Fault-Tolerant Postselection for Low-Overhead Magic State Preparation

Héctor Bombín, Mihir Pant, Sam Roberts, and Karthik I. Seetharam*
PsiQuantum, Palo Alto, California 94304, USA

(Received 8 December 2022; accepted 11 September 2023; published 4 January 2024)

We introduce a framework for fault-tolerant postselection (FTPS) of fault-tolerant codes and channels—such as those based on surface codes—using soft-information metrics based on visible syndrome and erasure information. We introduce several metrics for ranking configurations of syndromes and erasures. In particular, we introduce the *logical gap* (and variants thereof) as a powerful soft-information metric for predicting logical error rates of fault-tolerant channels based on topological error-correcting codes. The logical gap is roughly the unsigned weight difference between inequivalent logical corrections and is adaptable to any tailored noise model or decoder. We deploy this framework to prepare high-quality surface-code magic states with low overhead under a model of independent and identically distributed (IID) Pauli and erasure errors. Postselection strategies based on the logical gap can suppress the encoding error rate (EER) of a magic state preparation channel to the level of the physical error rate with low overhead. For example, when operating at 60% of the bulk threshold of the corresponding surface code, an overall reduction of the EER by a factor of 15 is achievable with a relative overhead factor of < 2 (approximately 23 times less than that of simple syndrome-counting rules). We analyze a schematic buffer architecture for implementing postselection rules on magic state factories in the context of magic state distillation. The FTPS framework can be utilized for mitigating errors in more general fault-tolerant logical channels.

DOI: 10.1103/PRXQuantum.5.010302

I. INTRODUCTION

Universal quantum computers are capable of solving problems that are otherwise intractable for conventional computers [1–3]. In order to perform the large-scale computations that are required to solve such problems, fault tolerance and quantum error correction are required to counter the effects of noise and errors that occur throughout the computation. However, the overheads (both the number of qubits and the time) to perform fault-tolerant quantum computation using currently available techniques, such as those based on the surface code [4], are very large. Reducing these overheads while simultaneously increasing error tolerance is a critical and active area of research in the pursuit of practical quantum computation. One of the significant contributors to this large overhead is the production of magic states, a necessary ingredient for computational universality in many fault-tolerant protocols. In this paper, we design new resource-efficient protocols for magic state preparation, which can reduce the overheads for magic state distillation—one of the most overhead-expensive primitives of a fault-tolerant quantum computer.

Postselection is an essential ingredient in many universal schemes of fault-tolerant quantum computation. For fault-tolerant architectures based on two-dimensional (2D) topological stabilizer codes, such as surface codes [4–18] (and related approaches [19–33]), it is ubiquitous; in order to perform logical non-Clifford gates, magic state distillation and injection are used [34–42], heavily utilizing postselection in the process. In particular, to prepare magic states that are required for injection, many noisy magic states are fed into a magic state distillation protocol, producing fewer magic states of significantly higher quality as a result of postselection—the high-quality magic states are only output if certain error-detecting measurements in the protocol do not flag the presence of an error.

For a given protocol, the total overhead of magic state distillation is strongly dependent on the quality of the initial noisy magic states. For example, to first order, the well known 15-to-1 distillation protocol takes initial magic states with error rate p and produces fewer magic states with error $35p^3$ [34,35]. If $p < 10^{-3}$, to reach a target logical error rate per logical operation of 10^{-14} as is needed in, e.g., quantum chemistry applications [43–46], one

*Lead author: karthik@psiquantum.com

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

typically needs to iterate this 15-to-1 procedure twice (i.e., two rounds of distillation). An additional round is needed if the physical error rate is $p = 10^{-2}$. Hence, the initial magic state quality can severely affect the resource overhead, with a large penalty incurred every time an additional round is added. This is under an idealized model with perfect Clifford gates utilized in the distillation protocol.

As both the magic states and gates in the distillation protocol are imperfect, they will be encoded in a quantum error-correcting code—such as the surface code [4–6]. These additional sources of noise reduce the performance of distillation and need to be accounted for to determine the overhead and output magic state(s) logical error rate(s). For instance, encoding a magic state in a quantum error-correcting code, such as the surface code, introduces additional *encoding error* that can be mitigated (using postselection strategies) in accordance with the desired overall error rate for a noisy encoded magic state. Hence, when estimating (or optimizing) the overhead for distillation, one should also include the cost for preparing the initial magic states at a desired quality.

Initial work by Li [47] demonstrates a reduction in the encoding error rate (EER) under circuit-level noise using a simple postselection scheme based on syndromes. In particular, the magic state preparation is only accepted if no syndromes are observed after a round of stabilizer measurements. This protocol produces low-error-rate magic states (particularly when the noise model is dominated by two-qubit errors), although a general analysis of encoding error versus overhead is absent. Singh *et al.* [48] modify the no-syndrome postselection protocol of Li by encoding the initial magic state in a small repetition code, yielding a reduction of the overall preparation block error rate when considered under a heavily biased noise model.

In this paper, we introduce a general framework for *fault-tolerant postselection* (FTPS) of surface-code channels (also known as logical blocks in Ref. [18]) along with several efficient rules for postselection based on soft information obtained from the visible syndrome and erasure. We apply these rules to the problem of preparing magic states encoded in the surface code. In particular, we find that for postselection rules based on the *logical gap* and its derivatives, we can improve the quality of the initial encoded magic states by suppressing the EER by orders of magnitude with modest additional overhead, under an independent and identically distributed (IID) model of Pauli and erasure errors, and over a wide range of error rates. For example, we see that when the physical error rate is approximately 60% of the bulk threshold, we can suppress the EER to that of the physical error rate using a postselection overhead of less than 2. This constitutes an overall reduction of the EER by a factor of approximately 15, leading to significant resource savings in the overall magic state distillation protocol.

II. MODELS OF FAULT-TOLERANT QUANTUM COMPUTATION

All models of fault-tolerant quantum computation use redundancy to protect information from noise. Practical schemes for fault tolerance then use measurements to infer where errors have occurred in a logical qubit to suppress the probability of logical error. Circuit-based quantum computation (CBQC) and fusion-based quantum computation (FBQC) [49] are two schemes for fault-tolerant quantum computation that differ in the types of binary-outcome measurements that are performed in order to infer the likelihood of different configurations of physical errors. Here, we focus on stabilizer-based fault tolerant schemes [50,51], where the binary-outcome measurements are Pauli operators.

In conventional surface-code-based CBQC [6], nondestructive four-qubit binary measurements are repeatedly performed on redundantly encoded static qubits. In the absence of errors, a measurement is expected to return the same outcome at all times. Therefore, a measurement that returns a different outcome compared to the previous instance tells us that at least one of the four qubits in the measurement saw an error and such deviations are called syndromes. These measurements are performed without measuring the logical information of the logical qubit and the outcomes can then be used by a decoder to reduce the probability of a logical error. The decoder is a classical algorithm that produces a Pauli recovery operation that explains the observed discrepancy (if any) in measurement outcomes (syndromes). The correction operation can be tracked classically and does not need to be physically applied to the qubits on every code cycle.

In FBQC, destructive two-qubit measurements called "fusions" are performed on small entangled states called "resource states" to infer where errors have occurred. In this paper, we use the "6-ring" FBQC protocol from Ref. [49] in which the resource states are stabilizer states that can be represented as a graph state in a form of a ring of six qubits with stabilizers $\langle Z_i X_{(i+1) \bmod 6} Z_{(i+2) \bmod 6},\ i \in \{0,1,2,3,4,5\}\rangle$ and every fusion measures two bits of information corresponding to the XX and ZZ measurements. In both FBQC and CBQC, the redundancy in the measurements can be represented as a "syndrome graph." In FBQC, each edge in the syndrome graph represents one binary measurement outcome from a fusion and every vertex is a "check," the value of which is the parity of incident edges. In the absence of errors, every check has an even parity from incident edges and, similarly to CBQC, odd parity checks are called syndromes and are used by the decoder (and its associated correction) to reduce the probability of logical error.

The choice of fault-tolerance scheme depends on the physical operations that are most natural for the hardware platform being used to build the quantum computer. Here,

the results are presented for a FBQC scheme. However, the methods presented herein allow efficient postselection in any syndrome-graph-based fault-tolerance scheme, including both CBQC and FBQC. For further discussion on how to map between the two schemes, see Ref. [52].

III. MAGIC STATE PREPARATION

To perform a distillation protocol with fault-tolerant gates, we require the input (noisy) magic states to be encoded. Here, we are specifically interested in fault-tolerant computations based on the surface code [4–6]. The surface code is a stabilizer code [50,51], meaning that it is defined by an Abelian subgroup $\mathcal{S}$ of the Pauli group $\mathcal{P}_n$ (on n qubits), not containing $-I$. Here, we consider the Wen version [53] (or $ZXXZ$ version [54]). It is defined by placing a qubit on the vertices of a square lattice, with one stabilizer generator per plaquette, formed as a product of Pauli $ZXXZ$ on the four qubits in its support. By introducing boundaries of the code, as depicted in Fig. 1, the code defines one logical qubit, with logical operators $\overline{X}$ and $\overline{Z}$ defined as strings of Pauli operators spanning opposite boundaries (also depicted in Fig. 1).

The preparation of these noisy magic states can be phrased as an encoding problem. Namely, letting Q be the state space of the noisy *initial magic state* qubit and letting X and Z be the single-qubit Pauli operators acting on it, we define a protocol to implement the following encoding isometry:

$$\mathcal{E} : Q \to Q^{\otimes d^2}, \text{ such that } X \mapsto \overline{X},\ Z \mapsto \overline{Z}, \qquad (1)$$

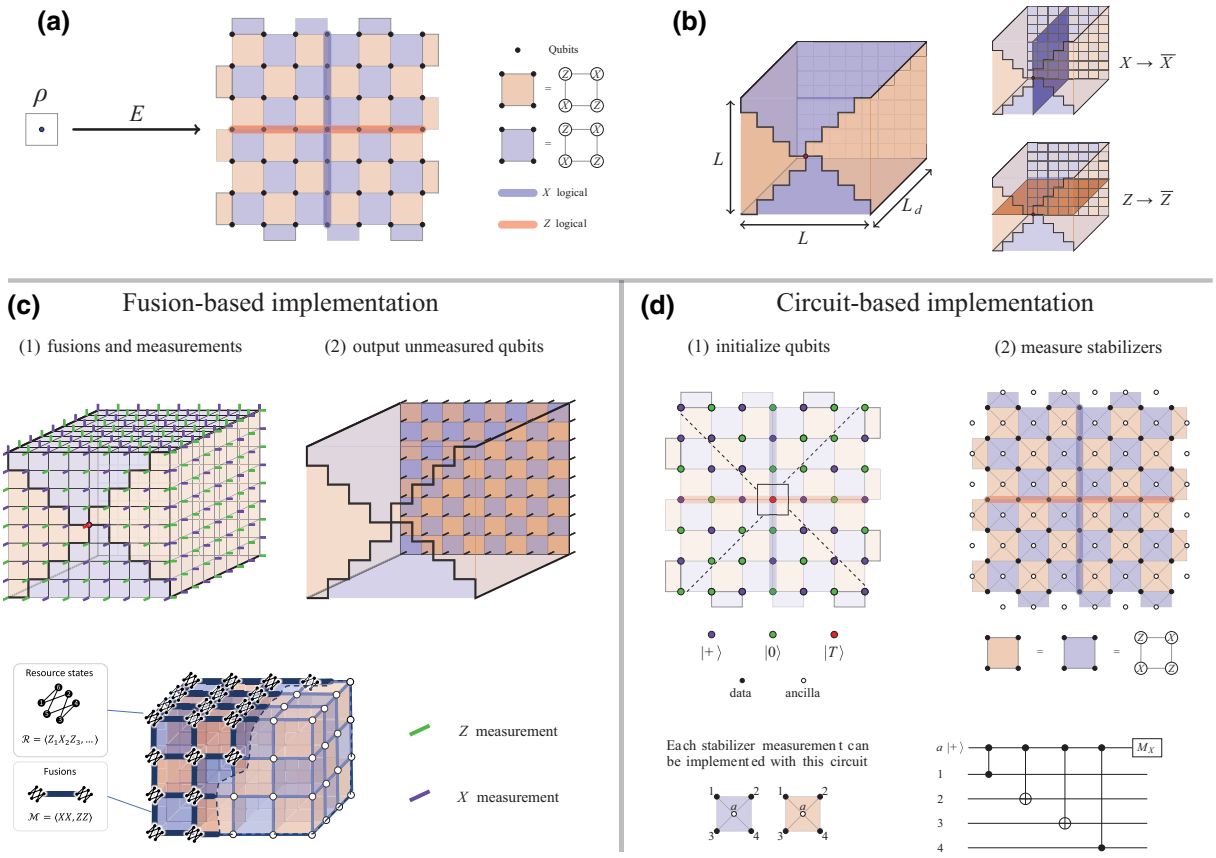

FIG. 1. (a) Noisy magic state preparation can be regarded as a channel taking a distance-1 code state (i.e., an unencoded state) to a distance-L code state. (b) The space-time diagram for the location of boundaries, where blue and red denote the two distinct boundaries. The noisy initial magic state supported on the central qubit on the front is encoded in a surface code on the rear. We remark that the surface code on the boundary has a local basis that depends on the orientation of the boundary (e.g., it may be the $ZXXZ$ or $XZZX$ version of the surface code). The membrane showing how the X (Z) operator of the initial magic state is mapped to a logical $\overline{X}$ ($\overline{Z}$) operator on the surface code is shown in blue (red). (c) The measurement pattern for implementation in FBQC using the 6-ring fusion network [18,57]. Individual qubits belonging to resource states on the boundary are measured in the X (Z) basis as accordingly depicted by the purple (green) edges. To prepare an encoded $|T\rangle = 1/\sqrt{2}(|0\rangle \pm e^{i\pi/4}|1\rangle)$ on the output, the qubit belonging to the central resource state is measured in the $1/\sqrt{2}(X+Y)$ basis; the $\pm$ sign is determined by the measurement outcomes. (d) The initial configuration of qubits for a circuit-based implementation with a planar array of qubits. Measurement of the surface-code stabilizers implements results in the desired encoding.

where $\overline{X}$ and $\overline{Z}$ are the logical operators of the surface code. We outline two approaches to achieve this encoding isometry—one based on fusion-based quantum computation (FBQC) with the 6-ring fusion network [49] and the other based on circuit-based quantum computation (CBQC) with a planar array of qubits. These approaches build upon the proposal of Ref. [55] and are also discussed in Refs. [18,56].

