

# Partially Fault-Tolerant Quantum Computing Architecture with Error-Corrected Clifford Gates and Space-Time Efficient Analog Rotations

Yutaro Akahoshi<sup>1,2,\*</sup>, Kazunori Maruyama<sup>1,2</sup>, Hirotaka Oshima<sup>1,2</sup>, Shintaro Sato<sup>1,2</sup> and Keisuke Fujii<sup>2,3,4,5</sup>

<sup>1</sup>Quantum Laboratory, Fujitsu Research, Fujitsu Limited, 4-1-1 Kawasaki, Kanagawa, 211-8588, Japan

<sup>2</sup>Fujitsu Quantum Computing Joint Research Division, Center for Quantum Information and Quantum Biology, Osaka University, 1-2 Machikaneyama, Toyonaka, Osaka, 565-8531, Japan

<sup>3</sup>Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka, 560-8531, Japan

<sup>4</sup>Center for Quantum Information and Quantum Biology, Osaka University, 1-2 Machikaneyama, Toyonaka, Osaka, 560-0043, Japan

<sup>5</sup>RIKEN Center for Quantum Computing, Wako, Saitama, 351-0198, Japan



(Received 28 March 2023; accepted 22 January 2024; published 5 March 2024)

Quantum computers are expected to drastically accelerate certain computing tasks versus classical computers. Noisy intermediate-scale quantum (NISQ) devices, which have tens to hundreds of noisy physical qubits, are gradually becoming available, but it is still challenging to achieve useful quantum advantages in meaningful tasks. On the other hand, full fault-tolerant quantum computing (FTQC) based on quantum error correction code remains far beyond realization due to its extremely large requirement of high-precision physical qubits. In this study, we propose a quantum computing architecture to close the gap between NISQ and FTQC architectures. Our architecture is based on erroneous arbitrary rotation gates and error-corrected Clifford gates implemented by lattice surgery. We omit the typical distillation protocol to achieve direct analog rotations and small qubit requirements, and minimize the remnant errors of the rotations by a carefully designed state injection protocol. Our estimation based on numerical simulations shows that for early-FTQC devices that consist of  $10^4$  physical qubits with physical error probability  $p = 10^{-4}$ , we can perform roughly  $1.72 \times 10^7$  Clifford operations and  $3.75 \times 10^4$  arbitrary rotations on 64 logical qubits. Such computations cannot be realized by the existing NISQ and FTQC architectures on the same device, as well as classical computers. We hope that our proposal and the corresponding development of quantum algorithms based on it will bring new insights into the realization of practical quantum computers in the future.

DOI: [10.1103/PRXQuantum.5.010337](https://doi.org/10.1103/PRXQuantum.5.010337)

## I. INTRODUCTION

Quantum computers are expected to provide exponential speedup of computation in certain tasks, including factoring [1], simulating quantum many-body systems [2,3], and linear algebraic operations [4]. To realize such a quantum computer, the development of quantum computing devices in various physical systems has been actively carried out in recent years. While fidelity and controllability are diverse, quantum devices with tens to hundreds of qubits have

emerged and are referred to as “noisy intermediate-scale quantum (NISQ) devices” [5]. Now we are entering an era of quantum computational supremacy [6–10], where simulating the behavior of a quantum computer itself is becoming intractable for a classical computer. Unfortunately, it is still challenging to extract useful quantum advantages over the classical best approaches from NISQ devices for practically meaningful tasks. Additionally, the classical simulation technology of a quantum computer using supercomputers has recently improved, and it has been reported that random quantum circuit sampling on Google’s quantum computer in 2019 can be simulated in comparable time [11].

The problem with NISQ devices is that they cannot provide an ultimate solution to the noise issue. Qubits and the gate operations on them lose their quantum nature due to decoherence caused by undesirable interactions with

\*akahoshi.yutaro@fujitsu.com

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article’s title, journal citation, and DOI.

the environment, which introduces errors into a quantum computer. Thus, useful tasks are difficult to perform reliably. Furthermore, the variational quantum algorithm [12] is based on the estimation of expectation values, and the number of measurements increases with the number of qubits. The accuracy becomes poor due to statistical errors as well as the effects of noise, and the optimization of the variational parameters becomes extremely difficult [12]. This problem may be solved by increasing the fidelity of quantum devices and by using techniques such as quantum noise mitigation [13], specifically designed for NISQ devices. However, it is a nontrivial question whether quantum noise mitigation can solve the noise problem at a realistic sampling size for quantum computation of the 50- to 100-qubit level, which is difficult to simulate even with a classical computer [14]. The ultimate long-term solution is the realization of fault-tolerant quantum computing (FTQC) by implementation of quantum error correction (QEC).

