, where the estimator is the unbiased negative binomial estimator.
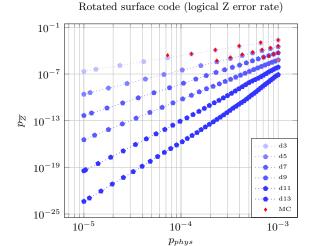

FIG. 7: Logical Z error rate for a rotated surface code for $p_{phys} \in [10^{-5}, 10^{-3}]$. Estimation using rare event simulation. The red diamonds show estimates using Monte Carlo (MC) simulation with at least 1,000 failures.

tion at the starting point where $p = 10^{-3}$ accounts for the bulk of the simulation time. Note that this starting point is the easiest/fastest for the Monte Carlo simulation because fewer runs are needed to generate a high quality estimate of the initial logical failure rate for seeding the rare event simulation. For higher distance codes, starting at a larger p (chosen below the pseudo-threshold[24]) may speed up the simulation. We also note that while the Monte Carlo simulation can be sped up through par-

allelization (e.g. using MPI or threads), our implementation of the rare event algorithm is not amenable to parallelization within a MCMC chain; however, parallelization helps with the quality of the estimator as used by convergence techniques like the Gelman-Rubin diagnostic [25].
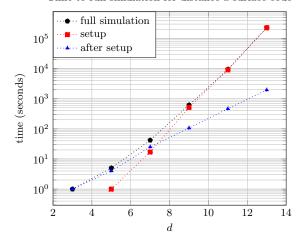

FIG. 8: Cumulative time to run the simulation with $p_{phys} \in [10^{-4}, 10^{-3}]$. The setup time includes running a Monte Carlo simulation at $p_{phys} = 10^{-3}$ to generate the initial failing events and to get the initial estimates of the logical failure rates, $p_{\overline{X}}$ and $p_{\overline{Z}}$.

Performing decoding, even with the MWPM decoder, is a relatively expensive step for the failure rate estimation. Compared to Monte Carlo simulations in which each sample is checked for failure, the rare event simulation relies on many fewer calls to the decoder. Figure 9 shows the cumulative number of calls to a matching (MWPM) decoder to estimate the logical failure rate for a rotated surface code at p in $[10^{-4}, 10^{-3}]$. The rare event simulation starts with a Monte Carlo simulation for $p = 10^{-3}$, and successive smaller physical error rates use the estimate from the previous (larger) physical error rate. We note that the frequency with which events in Ω of different cardinality are visited in the Markov chain changes with p. This is illustrated in Figure 13. At lower p, a larger portion of the events have cardinality $\lceil \frac{d}{2} \rceil$. This behavior is expected due to the fact that the expected number of failing gates goes as $G \times p$, where G is the number of quantum gates in the encoded circuit. However, when p becomes smaller, this expected number of failing gates is below the correction power of the code, $\lceil \frac{d}{2} \rceil$, so the rare event simulation steadies around events with cardinality of $\lceil \frac{d}{2} \rceil$.

For the simulations shown in the previous figures, the circuit noise was symmetrical. However, the rare event simulation easily extends to cases with asymmetrical noise, such as if qubits in certain location of a surface code are more prone to failure than others, as might be expected in actual hardware (see, e.g., [8]). Figures 14(b)
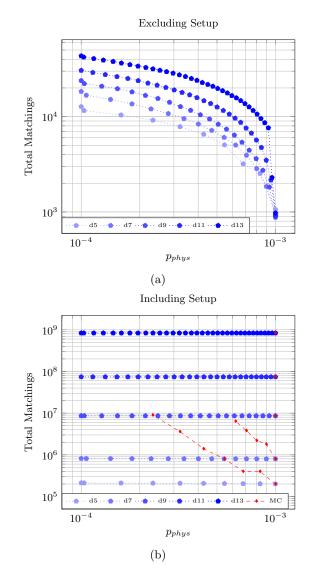

(a)

(b)

FIG. 9: Total Cumulative Matchings for $p_{\overline{X}} + p_{\overline{Z}}$ estimate (using cache). (a) Excluding the setup in which initial events and $p_{\overline{X}}$ and $p_{\overline{Z}}$ are obtained using a Monte Carlo simulation at $p_{phys} = 10^{-3}$ or (b) including the setup. The ♦ points show numbers of matchings to get an estimate using Monte Carlo simulation with at least 100 failures at a specified p_{phys} (not accumulated across different p_{phys}).

and 14(c) show logical X ($p_{\overline{X}}$) and Z error rates for a distance-11 rotated surface code in which the checks highlighted in Figure 14(a) are more likely to fail. Figure 15 shows a portion of the same plots with a comparison to estimates found using standard Monte Carlo simulations.
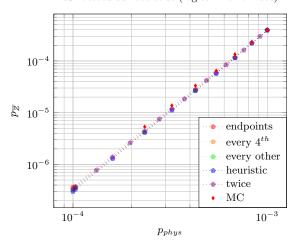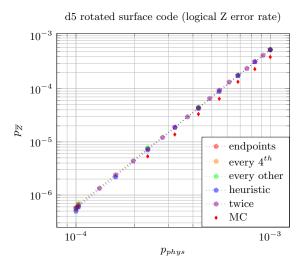
FIG. 10: Logical Z error rate for a distance-5 rotated surface code for $p_{phys} \in [10^{-4}, 10^{-3}]$. Estimation using Rare Event Simulation with sequences of physical failure rates. The legend entries refer to the points taken relative to the splitting heuristic. For the "twice" points, a point is added halfway between each point of the heuristic. The ◆ points show Monte Carlo estimates.