A. Preparation protocol

We follow the construction in Ref. [18]. The protocol contains two parameters, L and L_d. We refer to L as the "distance" of the scheme—it determines the code distance of the surface-code state we are preparing. We refer to L_d as the "depth" of the scheme—it can be thought of as simulated time, i.e., the number of rounds of stabilizer measurements in CBQC, or the number of layers of resource states in FBQC, and it determines the number of stabilizer checks in the protocol from which we can gather soft information for postselection. One may choose a minimal depth of $L_d = 2$, as is done in Refs. [47,48], allowing the encoded magic states to be prepared. However, we consider longer depths (which requires more temporal overhead in both FBQC and CBQC for the construction of a single block [58]) to allow for more information to be collected in order to better predict logical errors on the output state.

1. Circuit-based protocol

In CBQC, the preparation protocol is described in Fig. 1(d). We begin with a 2D $L \times L$ array of qubits. One of these qubits is prepared as (a noisy version of) the initial magic state $|T\rangle$, while the remaining qubits are prepared in an eigenstate of Pauli-X or -Z according to the figure. We then perform L_d repeated rounds of surface-code stabilizer measurements. The state after these measurements is an encoded version of the initial magic state qubit.

One can verify that this produces the intended encoded state as follows. First, note that the $|T\rangle$ state is a $+1$ eigenstate of $1/\sqrt{2}(X + Y)$. Consider the logical $\overline{X}$ and $\overline{Z}$ representatives for the surface code depicted in Fig. 1. During the initialization step, for these representatives, the state is a $+1$ eigenstate of $1/\sqrt{2}(\overline{X} + \overline{Y})$, owing to the particular configuration of $|0\rangle$ and $|+\rangle$ data qubits that match the stabilizer patterns in $\overline{X}$ and $\overline{Z}$ such that $X \sim \overline{X}$ and $Z \sim \overline{Z}$ (in other words, single-qubit Pauli X and Z are stabilizers equivalent to logical $\overline{X}$ and $\overline{Z}$). These representatives commute with the surface-code stabilizers (by definition) and therefore the state after measurement, which is an eigenstate of the surface-code stabilizers, remains a $+1$ eigenstate of $1/\sqrt{2}(\overline{X} + \overline{Y})$.

2. Fusion-based protocol

In FBQC [18,49], the bulk of the preparation block consists of 6-ring resource states that are fused along a cubic lattice of size $L \times L \times L_d$, with each pair of qubits from adjacent resource states in each of the three orthogonal directions undergoing a two-way fusion, i.e., a Bell measurement (e.g., XX and ZZ measurements) as in Fig. 2. Boundaries are formed by single-qubit measurements in an alternating X and Z pattern, with the distinction between primal (blue) and dual (red) boundaries given by a translation of the alternating pattern by one site (or, alternatively, flipping the X and Z measurements), as in Fig. 1(c). There is redundancy among the measurement outcomes; certain measurements may be multiplied together to form a *check* operator, the outcome of which can be used to detect errors. More precisely, check operators are elements of both the (joint) stabilizer group of the resource states as well as the measurement group (which includes fusions and boundary measurements) [49]. One may multiply the measurement outcomes comprising a check to construct the *syndrome*—in the absence of error, these syndrome measurements should have even parity and, as such, an odd parity signals the presence of one or more errors.

To complete the protocol, a single qubit in the resource state at the preparation point on the input port is measured

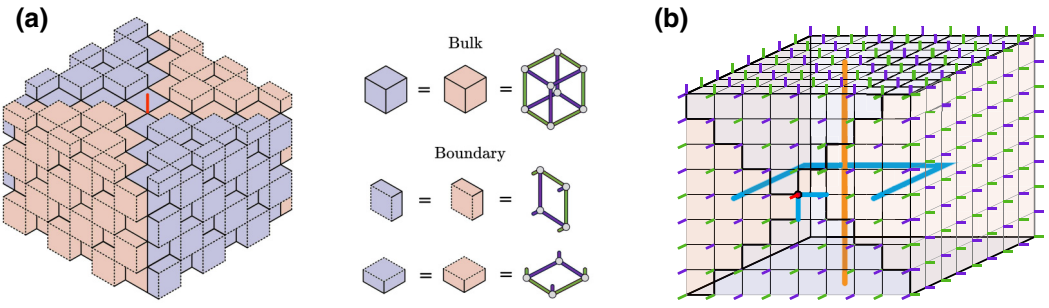

FIG. 2. (a) The check-operator structure of the preparation protocol in the 6-ring fusion network. There is a check operator for each cube of the fusion network (suitably truncated for boundaries). Here, the block has a different orientation (and, as such, the measurement pattern is slightly different), such that "time" flows from top to bottom. (b) Examples of logical errors for the preparation protocol. Chains of flipped fusion outcomes between distinct boundaries lead to logical errors. In particular, there are weight-2 logical errors supported near the central magic state. Red (blue) error chains flip red (blue) check operators and logical membranes.

in the magic state basis $TXT^\dagger = 1/\sqrt{2}(X + Y)$. This yields an initial magic state $T|\pm\rangle$ qubit that is entangled with the rest of the block via the bulk two-way fusions (where $|\pm\rangle$ is the $\pm$ eigenstate of X and is determined by the measurement outcome of the $1/\sqrt{2}(X + Y)$ measurement). The output of this channel is an encoded (noisy) $|T\rangle$ state on surface code supported on the remaining unmeasured qubits, up to a Pauli operator depending on fusion and measurement outcomes.

We can verify that this protocol produces an encoded magic state by analyzing the stabilizers of the postmeasured state. Consider the output unmeasured qubits of the fusion-based implementation in Fig. 1. On these qubits, we have a checkerboard pattern of boundary check operators from Fig. 2(a), the resource states of which are all $+1$ eigenstates. Since these checks also commute with the fusions, the postmeasured state is an eigenstate of these boundary operators restricted to the output unmeasured qubits, which are identical to the surface-code stabilizers. Therefore, the output state is a surface-code eigenstate. To determine that it is indeed an encoded magic state, we make use of the logical membrane operators, the resource states of which are also $+1$ eigenstates. The supports of these logical membranes are depicted in the top left of Fig. 1 and include the central qubit. Consider performing all measurements apart from the central $M_c = 1/\sqrt{2}(X_c + Y_c)$ measurement (which includes both the bulk fusions and the single-qubit X, Z measurements), where X_c and Z_c are single-qubit Pauli operators on the central qubit. The logical membranes commute with all of these measurements and thus the postmeasured state is an eigenstate of $X_c \otimes \overline{X}$ and $Z_c \otimes \overline{Z}$, where $\overline{X}$ and $\overline{Z}$ are surface-code logical operators on the output qubits. Thus, we have a Bell state between an unencoded qubit on the input port and a surface code on the output port. Performing a measurement of M_c on the central qubit produces the encoded state as desired.

3. Space-time diagram

An abstract space-time diagram of this channel used to achieve $\mathcal{E}$ is depicted in Fig. 1 (b). In particular, time can be thought of as running into the page, with the noisy initial magic state situated in the center of the input port (initial time slice), which we call the *preparation point*, and with the encoded magic state supported on the output port (final time slice). The operator X (Z) on the input port is mapped to $\overline{X}$ ($\overline{Z}$) on the output port via the logical membrane. Here, following Ref. [18], a logical membrane is the world sheet of a logical operator. It specifies how the input and output logical operators are correlated.

B. Distances and logical errors

The fault distance of the protocol that we present is constant, as there is a space-time volume around the preparation point where low-weight errors can give rise to logical errors. In particular, for the FBQC protocol, barring the initial magic state measurement itself, the fault distance is 2; minimally, two fusion outcomes neighboring the initial magic state measurement can be flipped in an undetectable way, yielding a logical error. Such a minimal error is shown in Fig. 2 in addition to other representative nontrivial logical errors. In CBQC (or measurement-based quantum computation), the corresponding protocol has a fault distance of 3, meaning that three single-qubit Pauli errors can introduce a logical error (see, e.g., Ref. [56, Fig. 13]). In principle, a depth of 2 is sufficient to produce an encoded magic state. In practice, however, choosing a larger depth provides more syndrome information to more reliably detect and correct such errors.

C. Error model

It is helpful to separate the overall preparation error into the error on the initial magic state and the error on the remainder of the channel. Assuming that both of these errors occur independently, the overall error of the magic state preparation block becomes

$$p_{\text{prep}} = p_{\text{init}}(1 - p_{\text{enc}}) + (1 - p_{\text{init}})p_{\text{enc}}, \tag{2}$$

where p_{init} is the error rate on the initial magic state (measurement) and p_{enc} is the error rate on the remainder of lattice that encodes the magic state in a surface code on the output port; we call the latter the encoding error rate (EER). To leading order in the low-error regime, $p_{\text{prep}} \approx p_{\text{init}} + p_{\text{enc}}$. We consider the situation where p_{init} is fixed, typically by physical hardware and design choices in the architecture, and we focus on reducing p_{enc} via postselection, which, as we will see later, is by far the dominant source of error contributing to p_{prep} in the absence of FTPS [59]. Therefore, in the following, we ignore p_{init} and consider only p_{enc} as the logical error rate on the encoding lattice arising from IID erasure and Pauli errors on every edge of the lattice. In FBQC, this error model corresponds to each measurement in a fusion suffering an erasure with probability p_{erasure} or suffering a Pauli error with probability p_{error} conditioned on not being erased. In CBQC, this error model corresponds to an erasure error on each qubit (or measurement), e.g., arising from a leakage error, or a Pauli error on each qubit (or measurement) conditioned on not being erased. Here, we will discuss only Pauli errors, relegating the discussion of mixed erasure and Pauli errors to Appendix B 2. In a similar vein, by Eq. (2), decreasing p_{init} by improving the quality of the initial magic states while not mitigating p_{enc} will also lead to diminishing returns when $p_{\text{init}} \ll p_{\text{enc}}$. Hence, it is desirable to mitigate both sources of error. Here, we address how to systematically suppress the EER p_{enc}, given a fixed p_{init}.

Given the constant fault distance of 2 when ignoring the initial magic state measurement in the FBQC preparation block, even with postselection, we cannot hope to reduce the EER p_{enc} of the channel to below $\mathcal{O}(p_{\text{error}}^2)$. In practice, this is not a bottleneck, as p_{prep} will always be limited by the quality of the initial magic state p_{init}, which is proportional to the single-qubit error rate p_{error}. Our goal is to reduce the EER p_{enc} by as much as possible.

IV. FAULT-TOLERANT POSTSELECTION

In this section, we introduce the general framework of fault-tolerant postselection and define a set of postselection rules for encoding magic states in surface codes. For a given logical block B (in any model of computation—CBQC, FBQC, or measurement-based quantum computation), we define a *block configuration* E as a set of Pauli errors ϵ and erasure errors ε on B. Given the check operators of the logical block (e.g., those of the 6-ring in Fig. 2), we can deduce the syndrome σ. The combined information of the syndrome and the erasure information is collectively called the *visible information* $v_E = (\sigma, \varepsilon)$. We let the space of all visible information for a given block be denoted V_B. Here, the logical block is the magic state preparation block, which has the parameters (L, L_d).

A postselection *rule* R observes the visible information v_E and decides whether to accept or reject the block with configuration E using a *soft-information function* Q followed by a *policy* P:

(1) A soft-information function $Q : V_B \to \mathbb{R}^q$ maps the visible information $v_E \in V_B$ to a vector of soft-information data $q_E \in \mathbb{R}^q$. This step distills useful and actionable information about B based on E.
(2) A policy $P : \mathbb{R}^q \to \{0, 1\}$ digests the soft information q_E and produces a decision on whether to accept (1) or reject (0) the block B based on the configuration E. In general, the policy can be any function of choice. Often, however, this is achieved by a *scoring function* $S : \mathbb{R}^q \to \mathbb{R}^+$ that maps q_E to a numerical score for the block, from which the binary decision is achieved by accepting blocks below a certain cutoff score, i.e., $P = \Theta(s^* - S(q_E))$, where Θ is the Heaviside function and s^* is a cutoff score such that all configurations with $S(q_E) \leq s^*$ are kept. Examples of both cases will be shown in Sec. IV A.