Several experiments have demonstrated the viability of QEC [15–17]. Error correction will soon allow us to store quantum information for longer than its physical coherence time, and to perform fault-tolerant logical gate operations. However, non-Clifford gates, which are necessary ingredients for quantum speedup, are difficult to implement fault-tolerantly on the QEC codes such as the surface code. A special protocol called “magic state distillation” is used to implement a non-Clifford  $T$  gate reliably [18]. Furthermore, the arbitrary angle rotation gates on a logical qubit require a huge number of  $T$  gates when they are decomposed into Clifford and  $T$  gates via Solovay-Kitaev decomposition. Together with the cost of magic state distillation and Clifford gate plus  $T$  gate decomposition, the realization of fully fledged FTQC requires an overhead of hundreds of thousands to millions of qubits [19–22].

In terms of the number of qubits for algorithm viability, a large gap will exist between the NISQ era and the FTQC era; the number of qubits that is needed for a meaningful quantum computation differs by several orders of magnitude. While experimental breakthroughs are expected to emerge to integrate  $\times 10^6$  qubits in the long term, in the meantime it is also necessary to establish a theoretical framework that meaningfully exploits early FTQC with  $10^3$ – $10^4$  qubits.

In this study, we propose a framework to hybridize NISQ and FTQC architectures to close the gap between them and provide evidence that a quantum computer of  $10^4$  qubits has great potential to exhibit quantum advantages in meaningful tasks. In this direction, quantum noise mitigation designed for NISQ devices has been applied for FTQC to reduce the required number of physical qubits, while magic state distillation still requires a huge number of physical qubits for quantum advantage [23,24]. Here we integrate the NISQ and FTQC approaches at a deeper level. More concretely, in our approach, continuous rotational

gates are not protected by QEC but are executed by injection of ancilla states without magic state distillation. This allows us to perform a rotation gate by an arbitrary angle directly. As a drawback, the injection of the ancilla states on a QEC code suffers from unavoidable errors. In our proposal, we carefully design an injection circuit so that most errors during the injection are detected and/or corrected so that the resultant special ancilla states have a minimum error. Furthermore, for the Clifford gates, such as the controlled-NOT (CNOT), Hadamard ( $H$ ), and  $S$  gates, we use the rotated planar surface code as usual, and hence errors are corrected in a scalable way. This allows an almost error-free implementation of logical Clifford operations versus the rotation gates.

Combining error-corrected Clifford gates and reasonably clean analog rotations, our resource estimation shows that  $3.75 \times 10^4$  arbitrary rotation gates and  $1.72 \times 10^7$  Clifford gates on 64 logical qubits are reliably executed with use of  $10^4$  physical qubits when the physical error probability is  $10^{-4}$ . Such computations cannot be simulated on classical computers, and even the existing NISQ and FTQC architectures on the same device cannot realize this amount of computational power. Our architecture can be applied to useful tasks such as quantum many-body simulation and the quantum approximation optimization algorithm (QAOA) thanks to the fast implementation of the analog rotations. The proposed space-time efficient analog rotation quantum computing architecture (hereinafter referred to as “STAR architecture”) provides a new framework for the use of quantum computers that fills the gap between the NISQ era and the FTQC era.

## II. OVERVIEW OF THE STAR ARCHITECTURE

Before delving into a comprehensive description, we provide an overview of the STAR architecture in this section.