FIG. 11: Logical Z error rate for a distance-5 rotated surface code for $p_{phys} \in [10^{-4}, 10^{-3}]$. Estimation using Rare Event Simulation with sequences of physical failure rates without expiration of events.

VI. CONCLUSIONS AND FUTURE WORK

A. Future Work

We identify several directions for future work.
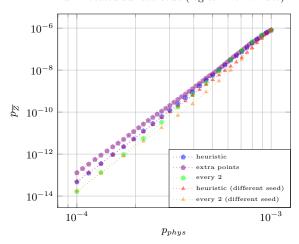
FIG. 12: Logical Z error rate for a distance-11 rotated surface code for $p_{phys} \in [10^{-4}, 10^{-3}]$. Estimation using Rare Event Simulation with sequences of physical failure rates without expiration of older events.

1. Confidence and Convergence

In the rare event simulation, the ratio C satisfying Equation 4 is estimated for each consecutive pair of physical error probabilities in the splitting sequence. Since the ratios are multiplied to estimate the logical failure rate, errors in these estimates accumulate. The matters of ensuring that enough jumps have been taken for convergence of the Markov chains and quantifying confidence for estimates remains open.

A diagnostic for convergence such as the Gelman-Rubin statistic [25] can provide an intuitive, computable test, but does not provide a measure of confidence while also being sensitive to iterative statistics following a long tail or with different within-chain variance.

2. Splitting Heuristics

To determine sequences of p values, we used the heuristic in Equation 1 chosen by [1] for its tradeoff between the number of splitting steps and statistical error. In cases with asymmetric noise, one could study a tradeoff between statistical error and the number of p values.

3. Extension for Postselection

For circuits with postselection, such as the ones used in state preparation, a Markov chain in which the failing events differ by a single (gate, fault) pair may not be ergodic. In particular, the chain will not be irreducible if more than one fault is needed to escape post-selection. To allow for full exploration of the state space, one approach would be to allow consecutive events to differ by

d7, $\overline{Z}$, $p_{phys} = 10^{-3}$,

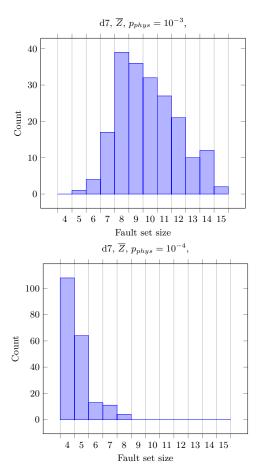d7, $\overline{Z}$, $p_{phys} = 10^{-4}$,

FIG. 13: Count histograms of sizes of fault sets visited in estimation of $p_{\overline{Z}}$ at (top) $p_{phys} = 10^{-3}$ and (bottom) $p_{phys} = 10^{-4}$ for a distance-7 surface code. Note that at the lower p_{phys}, faults close to size $\left\lceil \frac{d}{2} \right\rceil$ account for a larger portion of events.

more than a single (gate, fault) pair. When doing so, the candidate events and acceptance probabilities must be selected such that detailed balance is satisfied.

4. *Extension to Leakage Error Models*

A `Pauli+` noise model introduces more errors and ways for errors to spread (see, e.g., [4]). Randomness in the interactions may lead to different failure outcomes stemming from the same initial set of (gate, fault) pairs. In a simulation of rare events, one might expand the sample space Ω to include tuples accounting for leakage-induced interactions stemming from a fault in addition to a gate and fault.

B. Conclusions

We have outlined an approach extending earlier work by Bravyi and Vargo [1] to the more complex circuit noise

(a)

(b)   (c)
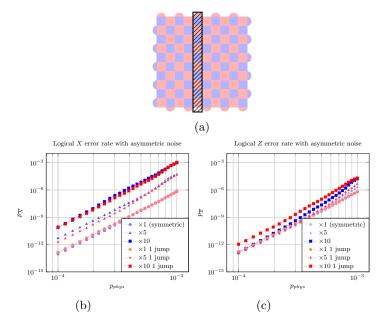
FIG. 14: Logical X and logical Z error rates for a distance-11 rotated surface code with asymmetric noise using a minimum of 10 jumps (number of times a new event is output in the Markov chain) or 1 jump in at least 18 of 20 processes before moving to the next p_{phys}.