For brevity, we will often write $S(Q)$ to refer to the function $S \circ Q$ that returns a score for some visible information. If, on average, κ fraction of blocks are kept, then the postselection rule has an average resource overhead of $O := 1/\kappa$ times the overhead of creating a single block (for more details, see Sec. V E). The goal is to construct a rule such that the logical error rate (determined via decoding) on the subset of the κ accepted blocks is significantly less, on average, than that on all blocks. This occurs when R strongly correlates the policy output (often achieved through the score $S(Q)$) with the likelihood of logical error, thereby facilitating easy selection of less-error-prone blocks. Furthermore, a high-performing rule in practice would also have low overhead. A rule that is high performing in terms of error suppression but that requires large, potentially exponential, overhead is likely impractical beyond small block sizes.

A. Rules

We now define several rules, which we name the annular syndrome, the logical gap, the nested logical gap, and the radial logical gap. We discuss an additional rule that we name the surviving distance in Appendix A. These rules can be applied to *any* block and configuration but here we tailor the rules toward the problem of magic state preparation, a natural setting for applying postselection techniques. We will further focus on FBQC with the 6-ring network for concreteness but the techniques readily generalize to other models and schemes.

In these definitions, and the simulations that follow, we make use of the notion of a *syndrome graph*. The syndrome graph is defined by placing a vertex for each check operator (bulk cubes and boundary checks) of the fusion network. We connect two vertices with an edge whenever the corresponding check operators utilize a common measurement outcome. For the 6-ring fusion network, there are two distinct syndrome graphs termed the primal or dual syndrome graphs, analogous to the planar surface code, with the bicolorability in Fig. 1 indicating checks belonging to the two independent syndrome graphs, i.e., neighboring vertices associated with blue (red) checks are connected with an edge, forming the primal (dual) syndrome graph. Furthermore, the magic state preparation block has only 2 logical membranes, one supported on the edges of each syndrome graph.

For more general logical blocks encoding channels from m to n qubits, there are $m + n$ independent logical membranes that generate all possible logical correlations from input to output (for more details, see Ref. [18]). We denote the set of independent logical membranes, which index 2^{m+n} logical sectors, by C.

1. Annular syndrome

The annular-syndrome rule $R_S = (Q_S, P_S)$ relies solely on syndrome information. It computes the weighted sum of the -1 ("lit-up") syndromes. We choose the weights according to a power-law decay from the preparation point. The intuition is that syndromes near the preparation point are more significant in predicting a logical error than those further in the bulk. As such, the soft-information function

maps to a vector of length 2 with the components

$$Q_{S,i}(v_E;\alpha) := \sum_{r=1}^{L_d} \frac{\sigma_i(r)}{\bar{\sigma}_i(r)\min(r,\lceil 3L/4\rceil)^\alpha}, \quad i = \text{primal, dual}, \tag{3}$$

where $\sigma_i(r)$ is defined as the total number of -1 syndrome outcomes at a distance of r from the initial magic state, termed an "annulus," with radius r, $\bar{\sigma}_i(r)$ is the total number of syndrome measurements (independent of outcome) in the same annulus of radius r, and α is a tunable parameter. One can choose any metric to define the radius; here, we use the L_∞ metric (also known as the supremum metric) on the fusion network (depicted in the bottom left of Fig. 1, where resource states reside on vertices of the cubic lattice) [60]. We apply a radial cutoff of $\lceil 3L/4\rceil$ to ensure that for large depth blocks where $L_d > L$, there is no tail region at large radius where syndromes are counted with almost no weight, i.e., there must be some minimum penalty for having syndromes. Note that for more general topological codes, one may not have a split primal and dual syndrome-graph structure and one can simply sum over all syndromes in a radius around the preparation point.

The policy is implemented by thresholding a score

$$S_S(Q_S) := \sum_{i=\text{primal, dual}} a_i Q_{S,i},$$
$$P_S(Q_S; s_S^*) := \Theta(s_S^* - S_S(Q_S)), \tag{4}$$

where the a_i are tunable linear weights to construct a combined score from the primal- and dual-graph annular syndromes.

2. Logical gap

The logical-gap rule $R_G = (Q_G, P_G)$ is inspired by the statistical-mechanical mapping of error correction in Ref. [6], whereby an error-correction threshold is equivalent to the phase transition in a related statistical-mechanical model. Above the threshold, logical errors are not suppressed due to a loss of distinguishability between distinct logical sectors. In other words, above the threshold, the decoder can no longer reliably differentiate which logical sector of the code space to recover to (as the code distance increases). In this spirit, one can define the logical gap as the difference between the correction weights that return the system to different logical sectors.

For example, in the simple case of a single logical $\bar{Z}$ operator in a surface-code memory block (e.g., only the primal syndrome graph), with a configuration E and possible corrections $\bar{l}_{\text{correct}}, \bar{l}_{\text{wrong}}$ such that composing the correction and error yields a logical operator on the code space—namely, $\bar{I}$ and $\bar{Z}$, respectively. The *signed logical gap* is defined as

$$\Delta_{\bar{Z}}(E) := w_{\bar{Z}}(\bar{l}_{\text{wrong}}) - w_{\bar{Z}}(\bar{l}_{\text{correct}}), \tag{5}$$

where $w_{\bar{Z}}(\bar{l})$ denotes the log-likelihood weight of the correction $\bar{l}$ for the $\bar{Z}$ sector given by a choice of decoder, defined as follows: an edge e has weight $w_e = \ln 1 - p_e/p_e$, where p_e is the (marginal) probability of Pauli error on that edge, edges $e \in \varepsilon$ supporting erasures have weight $w_e = 0$, and the total weight of a correction $\bar{l}$ is $w_{\bar{Z}}(\bar{l}) = \sum_{e\in\bar{l}} w_e$. In reality, the error ϵ as part of E is unknown and therefore which correction is correct is unknown; hence we only have access to the *unsigned* logical gap, which we refer to simply as the *logical gap* $|\Delta_{\bar{Z}}(E)|$ (below, we will drop the dependence on E for brevity).

In general, any decoder can be used to compute a logical gap and biased noise can be accommodated by modifying the weights appropriately. If one chooses a minimum-weight perfect-matching (MWPM) decoder, then the decoder will always choose the minimum weight correction. If $\Delta_{\bar{Z}} < 0$, the decoder will fail and a logical error will be introduced. If $\Delta_{\bar{Z}} > 0$, the decoder will succeed in correcting the error and if $\Delta_{\bar{Z}} = 0$, the decoder will equally succeed or fail half of the time. Therefore, the EER for the block becomes

$$p_{\text{enc}} = \sum_{i\in(\bar{Z},\bar{X})} \int_{-\infty}^{0} P(\Delta_i) d\Delta_i, \tag{6}$$

where $P(\Delta_i)$ is the distribution of logical gaps of logical membranes i for a fixed block size and error rate. In more complex *logical blocks* [18] (i.e., surface-code protocols or channels), there will be many logical membranes and so one can compute a vector of logical gaps as the soft information of interest:

$$Q_{G,i}(v_E) := |\Delta_i|, i \in C, \tag{7}$$

where, recall, C is the set of distinct logical membranes. We can create a combined score for the block to be thresholded by the policy as

$$S_G(Q_G) := \sum_{i\in C} a_i e^{-Q_{G,i}},$$
$$P_G(Q_G; s_G^*) := \Theta(s_G^* - S_G(Q_G)), \tag{8}$$

where the a_i represent tunable linear weights to add the scores of all logical membranes. We explain how to compute the logical gap in Sec. V B.

3. Nested logical gap

The nested-logical-gap rule $R_N = (Q_N, P_N)$ is a derivative of the logical-gap rule that combines information of

the annular syndrome as the soft information of interest:

$$Q_N(v_E;\alpha) := (Q_G(v_E), Q_S(v_E;\alpha)). \tag{9}$$

The policy is given by conditional thresholding, expressed as

$$P_N(Q_N; s_G^*, s_S^*) = \begin{cases} 1, & \text{if } S_G(Q_G) = s_G^* \text{ and} \\ & S_S(Q_S) \le s_S^*, \text{ or} \\ & S_G(Q_G) < s_G^*, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

If one were to imagine a scenario of choosing M out of N configurations, then this policy amounts to sorting all N configurations first by the logical gap and then by annular syndromes, choosing the best M configurations in order. The intuition is that the preparation block has constant distance of 2 to flip logical sectors and so up to normalization, the gap, for a single graph (primal or dual), is bounded to $|\Delta| \le 2$, thus leading to a large degeneracy, as we will see in Sec. V C. The idea is to use the annular-syndrome rule to break this degeneracy.

4. Radial logical gap

The radial-logical-gap rule $R_{\mathrm{RG}} = (Q_{\mathrm{RG}}, P_{\mathrm{RG}})$ is a derivative of the logical-gap rule that caters specifically to the structure of the preparation block. The radial-logical-gap rule computes the logical gap but with a radial power-law (similar to the annular-syndrome rule with same cutoff) reweighting of the edge weights such that $\tilde{w}_i := w_i/\min(r, \lceil 3L_d/4 \rceil)^\alpha$. This yields $\tilde{\Delta}_i := \tilde{w}_i(\bar{l}_{\mathrm{wrong}}) - \tilde{w}_i(\bar{l}_{\mathrm{correct}})$ and

$$Q_{\mathrm{RG},i}(v_E) := |\tilde{\Delta}_i|, i \in C. \tag{11}$$

We can create a combined score for the block to be thresholded by the policy as

$$S_{\mathrm{RG}}(Q_{\mathrm{RG}}) := \sum_{i \in C} a_i e^{-Q_{\mathrm{RG},i}}$$
$$P_{\mathrm{RG}}(Q_{\mathrm{RG}}; s_{\mathrm{RG}}^*) := \Theta(s_{\mathrm{RG}}^* - S_{\mathrm{RG}}(Q_{\mathrm{RG}})). \tag{12}$$

The intuition here is to break the aforementioned degeneracy of the logical gap in the preparation block by biasing the decoder to compute corrections *away* from the preparation point (and into the bulk) so that some entropic contributions are, in a heuristic manner, included.

V. RESULTS AND DISCUSSION

A. Simulation details

We consider the magic state preparation block and error model described in Sec. III. Specifically, we consider IID bit-flip and erasure errors with strength p_{error} and p_{erasure} on the XX and ZZ fusion outcomes, as well as single-qubit X- and Z-measurement outcomes. We Monte Carlo sample $n_{\mathrm{trials}} = 10^5$ trials of preparation block configurations, apply each rule to all samples, and selectively keep the best κ fraction of them. We assess the EER p_{enc} [recall Eq. (2)] of each rule as a function of κ. On a real quantum computer, any desired κ can be achieved, on average, by running the policy in real time with appropriate choice of score cutoffs. The MWPM decoder [6,61] is used for both decoding and computing the logical gap (as described below in Sec. V B). Surviving-distance computations can be performed using Djikstra's shortest-path algorithms, modified to keep track of multiplicities. Unless otherwise stated, all linear weights $\{a_i\}$ for all rules are set to unity in the spirit of being fully agnostic between primal and dual graphs.

We primarily consider a block size of $L = 8$ for our analysis, as input magic states of this distance can be distilled using two rounds of the (optimized) 15-to-1 distillation protocol of Ref. [41], resulting in an output encoded magic state of $d = 32$ and a logical error rate that is sufficient for many interesting applications [1–3,18,45]. However, we do show results for $L = 4$ in Appendix B and we observe that qualitatively, the results do not depend on L and can be used for larger distances as desired.

We assume on the output port that all surface-code stabilizers are measured noiselessly, allowing for a logical readout in each basis, i.e., while the output qubits themselves are subject to noise, we assume no measurement noise on the stabilizer measurements (for more details, see Ref. [18]).

B. Computing correction weights for the logical gap

To compute the logical gap (or surviving distance) for surface-code protocols, we use the MWPM decoder. In the syndrome-graph representation, on each boundary there are "rough edges" corresponding to errors that only flip a single check generator (and should be thought of as locations where error chains can end without detection). We attach all of these rough edges along each boundary to an additional *pseudosyndrome* vertex. Pairs of pseudosyndromes on opposite boundaries are associated with a logical error class. For the purposes of decoding, we are free to choose whether each pseudosyndrome is flipped, subject to the constraint that pairs of pseudosyndromes must have a parity equal to that of the bulk syndrome graph. For a given parity, by setting the two different choices of flipped pseudosyndrome and running MWPM on each, we can obtain the minimum-weight recoveries for each sector (i.e., the two distinct logical corrections for each syndrome graph) which can then be used to compute the gap as defined in Sec. IV A. For an illustration of the syndrome graph, see Fig. 5 in Appendix B 1.