Currently, we are in the NISQ era, and the number of qubits is increasing to hundreds. However, it will be extremely difficult to fully exploit the computational power of NISQ devices with hundreds to thousands of qubits, because NISQ devices suffer from errors in both Clifford and non-Clifford operations. The number of gates increases due to the SWAP operations at the stage of compiling a quantum algorithm to be executable on actual quantum computing devices with limited qubit connectivity. Moreover, in applications to quantum chemistry, fermionic rotations, such as the unitary coupled cluster ansatz, require the entangling gates to generate multi-qubit Pauli rotations. Most of the gates used there are Clifford gates such as CNOT gates, which do not make classical simulation difficult from the viewpoint of the Gottesman-Knill theorem, while they result in the accumulation of errors. As a result, the total number of non-Clifford gates that can be executed is rather limited.

QEC is a method for entangling multiple qubits and encoding quantum information in a special subspace to protect it from noise [25]. Unlike a classical bit, a qubit, which takes a superposed state through continuous complex probability amplitudes, suffers from continuous analog noise. The orthogonal subspace structure introduced by a QEC code can collapse such analog noise into digitalized Pauli  $X$ ,  $Y$ , and  $Z$  errors, which are corrected appropriately. However, the orthogonal subspace structure also makes operations of the encoded degrees of freedom (DOF) difficult [26]. Particularly, a fault-tolerant implementation of the  $T$  gate, which is a non-Clifford gate and an essential ingredient for universal quantum computation, on an encoded degree of freedom is highly nontrivial. Most QEC codes do not support fault tolerance for the  $T$  gate in a native way. A special protocol, called “magic state distillation” [18], is necessary to purify noisy magic states and execute a  $T$  gate via gate teleportation [27]. Furthermore, since the  $T$  gate is an  $\pi/8$  rotation gate around the  $z$  axis, an arbitrary rotation gate can be compiled to the sequence of Clifford gates and  $T$  gates by use of the Solovay-Kitaev algorithm. The state-of-the-art optimal Clifford gate plus  $T$  gate decomposition [28] still requires several tens of  $T$  gates to achieve the  $10^{-4}$  accuracy of an arbitrary single-qubit rotational gate, even if this accuracy can be achievable by physical single-qubit rotation.

If a quantum computer can execute all Clifford operations ideally and errors are introduced only in analog non-Clifford operations, more advanced quantum algorithms can be executed even in the era of early FTQC. Our approach is to construct such an architecture by successfully combining fault-tolerant error correction in FTQC

and analog operations in NISQ devices. An overview of the STAR architecture is summarized in Fig. 1. The key points are as follows:

- (i) Fault-tolerant Clifford gates with QEC.
- (ii) Analog rotation gates with reasonably clean ancilla state injection.

Regarding point (i), since the Clifford gates are protected by QEC, the connectivity of physical qubits and Clifford transforms for gates such as multi-qubit Pauli rotations are not limiting factors to design reliable quantum computing. We use the rotated planar surface code [29] as a logical qubit and use lattice surgery [30] to implement the fault-tolerant logical Clifford gates, as explained in Sec. III.

On the other hand, by virtue of point (ii), we can avoid the magic state distillation, which is the costly part of FTQC. This also successfully reduces the computational cost in a double sense in that it eliminates the need for decomposition into  $T$  gates when one is performing continuous rotation gates. As explained in Sec. IV, we carefully design a quantum circuit to inject a special ancilla state into the planar surface code with error detection and post-selection. Then the reasonably clean ancilla states are used to implement analog rotation gates via gate teleportation, where the byproduct is treated by a repeat-until-success (RUS) approach. In Sec. V, we provide typical logical qubit arrangements in the STAR architecture. As shown in Sec. VI, according to our numerical simulations, the STAR architecture surpasses the existing NISQ and FTQC architectures on the early-FTQC device as well as classical computers. We also discuss the possible applications