The physical error in the column marked in (a) is multiplied by the value indicated in the legend. While using 1 jump returns results faster, doing so may not provide sufficiently accurate results. This was simulated using a different matching decoder from the one used in the symmetric case.
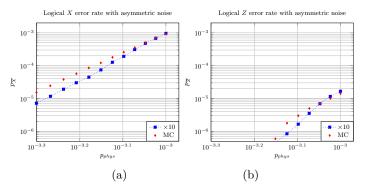

(a)   (b)

FIG. 15: Parts of the plots in Figure 14 with 10× the noise in a column and ≥ 10 jumps compared to Monte Carlo estimates. The Monte Carlo process use at least 1,000 failures (of any type) to estimate $p_{\overline{X}}$ and $p_{\overline{Z}}$.

model. For our extension to the circuit model, we performed simulation for logical failure rates below 10^{-20} regime and showed agreement with Monte Carlo simulations in a more accessible ($> 10^{-6}$) regime. Our work has provided a technique for which other simulation software can be developed for studying teraquop logical failure rates deemed useful for meaningful quantum computa-

tion under realistic noise conditions.

ACKNOWLEDGMENTS

[1] S. Bravyi and A. Vargo, Simulation of rare events in quantum error correction, Phys. Rev. A **88**, 062308 (2013).

[2] M. E. Beverland, B. J. Brown, M. J. Kastoryano, and Q. Marolleau, The role of entropy in topological quantum error correction, Journal of Statistical Mechanics: Theory and Experiment **2019**, 073404 (2019).

[3] J. P. Bonilla Ataides, D. K. Tuckett, S. D. Bartlett, S. T. Flammia, and B. J. Brown, The XZZX surface code, Nature communications **12**, 2172 (2021).

[4] Google Quantum AI and Collaborators, Quantum error correction below the surface code threshold, Nature **638**, 920 (2025).

[5] M. AbuGhanem, IBM quantum computers: evolution, performance, and future directions, The Journal of Supercomputing **81**, 687 (2025).

[6] A. Di Meglio, K. Jansen, I. Tavernelli, C. Alexandrou, S. Arunachalam, C. W. Bauer, K. Borras, S. Carrazza, A. Crippa, V. Croft, *et al.*, Quantum computing for high-energy physics: State of the art and challenges, Prx quantum **5**, 037001 (2024).

[7] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, Journal of Mathematical Physics **43**, 4452 (2002).

[8] Google Quantum AI, Suppressing quantum errors by scaling a surface code logical qubit, Nature **614**, 676 (2023).

[9] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Phys. Rev. A **86**, 032324 (2012).

[10] D. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter, Surface code quantum computing by lattice surgery, New Journal of Physics **14**, 123011 (2012).

[11] B. M. Varbanov, M. Serra-Peralta, D. Byfield, and B. M. Terhal, Neural network decoder for near-term surface-code experiments, Physical Review Research **7**, 013029 (2025).

[12] C. Gidney, M. Newman, P. Brooks, and C. Jones, Yoked surface codes, Nature Communications **16**, 4498 (2025).

[13] N. Delfosse, Decoding color codes by projection onto surface codes, Physical Review A—Atomic, Molecular, and Optical Physics **89**, 012317 (2014).

[14] N. P. Breuckmann and J. N. Eberhardt, Quantum low-density parity-check codes, Prx Quantum **2**, 040101 (2021).

[15] P. Panteleev and G. Kalachev, Quantum LDPC codes with almost linear minimum distance, IEEE Transactions on Information Theory **68**, 213 (2021).

[16] I. Dinur, M.-H. Hsieh, T.-C. Lin, and T. Vidick, Good quantum LDPC codes with linear time decoders, in *Proceedings of the 55th annual ACM symposium on theory of computing* (2023) pp. 905–918.

[17] N. Delfosse, V. Londe, and M. E. Beverland, Toward a union-find decoder for quantum LDPC codes, IEEE Transactions on Information Theory **68**, 3187 (2022).

[18] S. Chib and E. Greenberg, Understanding the Metropolis-Hastings algorithm, The American Statistician **49**, 327 (1995).

[19] S. Heußen, D. Winter, M. Rispler, and M. Müller, Dynamical subset sampling of quantum error-correcting protocols, Physical Review Research **6**, 013177 (2024).

[20] C. H. Bennett, Efficient estimation of free energy differences from Monte Carlo data, Journal of Computational Physics **22**, 245 (1976).

[21] N. Delfosse and N. H. Nickerson, Almost-linear time decoding algorithm for topological codes, Quantum **5**, 595 (2021).

[22] N. Delfosse and M. B. Hastings, Union-Find Decoders For Homological Product Codes, Quantum **5**, 406 (2021).

[23] C. Gidney, Stim: a fast stabilizer circuit simulator, Quantum **5**, 497 (2021).

[24] The pseudo-threshold for a code is the point below which the logical failure rate is below the physical gate failure rate.

[25] A. Gelman and D. B. Rubin, Inference from Iterative Simulation Using Multiple Sequences, Statistical Science **7**, 457 (1992).