C. Rule performance

We show the performance of all rules [62] in Fig. 3(a), as a function of the keep fraction κ for pure Pauli error $p_{\text{error}} = 0.6p^*_{\text{error}}$ and $p_{\text{erasure}} = 0$, where $p^*_{\text{error}} = 0.0108$ is the bulk threshold of the memory block. In other words, we show $p^R_{\text{enc}}(\kappa; p_{\text{error}} = 0.6p^*_{\text{error}}, p_{\text{erasure}} = 0)$ for $R \in \{R_S, R_G, R_N, R_{\text{RG}}\}$. At $\kappa = 1$, there is no postselection and hence all rules have the same EER, i.e., the same p_{enc}. As κ decreases and fewer blocks are accepted, all rules suppress the EER, albeit at different rates. The overhead O for postselection is equal to $1/\kappa$. If one assumes the same error rate on the initial magic state such that $p_{\text{init}} = p_{\text{error}}$, then the intersection of the EER of each rule with the magenta line of Fig. 3 indicates the "break-even" keep value $\kappa^*(R)$ (or overhead $O^* = 1/\kappa^*$) at which the EER is the equal to the initial magic state error. As per Eq. (2), postselection yields diminishing returns for $\kappa < \kappa^*(R)$ as the overall error rate p_{prep} becomes dominated by p_{init} in this regime.

For $p_{\text{error}} < p^*_{\text{error}}$, as in Fig. 3(a), the EER suppression is superexponential (in κ) for the gap rule and its variants, with an overhead of $O^* \lesssim 2$, below which there are diminishing returns, as the initial magic state error p_{init} will become the dominant source of error. When the differential overhead cost is low, i.e., $d \ln p^R_{\text{enc}}/d\kappa \gg 0$ in the regime around κ^*, it might be desirable to use (relatively small) extra overhead to suppress the EER further below the initial magic state error. In contrast, in the same regime, the annular-syndrome rule R_S has poor suppression of the EER since the syndrome fraction is only loosely correlated with the EER. As shown in Appendix B 1 a, as L increases, the annular-syndrome rule performs increasingly poorly since statistical fluctuations of obtaining finite-size samples with few syndromes are exponentially suppressed [cf. Fig. 3(b)]. However, for larger L, the gap rules all still perform well as the gap is effectively utilizing a decoder rather than being reliant on statistical fluctuations at finite size. For $p_{\text{error}} \approx p^*_{\text{error}}$ (at the bulk threshold value), as shown in Appendix B 1 a, interestingly the same qualitative behavior holds but quantitatively the EER suppression rate is reduced as expected when the system is inherently more noisy.

To understand the performance differences between the various gap rules, it is instructive to analyze the distributions of scores and the correlations of the scores with their respective EERs. From Fig. 3(c), the distribution of gap

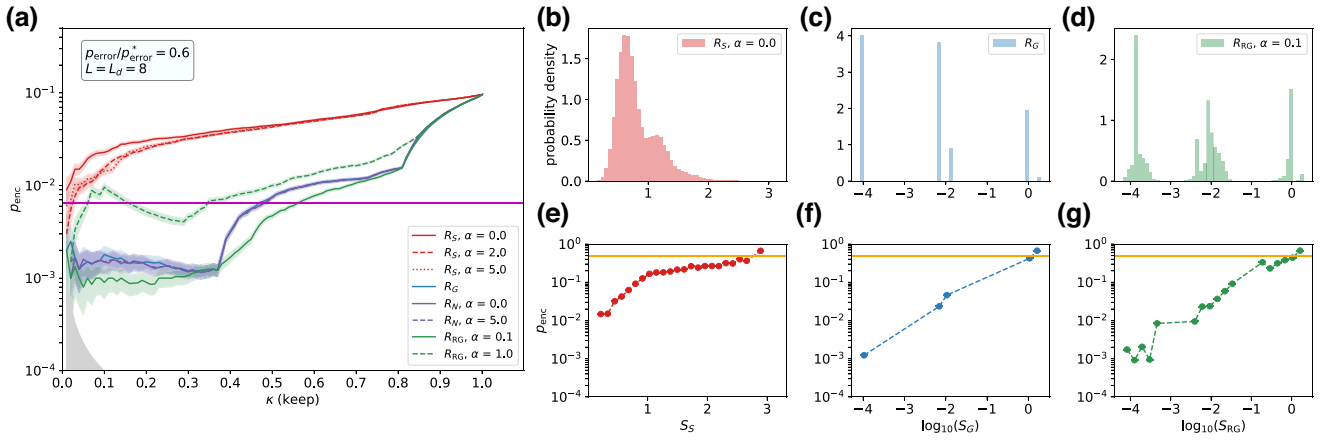

FIG. 3. (a) The encoding error rate (EER, p_{enc}) of a $L = L_d = 8$ cubic magic state preparation block at $p_{\text{error}} = 0.6p^*_{\text{error}}$, where p^*_{error} is the bulk threshold, for the annular-syndrome (red), logical-gap (blue), nested-logical-gap (purple), and radial-logical-gap (green) rules. The logical-gap-based rules give rapid suppression [steplike due to the discrete logical-gap sectors seen in (c)] in the EER as compared the annular-syndrome rule, with the radial-gap rule at low power $\alpha = 0.1$ performing the best. The magenta line indicates the "break-even" line, where $p_{\text{enc}} = p_{\text{init}} = p_{\text{error}}$, whereafter diminishing returns occur with decreasing κ as per Eq. (2). The gray region, where $p_{\text{enc}} \leq 1/n_{\text{trials}}\kappa$, indicates the limits of sampling in the simulation. The shading around the colored lines denotes the standard error $(p_{\text{enc}}(1 - p_{\text{enc}})/(n_{\text{trials}}\kappa))^{1/2}$. (b)–(d) Distributions of scores for the (b) annular-syndrome (S_S), (c) logical-gap (S_G), and (d) radial-logical-gap (S_{RG}) rules, respectively. Annular-syndrome scores have a continuous distribution, whereas the logical-gap sectors are discrete. The radial-logical-gap score at $\alpha = 0.1$ weakly breaks the degeneracy of the logical-gap scores. (e)–(g) The correlation of EER and score for the (e) annular-syndrome, (f) logical-gap, and (g) radial-logical-gap rules, respectively. The postselection policy function prefers configurations with smaller scores. As can be seen in all of the plots, configurations with smaller scores have lower EERs on average. The orange line is at $p_{\text{enc}} = 0.5$, indicating the absence of correlation (uniform probability of either logical sector). The annular-syndrome score has a poor correlation with EER while the logical-gap and radial-gap rules have a strong exponential correlation with EER, thereby endowing the logical gap with predictive power. The radial-logical-gap rule, at $\alpha = 0.1$, has a more continuous distribution and smoother correlation with the EER, thus improving upon the logical-gap rule by smoothing out the "steplike" features in (a).

rule scores $S_G(Q_G)$ is highly degenerate due to the fact the gap for each logical membrane is bounded by the constant fault distance of the preparation block ($|\Delta_{\text{primal}}|, |\Delta_{\text{dual}}| \in \{0, 1, 2\}$) and hence the combined score can only take on five distinct values, $S_G(Q_G) \in \{0, 1, 2, 3, 4\}, \forall E$, up to normalization. As p_{error} increases, the distribution of gaps for concentrates around $\Delta = 0$. This results in an overall shift of the gap score distribution toward higher values, indicating that the configurations are typically more error prone (due to decreased distinguishability between logical sectors—for more details, see Appendix B 1 b). From Fig. 3(f), a decreasing gap score has an exponentially smaller EER and this strong correlation of the score and EER gives rise to the predictive power of the gap rule. Furthermore, the fact that the gap rule does not have an exponentially vanishing number of configurations at low scores is what also makes it practical with low overhead.

In the hope of improving upon the gap rule by breaking the degeneracy of the gap sectors to yield a more fine-grained score, one can nest the annular-syndrome score inside each of the discrete gap sectors and assess the performance of this nested rule R_N. This does provide minor improvements over certain ranges of κ as compared to the gap rule [Figs. 3(a) and 6] but is not particularly remarkable due to the poor correlation of syndromes with EER [Fig. 3(e)] persisting inside each gap sector. In contrast, modifying the gap by adding an inverse radial weighting in R_{RG} breaks the degeneracy of the gap scores [Fig. 3(d)] by favoring corrections—that determine the gap—away from the initial magic state (where the fault distance is constant). For low power $\alpha = 0.1$, the radial-gap score $S_{\text{RG}}(Q_{\text{RG}})$ weakly breaks the degeneracy of the gap score while still preserving the gap sectors. This heuristically incorporates more entropic effects in the bulk, providing a smoother graded correlation of the radial-gap score and the EER [Fig. 3(g)], thus leading to improved predictive power of R_{RG} over R_G. We find that R_{RG} is at least as good if not better than R_G for all κ. At higher α, e.g., $\alpha = 1.0$ as in Fig. 3(a), the radial-gap-rule performance degrades due to the now strong power law that mixes gap sectors, thus obtaining a poor nonmonotonic correlation with the EER (see Appendix B 1 c) and hence losing the original predictive power of the gap.

To compare the performance of the rules over a range of Pauli-error rates, in Fig. 4(a) we show the break-even overhead as a function of the fraction of the bulk threshold $p_{\text{error}}/p^*_{\text{error}}$. Over this entire range, the radial-gap rule at low α has the lowest overhead required to reach the break-even point, everywhere performing better than the gap and nested gap. In contrast, the annular-syndrome rule performs poorly. At an error rate of $p_{\text{error}} = 0.6p^*_{\text{error}}$, the radial-gap rule has a relative overhead of only 1.78, which is approximately 23 *times* lower than the best annular-syndrome rule and 1.17 *times* lower than the gap rule. As the error rate increases, the annular-syndrome break-even overhead increases exponentially due to its reliance on statistical fluctuations (e.g., configurations

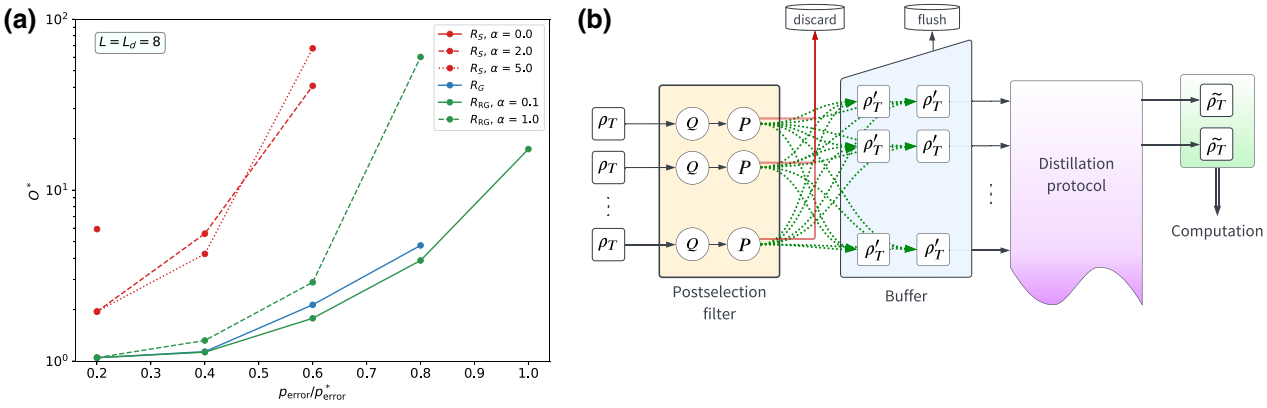

FIG. 4. (a) The break-even overhead $O^* = 1/\kappa^*$ as a function of the fraction of the bulk threshold $p_{\text{error}}/p^*_{\text{error}}$ (where p^*_{error} is the bulk threshold) at $L = L_d = 8$ for different postselection rules. The annular-syndrome rule has relatively poor performance, with large overhead needed (as a function of error rate *and* L, low overhead is desired) to achieve the break-even point, suggesting that simple syndrome-counting based rules are inefficient. In stark contrast, the radial-gap rule (green), at $\alpha = 0.1$, outperforms the logical-gap (blue) and annular-syndrome (red) rules by having the lowest overhead to achieve the break-even condition $p_{\text{enc}} = p_{\text{init}} = p_{\text{error}}$ over the entire error range. In particular, the radial-gap rule has a benign scaling for error rates below the bulk threshold with, e.g., only a modest approximately 1.78 overhead at $0.6p^*_{\text{error}}$ to reach the break-even point. The nested-gap rule has similar performance to the gap rule as seen and is therefore not shown for clarity. (b) Magic states are prepared, postselected, and then stored in a buffer for distillation. In the figure, the initial encoded magic states ρ_T are prepared and postselected upon using the soft-information function Q and policy P. Magic states are either discarded (if rejected by the policy) or accepted and if accepted they are sent to the buffer. Accepted magic states are denoted ρ'_T and have error rate p_{prep} and are utilized in the distillation protocol. The output states of the distillation protocol are denoted $\tilde{\rho}_T$ and are used for fault-tolerant quantum computation.

with zero syndromes are desirable but are exponentially rare) and quickly surpasses tractable simulation; hence the absence of break-even points at higher errors in Fig. 4(a). This is similarly seen, even for the gap rule with an absence of a break-even point in the current simulation at $p_{\text{error}} = p^*_{\text{error}}$. On the flip side, however, even at the bulk threshold error rate, the radial-gap rule can still achieve the break-even condition at a reasonable overhead of approximately 17. It is important to note that for a given rule, there may not always be a break-even point, even with infinite simulation capacity, since above the optimal decoding threshold (in the thermodynamic limit), the visible information cannot be used to reliably distinguish logical sectors. These qualitative results hold true even in the presence of nonzero erasure, as shown in Appendix B 2.