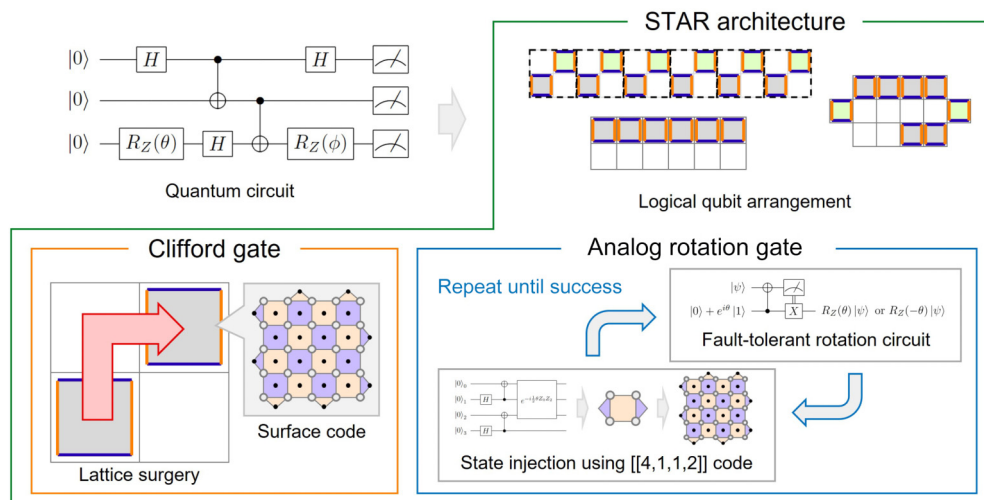


FIG. 1. Overview of the STAR architecture proposed in this work. The STAR architecture performs universal quantum computation using the error-corrected Clifford gate and an analog rotation gate. Clifford gates are implemented by lattice surgery based on the rotated planar surface code. Analog rotation is directly performed, avoiding the magic state distillation, and a special ancilla state needed for the rotation is cleanly injected into the rotated planar surface code by error detection and postselection. Combined with an appropriate logical qubit arrangement, the STAR architecture fully exploits the computational power of early-FTQC devices.

of our architecture there. Section VII concludes this study and provides a discussion of future directions.

### III. FAULT-TOLERANT CLIFFORD GATES

To make the discussion self-contained, we start by reviewing the existing approaches for encoding quantum information into the rotated planar surface code and protecting Clifford gates on the rotated planar surface codes.

#### A. Rotated planar surface code

The rotated planar surface code is a QEC code that has good features suitable for early-FTQC devices: a relatively high threshold value in comparison to other QEC codes and the requirement of a small number of physical qubits [29,31]. We summarize its definition in Fig. 2.

Physical qubits constructing the rotated planar surface code are arranged on the vertices of a two-dimensional lattice (white circles in Fig. 2).  $X$  ( $Z$ ) stabilizer operators are defined on the faces of the lattice [orange (blue) faces in Fig. 2], and the logical state is defined as the simultaneous eigenstate of those stabilizer operators with eigenvalues of +1. A single surface has two types of boundary, namely, the  $X$  and  $Z$  boundaries, along which the logical  $X$  and  $Z$  operators are defined as chains of physical  $X$  and  $Z$  operators (orange and blue lines on the boundaries in Fig. 2). The code distance  $d$  is equal to the length of a side of the lattice. In the following discussion, we call a surface that carries a single logical state a “logical patch” or simply a “patch.”

To perform the error correction, one measures the eigenvalues of the stabilizers using measurement qubits arranged on the faces of the lattice (black circles in Fig. 2). Measured eigenvalues are called “syndromes” and are used to infer a possible error pattern. The syndrome measurement circuits that we use in this study are shown in Fig. 3. Using this circuit, we can measure simultaneously all

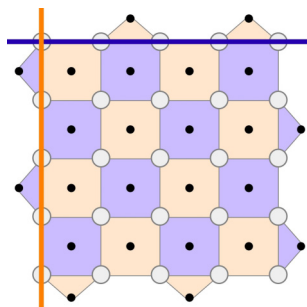


FIG. 2. Definition of the rotated planar surface code with code distance  $d = 5$ . White and black circles represent physical qubits used for encoding and measurement qubits, respectively. Stabilizer operators, which define the logical state, are shown as orange and blue surfaces. Representative logical operators are given as solid lines on boundaries.