D. Comparison to prior work

Here, we compare our results with that of Li [47] and Singh *et al.* [48]. We remark that a direct comparison with their results is difficult due to differences in error models and intended purpose. Both prior works design their protocols to exploit structure in the noise, either targeting noise models dominated by two-qubit errors or noise models with a high bias between X and Z errors, respectively. Such error models are motivated by, e.g., certain superconducting architectures. In contrast, we focus on generic error-model-independent strategies and present results on a nontailored IID noise model. While our simulations are based on an unstructured 6-ring fusion-based error model, we normalize the error strength to the bulk threshold to allow for the transferability of our results to different architectures.

The annular-syndrome rule is the one closest in spirit to the syndrome-based postselection rules of Li [47] and Singh *et al.* [48]. As described in detail in Sec. V C, variants of the gap rule significantly outperform the annular-syndrome rule in terms of error-rate suppression across a range of physical error rates, requiring only modest overhead to do so. Furthermore, the gap-based rules allow us to output low-error encoded magic states even at high physical error rates (relative to the bulk threshold). For example, even with error rates close to the bulk threshold, the gap-based rules can reach the break-even point with a tractable overhead. In contrast, for the same situation, the annular-syndrome rule requires physically unrealistic overheads (see Figs. 4 and 6).

E. Architectural design

In reality, the input magic state preparation blocks must be selected in real time from a finite set. Further, several magic states are required for each round of distillation and so one must determine how many parallel preparation sites—called *preparation factories*—are required such that there is a sufficient rate of initial magic states reaching the first level of distillation. We propose and analyze a simple buffer-based architecture to obtain a more accurate estimate of the cost and performance of the postselection rules proposed in the previous sections. This buffer architecture is particularly well suited to photonic FBQC architectures but is applicable to matter-based CBQC architectures, provided that the routing costs are accounted for.

Consider n_{fac} preparation factories, each of which synchronously generates a magic state block on a clock with time interval t_{fac}. Consider also a collective memory buffer that can store a number of magic state blocks for a time $t_{\text{flush}} = n_{\text{cycles}} t_{\text{fac}}$, measured in the number of factory clock cycles n_{cycles}, before the entire buffer, i.e., *all* of its magic states, is erased. For a distillation protocol that takes in m_{in} blocks and outputs m_{out} blocks, if the buffer is not filled with m_{in} magic state blocks by t_{flush}, distillation cannot proceed, leading to wasted resources when the buffer is flushed. We assume that the temporal overhead for the classical computation needed for postselection is negligible in between the factories and the collective buffer, that there is all-to-all connectivity between factories and memory slots in the buffer as shown in Fig. 4(b), that routing magic states is free (in space and time), as in Ref. [57], and that all magic state factories are uncorrelated in terms of quality of initial magic states.

For a given postselection rule R, each keep ratio κ corresponds to a cutoff score(s) that we more explicitly denote $s^*(\kappa; R)$ (for score $S_R(Q)$), determined by numerical simulation *a priori*. At each factory clock cycle, n_{fac} magic state blocks are produced. A classical computational filter then applies the policy of the rule on each block, only accepting a block if $S_R(Q) \leq s^*(\kappa; R)$. The accepted blocks are moved into the buffer and since the probability of accepting a single magic state block is by construction κ, there are on average κn_{fac} blocks stored in the buffer after one clock cycle. Since the magic states produced in each clock cycle are uncorrelated with those produced in previous cycles, the collection of accepted magic state blocks after n_{cycles} follows a binomial distribution with mean $\mu = n_{\text{cycles}} n_{\text{fac}} \kappa$ and variance $\sigma^2 = n_{\text{cycles}} n_{\text{fac}} \kappa (1 - \kappa)$. To ensure a filled buffer of size m_{in} up to failure probability p_{flush} for an acceptance probability κ, we solve $p_{\text{flush}} = F(m_{\text{in}} - 1; n_{\text{cycles}} n_{\text{fac}}, \kappa)$ for $n_{\text{cycles}} n_{\text{fac}}$, where $F(x; n, p)$ is the cumulative distribution function (cdf) for the binomial distribution of n trials and success probability p. Note that p_{flush} rapidly decays in the regime of interest where $m_{\text{in}} - 1 < \mu$ and that the solution allows for a simple space-time trade-off between n_{cycles} and n_{fac} (since the product remains fixed), which is useful for working around any physical-resource constraints that might be present. Furthermore, $\sigma/\mu \sim 1/\sqrt{n_{\text{cycles}} n_{\text{fac}}}$ means that the relative fluctuations of buffer filling vanish with larger magic state requirements, as would be the case for multiple rounds of distillation. Practical implementation of this collective buffer scheme with a common flush time, or a

variation that allows for individual flush times rather than a collective flush time, both require a detailed specification of a physical architecture and its description of errors that will, e.g., inform constraints on total overhead and constraints on space-time geometries for routing magic state blocks.

A distillation protocol produces magic states with output error rate $f(p_{\text{prep}}; c, k) = cp_{\text{prep}}^k$ (to first order), for some constants c and k, assuming that the input magic state error rate p_{prep} is sufficiently small. For example, the well-known 15-to-1 distillation protocol [34,41] outputs 1 magic state of quality arbitrarily close to $35p_{\text{prep}}^3$ using 15 input magic states of quality p_{prep}. This assumes that the code distances for the input surface codes are large such that errors in the Clifford operations are negligible. For an algorithm of interest, with n_T T gates and n_Q qubits, one must distill magic states of a error rate $p_{\text{alg}} = O(1/n_T n_Q)$ to run the entire algorithm with constant error rate. To achieve this, one must choose the distillation protocol such that $f(p_{\text{prep}}; c, k) < p_{\text{alg}}$.

We would like to choose the distillation protocol that achieves this output error rate with the fewest resource states possible. One should jointly optimize the distillation protocol (across the landscape of possible distillation protocols [37,39,63]) and postselection protocol (i.e., the postselection rule, how many preparation factories are required, the preparation postselection rate, and the distance of the protocol) to minimize overall resources. As we have seen, the radial-gap rule achieves the lowest error rates to prepare magic states for a given postselection overhead (i.e., fixed O).

VI. CONCLUSIONS AND FUTURE WORK

As fault-tolerant demonstrations on current quantum technologies are becoming more prevalent [64–68], it is essential to develop more accurate modeling and resource-estimation tools to determine the requirements for large-scale quantum computations. We have established a framework for fault-tolerant postselection and applied it to the magic state preparation problem, a dominant source of overhead for fault-tolerant quantum computations. Our numerical results demonstrate that the postselection rules that we propose rapidly suppress the EER of initial magic states in surface-code blocks—under an error model of IID Pauli errors and erasure errors and over a wide range of error rates—to the level of the initial magic state error, all for low constant multiplicative overhead of approximately 1.5–5 times the cost of a single magic state preparation block. In particular, the logical gap—a postselection rule inspired by the statistical mechanics-to-quantum error-correction correspondence—and its variants serve as a powerful soft-information metric at the topological level. We observe up to an approximately 25 times reduction of overhead compared to commonly used syndrome-based postselection strategies, for practical operational regimes.

The proposed gap-based postselection protocols are general and can be applied to a variety of different fault-tolerant primitives, e.g., including the modified preparation protocols of Ref. [69]. Furthermore, while we use an implementation of MWPM to compute the gap, it may be computed with other more time-efficient decoding algorithms (e.g., UnionFind [70]). Recent advances in matching decoders (of which MWPM is the prototype) such as the SPARSE BLOSSOM [71] and FUSION BLOSSOM [72] decoders, along with other parallelized approaches [73], are significantly faster and may be used for practical implementations of our work. This may be an important consideration for practical implementations; Magic state preparation using syndrome-based postselection rules is now being explored in experiments [74] and our investigation shows that the logical gap could greatly improve the fidelity of the output, for the same overhead.

The gap-based soft information can inspire and serve as a foundation for postselection rules and/or multiplexing strategies for other logical blocks as part of the error requirements in a larger quantum architectural stack. As a future avenue, the framework introduced here can be viewed as a binary classification problem and such a problem may potentially admit a high-performing neural-network strategy, e.g., with one hidden layer representing the soft information. Specifically, the input to the neural network can include V_B, the error model parameters if known (e.g., p_{error}, p_{erasure}), and user-specified parameters

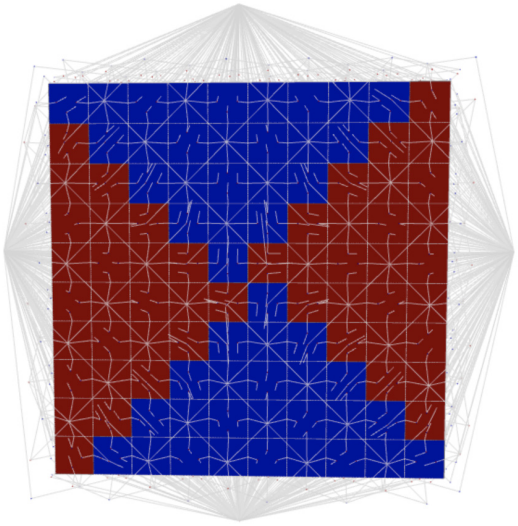

FIG. 5. A representation of the magic state preparation block (with $L = 10$) along with both primal and dual syndrome graphs used for the simulation. The pseudosyndromes used for the calculation of the gap are depicted on the top, bottom, left, and right.

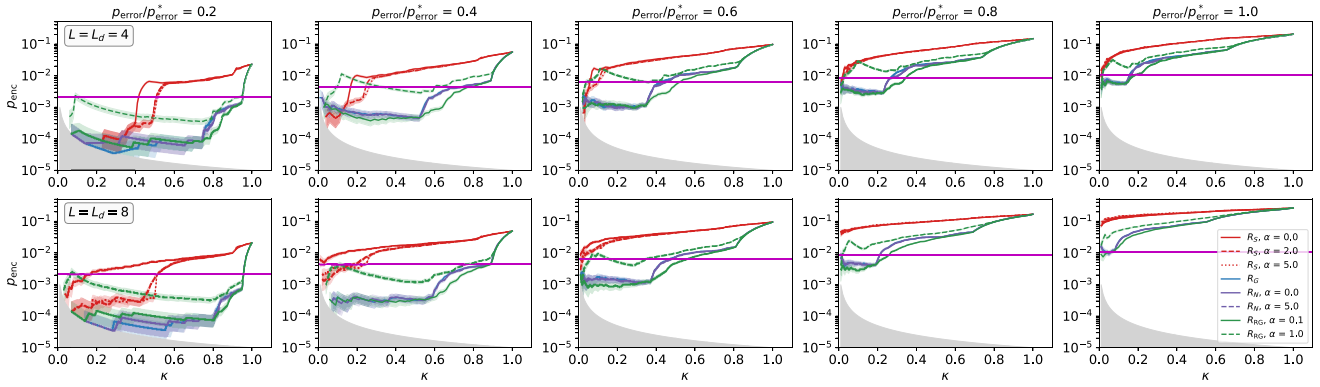

FIG. 6. The EER of a $L = L_d = 4, 8$ (first and second row, respectively) cubic magic state preparation block over a range of Pauli-error rates $p_{\text{error}}/p^*_{\text{error}} \in [0, 1]$ for the annular-syndrome, logical-gap, nested-logical-gap, and radial-logical-gap rules. The shading around the colored lines denotes the standard error $(p_{\text{enc}}(1 - p_{\text{enc}})/(n_{\text{trials}}\kappa))^{1/2}$.

such as the desired keep ratio. The output is simply the classification of whether to accept or reject the sample.