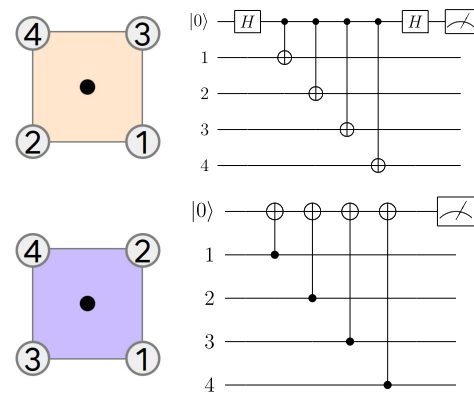


FIG. 3. Syndrome measurement circuits.  $X$  syndrome measurement circuit (top) and  $Z$  syndrome measurement circuit (bottom). The order of CNOT operations is represented by circled numbers on the left-hand side.

syndromes in eight fundamental operation steps. The order of CNOT operations between physical and measurement qubits in Fig. 3 is important for preserving commutation relations between stabilizer operators. In this study, we use the order proposed in Ref. [32] to prevent hook errors along the logical operators.

Errors that occur in the logical qubit are inferred by observed error syndromes, where the eigenvalue of the stabilizer is flipped from +1 to  $-1$ . In the surface code, we can infer the most likely error pattern as follows. First, we construct a decoder graph, in which syndromes and error events are represented by nodes and edges, respectively. Paths that connect error syndromes in the graph provide candidates for the actual error pattern, and their length is related to the number of errors. Therefore, we can adopt the shortest path among these candidates as the most likely error pattern by assuming that the errors occur independently. The shortest path connecting the error syndrome is determined by a certain matching algorithm, e.g., the Edmonds minimum-weight perfect matching (MWPM) algorithm [33]. In practice, measured syndromes are also unreliable due to measurement errors; thus, the syndrome measurement is performed  $d$  times and the differences on the time axis are calculated by our taking the XOR operation of the temporally neighboring two syndromes (in the following, we call these differences “syndromes” unless stated otherwise). Then we can construct a spatiotemporal decoder graph from  $d$  sets of syndromes and infer the most likely error chains including the measurement errors by the MWPM algorithm. In addition to the MWPM algorithm, several other ways to perform this error inference have been proposed, such as the union-find algorithm [34], the renormalization group decoder [35], and the Ising model-based approach [36,37]. In this study, we use the MWPM algorithm to benchmark the performance of the logical patch.

### B. Clifford gates by lattice surgery

In principle, logical Clifford gates such as the CNOT gate and the Hadamard gate can be transversally performed in the planar rotated surface code [29]. In practice, however, the transversal CNOT gate is difficult to realize for some devices in which the connectivity between physical qubits is restricted. A clever way to implement Clifford gates in this situation is lattice surgery, which consists of two-patch merging, splitting, and patch deformation [29,30]. The Clifford gates implemented in the STAR architecture rely on this technique. Here we discuss typical examples to implement the logical CNOT gate and the logical Hadamard gate, based on the fundamental operations introduced in Ref. [30].

A standard logical CNOT operation using lattice surgery is achieved by merging and splitting a control logical qubit  $|C\rangle$  and a target logical qubit  $|T\rangle$ . Figure 4 shows a sequence of lattice surgery operations for performing the logical CNOT operation. The logical states  $|C\rangle$  and  $|T\rangle$  are placed as in Fig. 4(a). Then the following lattice surgery operations are performed: expand  $|C\rangle$  along the  $X$  boundary [Fig. 4(b)], split  $|C\rangle$  into two patches by the  $Z$  boundary and merge one of them with  $|T\rangle$  along the  $X$  boundary [Fig. 4(c)], and contract  $|T\rangle$  along the  $Z$  boundary [Fig. 4(d)]. The operations in Figs. 4(b) and 4(c) need  $d$  rounds of syndrome measurement to determine syndrome values, and a total of  $2d$  rounds for the logical CNOT operation. If the measured  $X_L \otimes X_L$  eigenvalue, which is a product of the eigenvalues of the  $X$  stabilizers newly introduced in the  $X$  boundary merging in the operation in Fig. 4(c), is  $-1$ , a byproduct operator  $Z_L$  subsequently acts on  $|C\rangle$ .