Further reduction of space-time volume might be possible by reducing the depth of the preparation block—e.g., preparation factories producing $(L, L_d) = (4, 2)$ for postselection might be sufficient for efficiently choosing quality blocks that can be routed into buffers, i.e., fused into large-depth-identity (memory) blocks. This

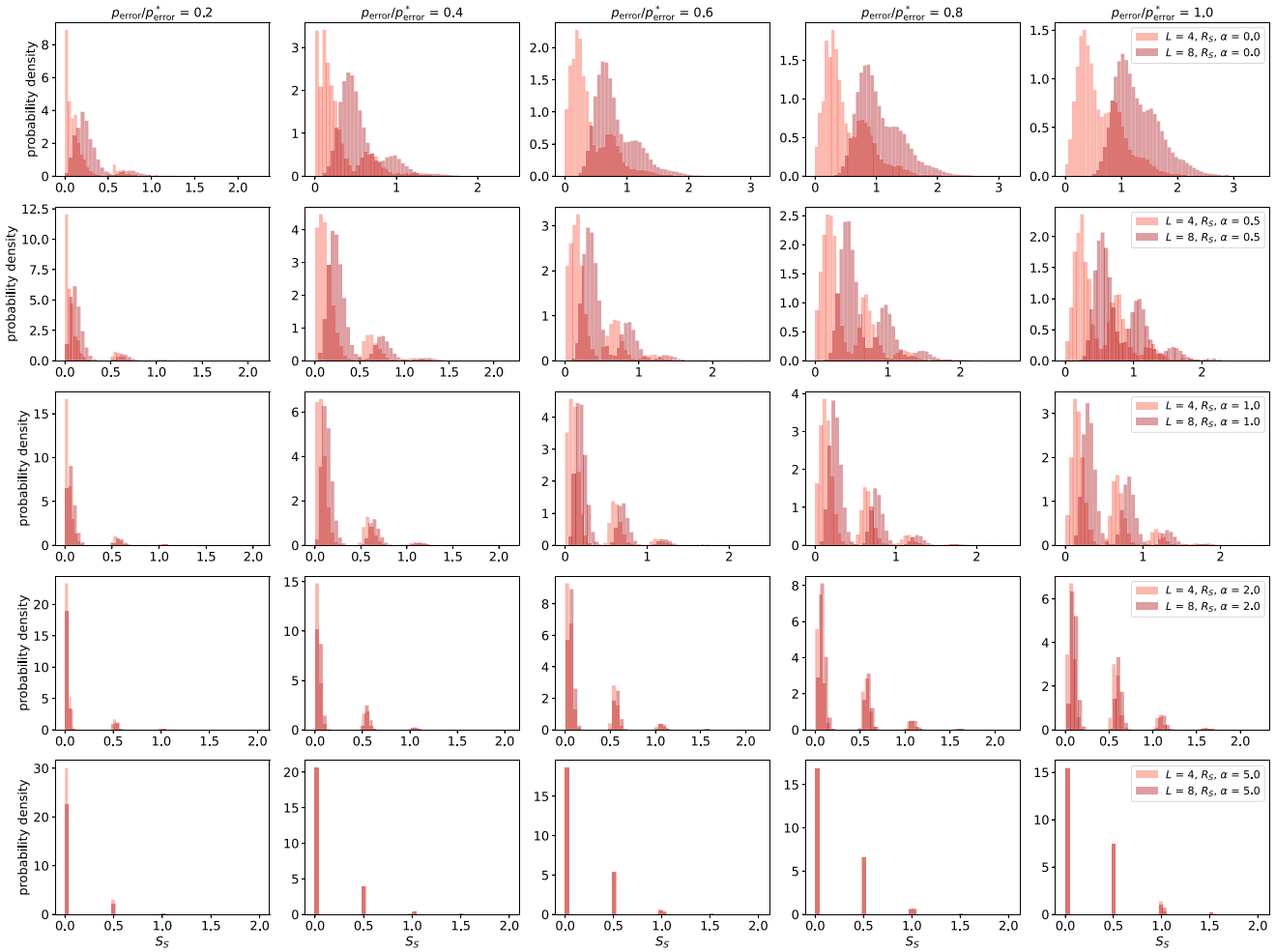

FIG. 7. The score distribution for the annular-syndrome rule. Each column has a fixed Pauli-error rate in the range $p_{\text{error}}/p^*_{\text{error}} \in [0, 1]$. Each row has a fixed choice of the rule parameter α, the power-law decay exponent of the radial weighting.

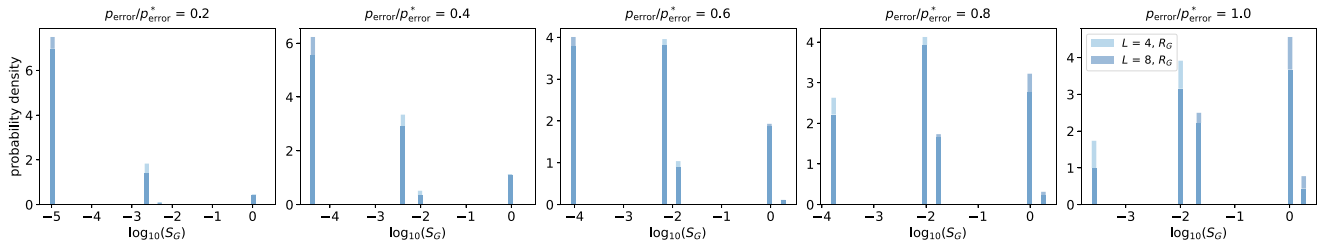

FIG. 8. The score distribution for the gap rule over Pauli-error rates in the range $p_{\text{error}}/p^*_{\text{error}} \in [0, 1]$.

short-depth situation is difficult to model as considered here (where postselection and decoding both take place on the full information of the block), since the final time-like boundary layer of perfect measurements in simulation becomes a large fraction of the block. For other logical blocks, fault-tolerant protocols, and error models (such as correlated or biased), the incorporation of more information about the noise model and geometry into the logical-gap computation may lead to further improvements.

ACKNOWLEDGMENTS

We thank Naomi Nickerson, Ye-Hua Liu, and Chris Dawson for detailed feedback during the course of the project, and Sara Bartolucci, Patrick Birchall, Hugo Cable, Axel Dahlberg, Andrew Doherty, Dan Dries, Megan Durney, Terry Farrelly, Mercedes Gimeno-Segovia, Eric Johnston, Konrad Kieling, Isaac Kim, Daniel Litinksi, Sam Morley-Short, Andrea Olivo, Sam Pallister, Fernando Pastawski, William Pol, Terry Rudolph, Jake Smith, Chris Sparrow, Mark Steudtner, Jordan Sullivan, David Tuckett, Andrzej Perez Veitia, and all our colleagues at PsiQuantum for useful discussions.

K.S., S.R., and M.P. developed the general fault-tolerant postselection framework. H.B. is responsible for the conception and initial investigation of the logical gap. K.S. and S.R. are responsible for initial investigations into FTPS of logical blocks, the further development and analysis of the logical gap, conception and development of its variants and other rules, and writing of the manuscript. K.S. is responsible for building the majority of the simulation.

APPENDIX A: SURVIVING-DISTANCE RULE

The surviving-distance rule $R_D = (Q_D, P_D)$ relies solely on erasure information and computes an analogue of the code distance of the block that remains after removing the erased clusters. Namely, we compute the length of the shortest path on the syndrome graph between two distinct boundaries, where erased edges have zero cost. In other words, the surviving distance is the minimal number of Pauli errors that can result in a logical error, given the

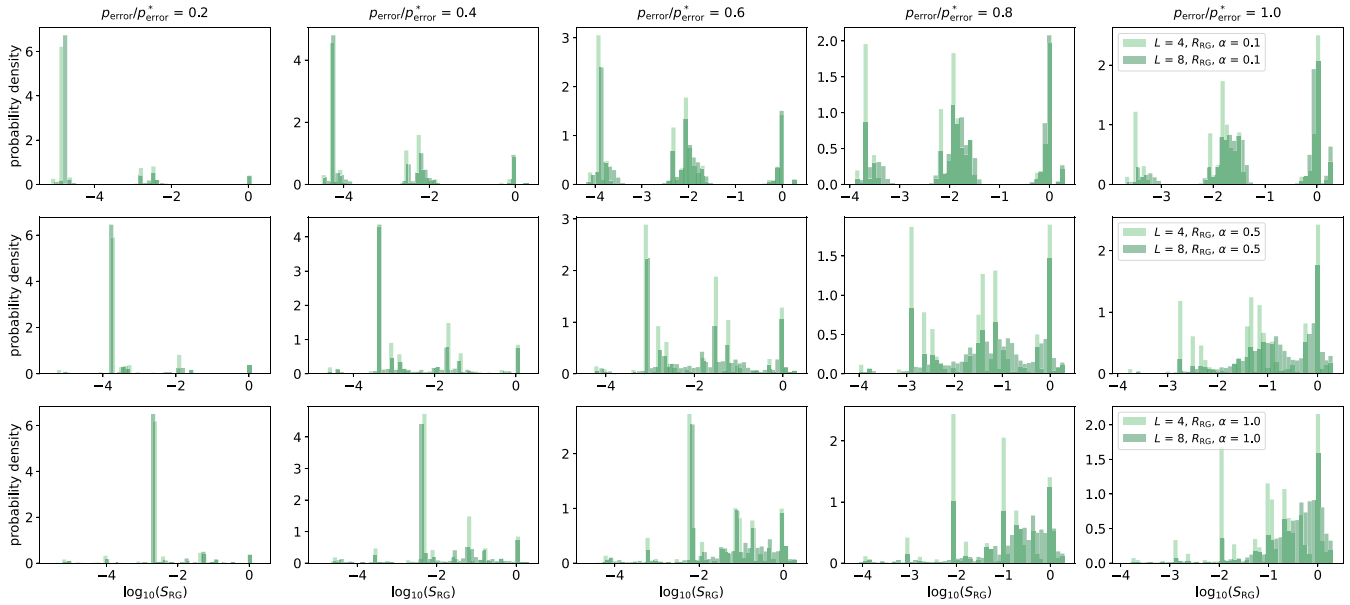

FIG. 9. The score distribution for the radial-gap rule. Each column has a fixed Pauli-error rate in the range $p_{\text{error}}/p^*_{\text{error}} \in [0, 1]$. Each row has a fixed choice of the rule parameter α, the power-law decay exponent of the radial weighting.

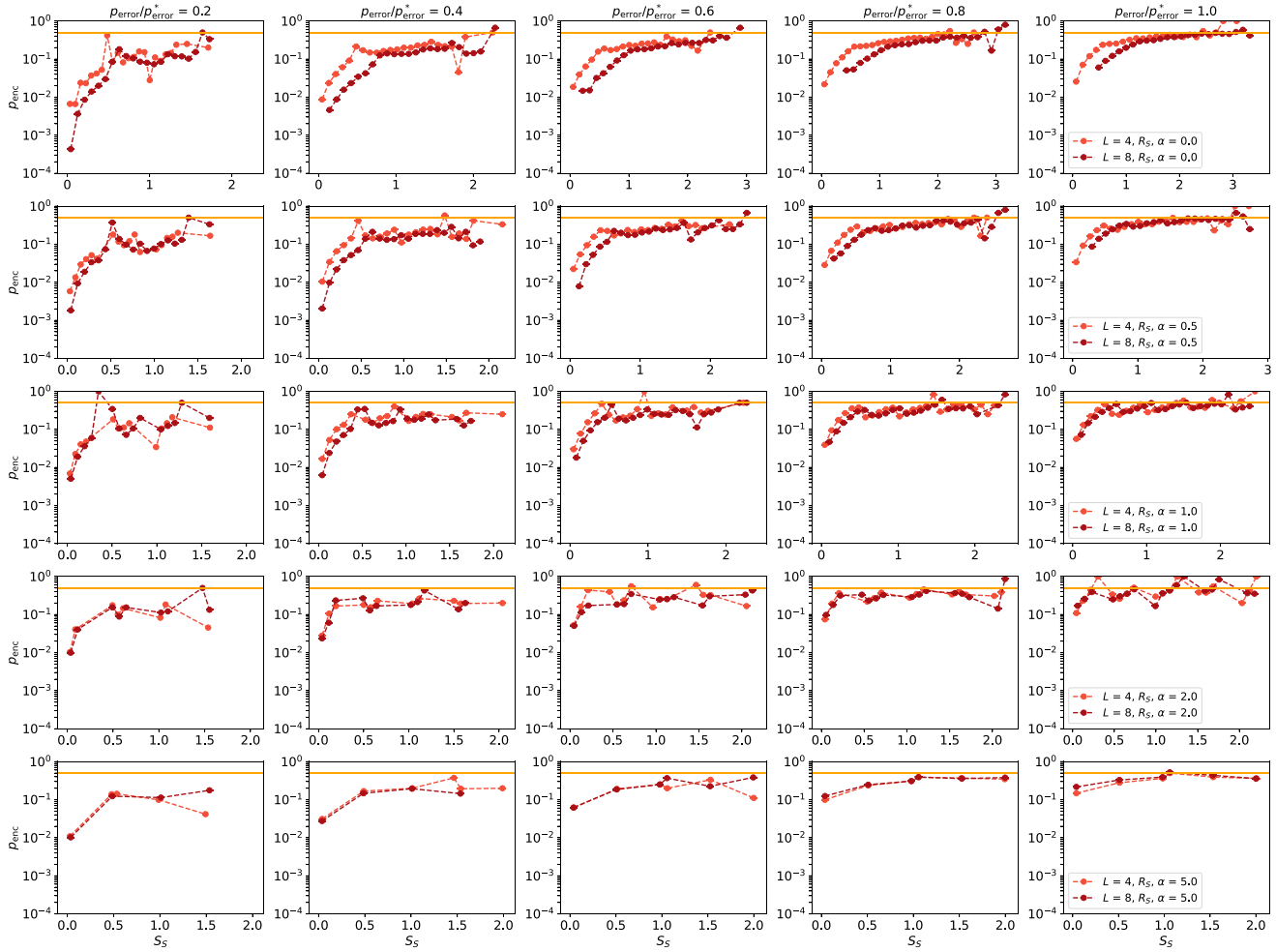

FIG. 10. The correlation of the EER and the score for the annular-syndrome rule. Each column has a fixed Pauli-error rate in the range $p_{\text{error}}/p^*_{\text{error}} \in [0, 1]$. Each row has a fixed choice of the rule parameter α, the power-law decay exponent of the radial weighting.

observed erasure. Furthermore, we may augment this rule by incorporating the multiplicity $m(d_i)$ of the shortest path into the scoring to compute an effective distance for each pair of boundaries as the soft information of interest:

$$Q_{D,i}(v_E; c) := d_i - c \ln m(d_i), i \in \text{boundary pairs}, \quad \text{(A1)}$$

where c is a tunable parameter that governs the weighting of the multiplicity. Each separated pair of boundaries (e.g., the two logical operators on a single surface-code memory block) contributes an effective distance and we can create a combined score for the block to be thresholded by the policy as

$$S_D(Q_D) := \sum_{i \in \text{boundary pairs}} a_i e^{-Q_{D,i}},$$
$$P_D(Q_D; s^*_D) := \Theta(s^*_D - S_D(Q_D)), \quad \text{(A2)}$$

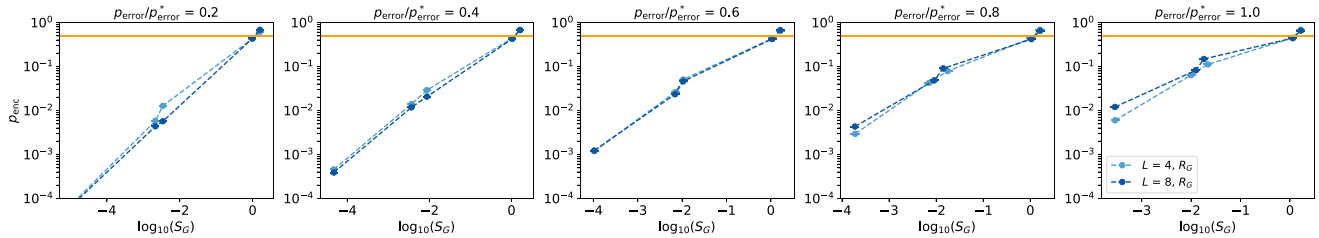

FIG. 11. The correlation of the EER and the score for the gap rule over range of Pauli-error rate in the range $p_{\text{error}}/p^*_{\text{error}} \in [0, 1]$.

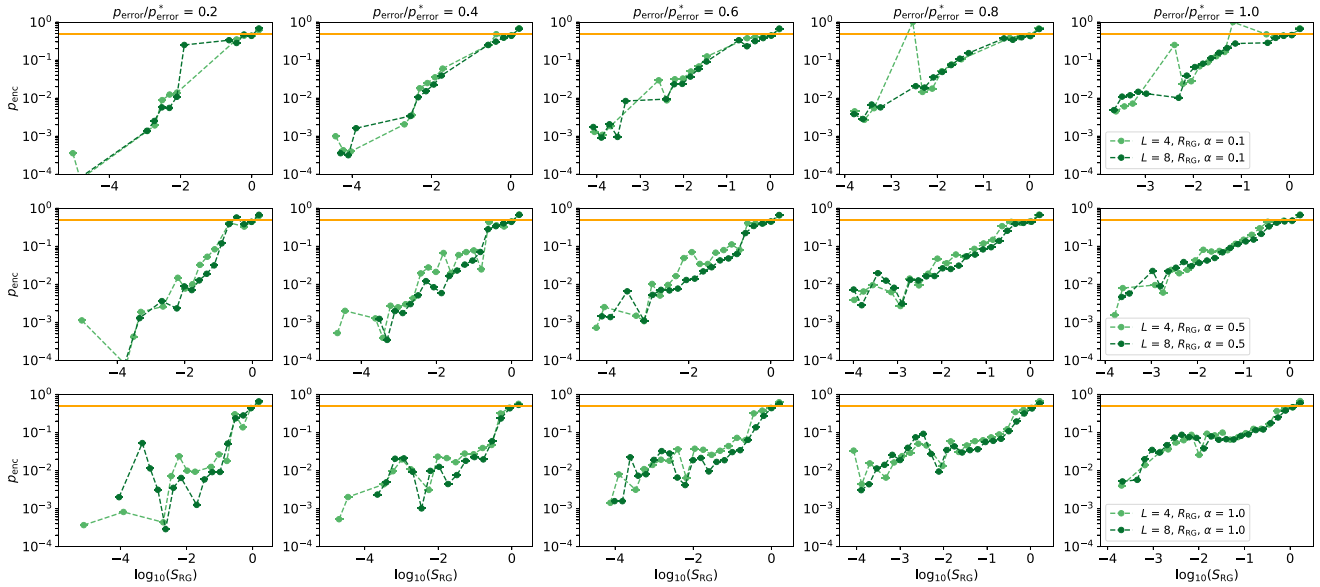

FIG. 12. The correlation of the EER and the score for the radial-gap rule. Each column has a fixed Pauli-error rate in the range $p_{\text{error}}/p^*_{\text{error}} \in [0, 1]$. Each row has a fixed choice of the rule parameter α, the power-law decay exponent of the radial weighting.

where the a_i are tunable linear weights to add the scores of different pairs of boundaries. The preparation block only has two pairs of boundaries terminating the $\bar{X}, \bar{Z}$ logical correlator and $i =$ primal, dual. The intuition is that spanning paths between boundaries mimic logical errors chains and so having smaller effective distances due to erasure makes the configuration less desirable.

APPENDIX B: DETAILED ERROR MODEL AND SIMULATION

We show extensive numerical results for the performance of the rules discussed in the main text over a range of Pauli error and erasure error as discussed in Sec. III C. In the case of pure Pauli errors, we also include the score distributions and the correlations of the EER and the scores for several different choices of rule parameters.

1. Simulation details

We consider the *hardware-agnostic fusion error model* of Ref. [49]. In this scheme, we assume that all errors arising during state preparation, idling, and fusions can be modeled by bit flips and erasures on the outcomes of the fusion. As such, we assume that each 6-ring cluster state is output with no errors and that each XX- and ZZ-fusion outcome is subject to a bit-flip error with rate p_{error} and an erasure with rate p_{erasure}. To perform simulations, we construct a syndrome graph for the protocol, sample bit flips and erasures on the edges of the syndrome graph, compute scores, decode, and measure whether a logical

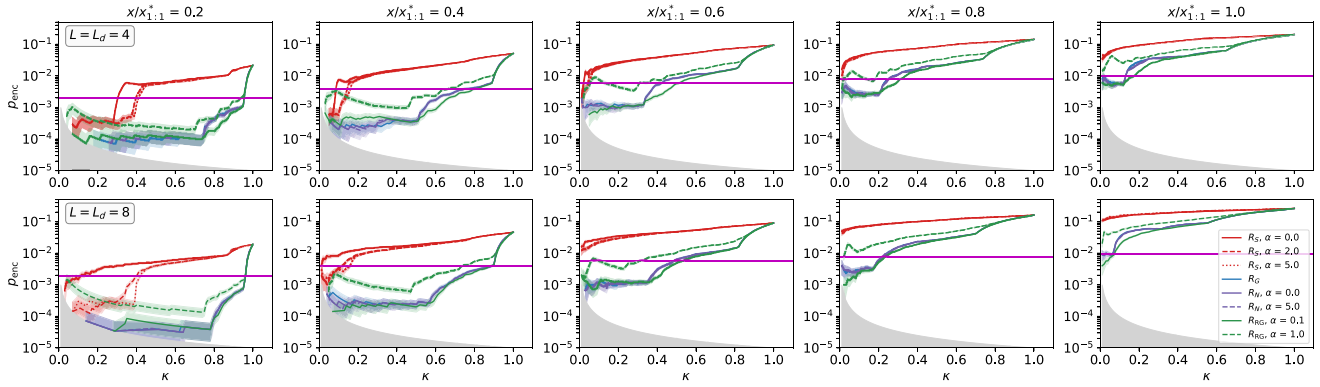

FIG. 13. The EER of a $L = L_d = 8$ cubic magic state preparation block over a mixed erasure and Pauli-error model with $(p_{\text{erasure}}, p_{\text{error}}) = (x, x)$, where $x^*_{1:1} = 9.71 \times 10^{-3}$ for the annular-syndrome, logical-gap, nested-logical-gap, and radial-logical-gap rules. The shading around the colored lines denotes the standard error $(p_{\text{enc}}(1 - p_{\text{enc}})/(n_{\text{trials}}\kappa))^{1/2}$.

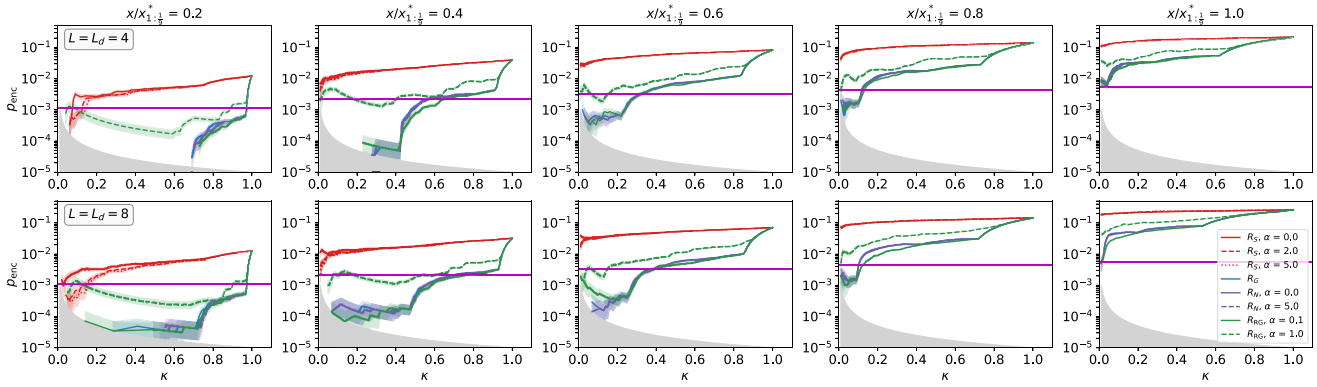

FIG. 14. The EER of a $L = L_d = 8$ cubic magic state preparation block over a mixed erasure and Pauli-error model with $(p_{\text{erasure}}, p_{\text{error}}) = (x, (x/9))$ where $x^*_{1:\frac{1}{9}} = 4.99 \times 10^{-2}$ for the annular-syndrome, logical-gap, nested-logical-gap, and radial-logical-gap rules. The shading around the colored lines denotes the standard error $(p_{\text{enc}}(1 - p_{\text{enc}})/(n_{\text{trials}}\kappa))^{1/2}$.

failure has occurred. For the gap-based scores, we append extra vertices to the syndrome graph called "pseudosyndromes," which, when flipped, allow for the computation of the minimal-weight recovery in each logical sector (as discussed in Sec. V B). These pseudosyndromes are shown in Fig. 5. After performing many such trials, one can estimate the logical error rate as a function of the physical error rate and acceptance rate (trials are ordered by their scores).

a. Rule performance

The EER as a function of the keep fraction is shown in Fig. 6 for select values of $p_{\text{error}}/p^*_{\text{error}} \in [0, 1]$. The first row shows results at $L = L_d = 4$ and the second row shows results at $L = L_d = 8$. Note that all the rules perform qualitatively similarly to that shown in the main text, with the expected degradation of performance as p_{error} approaches the bulk threshold p^*_{error}. It is interesting to note, that even at threshold, at the smaller size where finite-size effects are strong and beneficial, one can still hit the break-even point with a relative overhead of approximately 6 to 7.

b. Score distributions

In Figs. 7–9, we show the distributions of scores for the annular-syndrome, logical-gap, and radial-logical-gap rules, over a range of rule parameters (each row) and for $p_{\text{error}}/p^*_{\text{error}} \in [0, 1]$ (each column), respectively. Increasing the power-law exponent α in the annular-syndrome rule squeezes the distribution of scores, which apparently leads to performance improvement as seen in Fig. 6. As discussed in the main text, the gap rule has a discrete distribution over the full range of errors. In the radial-gap rule, a small power-law exponent α mildly breaks this degeneracy and spreads the gap sectors, while a large α mixes and reorganizes the gap sectors entirely, with the former yielding superior rule performance as seen in Fig. 6.

c. Correlation of EER and scores

In Figs. 10–12, we show the correlations of scores and the EER for the annular-syndrome, logical-gap, and radial-logical-gap rules, over a range of rule parameters (each row) and for $p_{\text{error}}/p^*_{\text{error}} \in [0, 1]$ (each column), respectively. As before, the correlation of annular-syndrome score and EER is weak, for all α, although for low values it is stronger and hence beneficial. Again, the gap and radial gap have a strong correlation of their respective scores and the EER, leading to significant performance gains as discussed in the main text, with the low α of the radial gap leading to a more monotonic and consistent correlation and thus yielding the best performance.

2. Mixed erasure and Pauli errors

We consider both erasure and Pauli errors. For this error model, fusion outcomes are erased with probability p_{erasure} and nonerased outcomes are further subject to a bit-flip outcome with rate p_{error}. We simulate two representative cases with $(p_{\text{erasure}}, p_{\text{error}}) = (x, x)$ and $(p_{\text{erasure}}, p_{\text{error}}) = (x, (x/9))$, where $x/x^* \in [0, 1]$ and x^* is the bulk threshold along the error ray parametrized by x. This threshold is determined empirically in both cases with MWPM decoding as $x^*_{1:1} = 9.71 \times 10^{-3}$ and $x^*_{1:1/9} = 4.99 \times 10^{-2}$, respectively. The results for the 1:1 case are shown in Fig. 13 and the results for the 1:1/9 case are shown in Fig. 14, the latter being a more physically relevant scenario for FBQC using photonics, where loss (which leads to fusion-outcome erasure) is a dominant source of error. In both cases, the qualitative behavior is similar to that of the pure Pauli-error case, demonstrating that the gap-based postselection rules yield significant improvement in the presence of erasures as well.