A logical Hadamard gate is simply achieved by transversally applying a physical Hadamard gate on all data qubits. An important obstacle is that the logical qubit patch after the operation rotates  $90^\circ$  from the original orientation [Fig. 5(a)]. This rotated orientation can be corrected by the lattice surgery operations. A typical sequence of the operation is shown in Fig. 5: expand a patch [Fig. 5(b)],

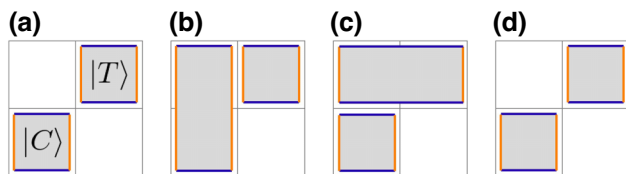


FIG. 4. Logical CNOT operation by lattice surgery. Blue (orange) lines indicate  $Z$  ( $X$ ) boundaries. (a) Initial configuration of two logical patches. (b) Expansion of the control patch along the  $X$  boundary. (c) Splitting of the control patch along the  $Z$  boundary and merging of one of them with the target patch along the  $X$  boundary. These two operations can be performed simultaneously. (d) Contraction of the target patch along the  $Z$  boundary.

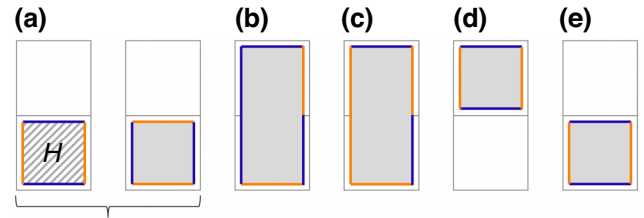


FIG. 5. Logical Hadamard operation with a boundary rotation by lattice surgery. (a) After application of the transversal Hadamard gate on a certain patch, its orientation rotates by  $90^\circ$ . To fix its orientation, (b) expand the patch first, (c) modify its boundary, and (d) contract the patch. At this moment its orientation is fixed correctly. (e) The patch is moved to the original position. This operation can be achieved by combining patch expansion and contraction.

deform the patch boundary [Fig. 5(c)], contract the patch [Fig. 5(d)], and move the patch to the original position [Fig. 5(d)]. In this example, the operations in Figs. 5(b), 5(c), and 5(e) need  $d$  rounds of syndrome measurement; thus,  $3d$  rounds are required in total.

Litinski [30] proposed another way to perform fault-tolerant quantum computation, in which the Clifford gates in quantum circuits are moved to the end of the circuits and absorbed into measurements. The modified circuits contain multi-qubit Pauli measurements and multi-qubit Pauli  $\pi/8$  rotations, which can also be performed by lattice surgery. A typical example of measuring a multi-qubit Pauli operator  $X_{L,1} \otimes Y_{L,2} \otimes Z_{L,3}$  is given in Fig. 6. Physical qubits in an ancilla region are first initialized to  $|+\rangle$  (a red region in Fig. 6), and then the stabilizer operators in the entire region are measured (including the hatched area in Fig. 6). The product of the eigenvalues of stabilizer operators whose eigenvalues are not determined by the initialization gives the measurement result of  $X_{L,1} \otimes Y_{L,2} \otimes Z_{L,3}$ . This operation is performed by  $d$  rounds of syndrome measurement.

The Clifford operations discussed in this section are closely related to the arrangement of logical qubits, which

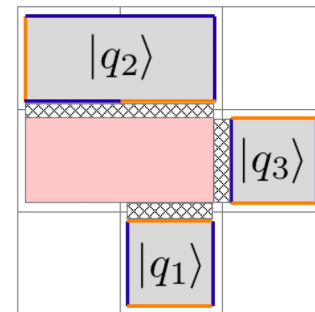


FIG. 6. Example of the multi-qubit Pauli measurement  $X_{L,1} \otimes Y_{L,2} \otimes Z_{L,3}$ .

plays an important role in performing quantum computations with small overheads. We discuss typical examples of the arrangement in Sec. V.

#### IV. SPACE-TIME EFFICIENT ANALOG ROTATION GATE

In this section, we discuss how to implement analog rotation gates with reasonable accuracy, which is a core technology of the STAR architecture. We directly perform analog rotation gates without the lengthy Solovay-Kitaev decomposition and avoid the costly magic state distillation. This approach is advantageous in terms of a physical qubit requirement and execution time. However, a major challenge is that the logical error rate of the analog rotation becomes relatively large at  $O(p)$ . To minimize the logical error rate, in our proposal, we carefully design a state injection protocol needed to generate a special ancilla state for the rotation. The remaining logical error of the analog rotation becomes a simple phase-flip error and can be further mitigated by the probabilistic error cancellation when the physical error probability is sufficiently small.