[1] C. Gidney and M. Ekerå, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits, Quantum **5**, 433 (2021).

[2] V. von Burg, G. Hao Low, T. Häner, D. S. Steiger, M. Reiher, M. Roetteler, and M. Troyer, Quantum computing enhanced computational catalysis, Phys. Rev. Res. **3**, 033055 (2021).

[3] S. Chakrabarti, R. Krishnakumar, G. Mazzola, N. Stamatopoulos, S. Woerner, and W. J. Zeng, A threshold for quantum advantage in derivative pricing, Quantum **5**, 463 (2021).

[4] A. Yu Kitaev, in *Quantum Communication, Computing, and Measurement* (Springer, 1997), p. 181.

[5] S. B. Bravyi and A. Yu Kitaev, Quantum codes on a lattice with boundary, arXiv preprint ArXiv:9811052 (1998).

[6] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, J. Math. Phys. **43**, 4452 (2002).

[7] A. Yu Kitaev, Fault-tolerant quantum computation by anyons, Ann. Phys. (NY) **303**, 2 (2003).

[8] A. Kitaev, Anyons in an exactly solved model and beyond, Ann. Phys. (NY) **321**, 2 (2006).

[9] H. Bombín and M. A. Martin-Delgado, Quantum measurements and gates by code deformation, J. Phys. A: Math. Theor. **42**, 095302 (2009).

[10] H. Bombín, Topological order with a twist: Ising anyons from an Abelian model, Phys. Rev. Lett. **105**, 030403 (2010).

[11] R. Raussendorf and J. Harrington, Fault-tolerant quantum computation with high threshold in two dimensions, Phys. Rev. Lett. **98**, 190504 (2007).

[12] R. Raussendorf, J. Harrington, and K. Goyal, Topological fault-tolerance in cluster state quantum computation, New J. Phys. **9**, 199 (2007).

[13] C. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter, Surface code quantum computing by lattice surgery, New J. Phys. **14**, 123011 (2012).

[14] M. B. Hastings and A. Geller, Reduced space-time and time costs using dislocation codes and arbitrary ancillas, arXiv preprint ArXiv:1408.3379 (2014).

[15] B. M. Terhal, Quantum error correction for quantum memories, Rev. Mod. Phys. **87**, 307 (2015).

[16] B. J. Brown, K. Laubscher, M. S. Kesselring, and J. R. Wootton, Poking holes and cutting corners to achieve Clifford gates with the surface code, Phys. Rev. X **7**, 021029 (2017).

[17] D. Litinski, A game of surface codes: Large-scale quantum computing with lattice surgery, Quantum **3**, 128 (2019).

[18] H. Bombin, C. Dawson, R. V. Mishmash, N. Nickerson, F. Pastawski, and S. Roberts, Logical blocks for fault-tolerant topological quantum computation, arXiv preprint ArXiv:2112.12160 (2021).

[19] H. Bombin and M. A. Martin-Delgado, Topological quantum distillation, Phys. Rev. Lett. **97**, 180501 (2006).

[20] A. J. Landahl, J. T. Anderson, and P. R. Rice, Fault-tolerant quantum computing with color codes, arXiv preprint ArXiv:1108.5738 (2011).

[21] M. Barkeshli, C.-M. Jian, and X.-L. Qi, Classification of topological defects in Abelian topological states, Phys. Rev. B **88**, 241103 (2013).

[22] M. Barkeshli, C.-M. Jian, and X.-L. Qi, Twist defects and projective non-Abelian braiding statistics, Phys. Rev. B **87**, 045130 (2013).

[23] T. J. Yoder and I. H. Kim, The surface code with a twist, Quantum **1**, 2 (2017).

[24] H. Bombin, Transversal gates and error propagation in 3D topological codes, arXiv preprint ArXiv:1810.09575 (2018).

[25] H. Bombin, 2D quantum computation with 3D topological codes, arXiv preprint ArXiv:1810.09571 (2018).

[26] A. Lavasani and M. Barkeshli, Low overhead Clifford gates from joint measurements in surface, color, and hyperbolic codes, Phys. Rev. A **98**, 052319 (2018).

[27] A. Lavasani, G. Zhu, and M. Barkeshli, Universal logical gates with constant overhead: Instantaneous Dehn twists for hyperbolic quantum codes, arXiv preprint ArXiv:1901.11029 (2019).

[28] P. Webster and S. D. Bartlett, Fault-tolerant quantum gates with defects in topological stabilizer codes, Phys. Rev. A **102**, 022403 (2020).

[29] S. Roberts and D. J. Williamson, 3-fermion topological quantum computation, arXiv preprint ArXiv:2011.04693 (2020).

[30] G. Zhu, T. Jochym-O'Connor, and A. Dua, Topological order, quantum codes and quantum computation on fractal geometries, arXiv preprint ArXiv:2108.00018 (2021).

[31] C. Chamberland and E. T. Campbell, Universal quantum computing with twist-free and temporally encoded lattice surgery, arXiv preprint ArXiv:2109.02746 (2021).

[32] A. J. Landahl and B. C. A. Morrison, Logical Majorana fermions for fault-tolerant quantum simulation, arXiv preprint ArXiv:2110.10280 (2021).

[33] K. Fukui, A. Tomita, A. Okamoto, and K. Fujii, High-threshold fault-tolerant quantum computation with analog quantum error correction, Phys. Rev. X **8**, 021054 (2018).

[34] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, Phys. Rev. A **71**, 022316 (2005).

[35] S. Bravyi and J. Haah, Magic-state distillation with low overhead, Phys. Rev. A **86**, 052329 (2012).

[36] A. G. Fowler, S. J. Devitt, and C. Jones, Surface code implementation of block code state distillation, Sci. Rep. **3**, 1 (2013).

[37] J. Haah, M. B. Hastings, D. Poulin, and D. Wecker, Magic state distillation with low space overhead and optimal asymptotic input count, Quantum **1**, 31 (2017).

[38] E. T. Campbell and M. Howard, Unified framework for magic state distillation and multiqubit gate synthesis with reduced resource cost, Phys. Rev. A **95**, 022316 (2017).

[39] J. Haah and M. B. Hastings, Codes and protocols for distilling t, controlled-s, and Toffoli gates, Quantum **2**, 71 (2018).

[40] C. Gidney and A. G. Fowler, Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $2|T\rangle$ transformation, Quantum **3**, 135 (2019).

[41] D. Litinski, Magic state distillation: Not as costly as you think, Quantum **3**, 205 (2019).

[42] A. Holmes, Y. Ding, A. Javadi-Abhari, D. Franklin, M. Martonosi, and F. T. Chong, Resource optimized quantum architectures for surface code implementations of magic-state distillation, Microprocess. Microsyst. **67**, 56 (2019).

[43] I. D. Kivlichan, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, W. Sun, Z. Jiang, N. Rubin, A. Fowler, and A. Aspuru-Guzik, *et al.*, Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via Trotterization, Quantum **4**, 296 (2020).

[44] V. von Burg, G. Hao Low, T. Häner, D. S. Steiger, M. Reiher, M. Roetteler, and M. Troyer, Quantum computing enhanced computational catalysis, arXiv preprint ArXiv:2007.14460 (2020).

[45] I. H. Kim, Y.-H. Liu, S. Pallister, W. Pol, S. Roberts, and E. Lee, Fault-tolerant resource estimate for quantum chemical simulations: Case study on Li-ion battery electrolyte molecules, Phys. Rev. Res. **4**, 023019 (2022).

[46] Y. Su, D. W. Berry, N. Wiebe, N. Rubin, and R. Babbush, Fault-tolerant quantum simulations of chemistry in first quantization, (2021), ArXiv:2105.12767.

[47] Y. Li, A magic state's fidelity can be superior to the operations that created it, New J. Phys. **17**, 023037 (2015).

[48] S. Singh, A. S. Darmawan, B. J. Brown, and S. Puri, High-fidelity magic-state preparation with a biased-noise architecture, Phys. Rev. A **105**, 052410 (2022).

[49] S. Bartolucci, P. Birchall, H. Bombin, H. Cable, C. Dawson, M. Gimeno-Segovia, E. Johnston, K. Kieling, N. Nickerson, and M. Pant, *et al.*, Fusion-based quantum computation, arXiv preprint ArXiv:2101.09310 (2021).

[50] D. Gottesman, Stabilizer Codes and Quantum Error Correction, Dissertation (Ph.D.), California Institute of Technology, 1997.

[51] D. Gottesman, in *Quantum information science and its contributions to mathematics, Proceedings of Symposia in Applied Mathematics*, Vol. 68 (2010), p. 13.

[52] H. Bombin, D. Litinski, N. Nickerson, F. Pastawski, and S. Roberts, Unifying flavors of fault tolerance with the ZX calculus, arXiv preprint ArXiv:2303.08829 (2023).

[53] X.-G. Wen, Quantum orders in an exact soluble model, Phys. Rev. Lett. **90**, 016803 (2003).

[54] A. Kay, Capabilities of a perturbed toric code as a quantum memory, Phys. Rev. Lett. **107**, 270502 (2011).

[55] J. Łodyga, P. Mazurek, A. Grudka, and M. Horodecki, Simple scheme for encoding and decoding a qubit in unknown state for various topological codes, Sci. Rep. **5**, 8975 (2015).

[56] B. J. Brown and S. Roberts, Universal fault-tolerant measurement-based quantum computation, Phys. Rev. Res. **2**, 033305 (2020).

[57] H. Bombin, I. H. Kim, D. Litinski, N. Nickerson, M. Pant, F. Pastawski, S. Roberts, and T. Rudolph, Interleaving: Modular architectures for fault-tolerant photonic quantum computing, arXiv preprint ArXiv:2103.08612 (2021).

[58] Here, we study the postselection overheads (for different rules) relative to the construction of a single block and do not compare overheads of postselection strategies on short-depth and larger-depth blocks.

[59] One may accomplish additional reduction of p_{init} with minimal overhead using hardware optimizations specifically tailored to the preparation point.

[60] For the preparation block, is equivalent to the graph distance on each of the syndrome graphs.

[61] V. Kolmogorov, BLOSSOM V: A new implementation of a minimum cost perfect matching algorithm, Math. Prog. Comput. **1**, 43 (2009).

[62] The logical gap generalizes the surviving-distance rule defined in Appendix A, which is designed for erasure errors. The gap rule effectively subsumes the distance rule, since it also incorporates syndromes resulting from Pauli errors, as seen in Figs. 3, 6, 13, and 14, and therefore is not shown.

[63] J. Haah, M. B. Hastings, D. Poulin, and D. Wecker, Magic state distillation at intermediate size, arXiv preprint ArXiv:1709.02789 (2017).

[64] K. J. Satzinger, *et al.*, Realizing topologically ordered states on a quantum processor, Science **374**, 1237 (2021).

[65] L. Egan, D. M. Debroy, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Newman, M. Li, K. R. Brown, and M. Cetina, *et al.*, Fault-tolerant control of an error-corrected qubit, Nature **598**, 281 (2021).

[66] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, and N. C. Brown, *et al.*, Realization of real-time fault-tolerant quantum error correction, arXiv preprint ArXiv:2107.07505 (2021).

[67] L. Postler, S. Heußen, I. Pogorelov, M. Rispler, T. Feldker, M. Meth, C. D. Marciniak, R. Stricker, M. Ringbauer, and R. Blatt, *et al.*, Demonstration of fault-tolerant universal quantum gate operations, arXiv preprint ArXiv:2111.12654 (2021).

[68] R. Acharya, I. Aleiner, R. Allen, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, J. Atalaya, and R. Babbush, *et al.*, Suppressing quantum errors by scaling a surface code logical qubit, arXiv preprint ArXiv:2207.06431 (2022).

[69] J. Gavriel, D. Herr, A. Shaw, M. J. Bremner, A. Paler, and S. J. Devitt, Transversal injection: A method for direct encoding of ancilla states for non-Clifford gates using stabiliser codes, arXiv preprint ArXiv:2211.10046 (2022).

[70] N. Delfosse and N. H. Nickerson, Almost-linear time decoding algorithm for topological codes, arXiv preprint ArXiv:1709.06218 (2017).

[71] O. Higgott and C. Gidney, SPARSE BLOSSOM: Correcting a million errors per core second with minimum-weight matching, arXiv preprint ArXiv:2303.15933 (2023).

[72] Y. Wu and L. Zhong, FUSION BLOSSOM: Fast MWPM decoders for QEC, arXiv preprint ArXiv:2305.08307 (2023).

[73] H. Bombín, C. Dawson, Y.-H. Liu, N. Nickerson, F. Pastawski, and S. Roberts, Modular decoding: Parallelizable real-time decoding for quantum computers, arXiv preprint ArXiv:2303.04846 (2023).

[74] Y. Ye, T. He, H.-L. Huang, Z. Wei, Y. Zhang, Y. Zhao, D. Wu, Q. Zhu, H. Guan, and S. Cao, *et al.*, Near-perfect logical magic state preparation on a superconducting quantum processor, arXiv preprint ArXiv:2305.15972 (2023).