##### A. Repeat-until-success implementation of analog rotation gate

In the typical Clifford gate plus  $T$  gate decomposition in FTQC, the  $T$  gate is implemented by the gate teleportation circuit with the magic state. Furthermore, to achieve an analog rotation gate with sufficient accuracy, approximately 100  $T$  gates are required via the Solovay-Kitaev decomposition [28]. In contrast, the STAR architecture directly implements the analog rotation gate by using a special ancilla state  $|m_\theta\rangle \equiv R_Z(\theta)|+\rangle = \frac{1}{\sqrt{2}}(e^{-i\theta/2}|0\rangle + e^{+i\theta/2}|1\rangle)$ , where the angle  $\theta$  can be chosen arbitrarily. The circuit for implementing the analog rotation is shown in Fig. 7.

Since we allow arbitrary rotation angles, this implementation is not deterministic: an output state is a correctly rotated state  $R_Z(\theta)|\psi\rangle$  if the measurement result in the circuit is  $+1$ ; otherwise the output is an inversely rotated state,  $R_Z(-\theta)|\psi\rangle$ . Both outputs evenly occur. If the inversely rotated state is obtained, we apply a rotation gate with angle  $2\theta$  on the output state to correct its angle. This correction is repeated until we obtain  $R_Z(\theta)|\psi\rangle$  (RUS).

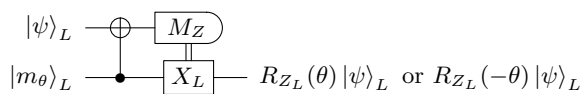


FIG. 7. Quantum circuit for the analog  $Z$  rotation gate.  $M_Z$  is a destructive  $Z_L$  measurement on a logical patch.

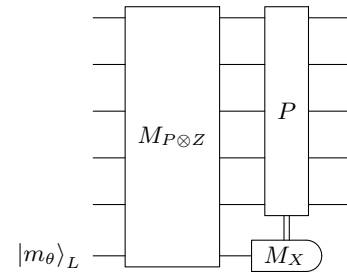


FIG. 8. Quantum circuit for the analog multi-qubit Pauli rotation gate.

The average RUS step number for success is given as

$$1 \times \frac{1}{2} + 2 \times \frac{1}{4} + 3 \times \frac{1}{8} + \dots = \sum_{i=1}^{\infty} \frac{n}{2^n} = 2. \quad (1)$$

The Clifford gates in Fig. 7 are performed by lattice surgery.

In the computational scheme using the multi-qubit Pauli measurement and the multi-qubit Pauli  $\pi/8$  rotation gate as mentioned in the previous section, the  $\pi/8$  rotation gate needs to be extended to an arbitrary angle multi-qubit Pauli  $P$  rotation gate (e.g.,  $P = X \otimes Y \otimes Z$ ). Such multi-qubit Pauli rotation gates can be realized by a multi-qubit Pauli  $P \otimes Z$  measurement with target logical qubits and the ancilla state  $|m_\theta\rangle$  [30], as in the circuit shown in Fig. 8. If the measurement result of  $P \otimes Z$  is  $+1$ , the rotation succeeds, otherwise we must apply  $R_P(2\theta)$  to the output state to correct its angle. An  $X$  measurement in the circuit checks whether the output state has a byproduct operator  $P$ . Since the byproduct  $P$  commutes with  $R_P(\theta)$  and satisfies  $P^2 = I$ , it is sufficient to cancel it after completion of the entire RUS protocol if the product of all  $X$  measurement values is  $-1$ .

##### B. Low-error state injection protocol

As discussed above, an analog rotation is implemented by circuits comprising error-corrected Clifford gates. Therefore, the accuracy of the analog rotation is dominated by the state injection protocol of the special ancilla state  $|m_\theta\rangle$ . In this section, we discuss a low-error state injection protocol based on postselection.

Error detection and postselection is a promising approach for near-term quantum computation. For example, a recent study [38] proposed a well-designed error detecting code and increased calculation accuracy by postselection over the entire calculation. In contrast, our postselection protocol is independent of the data logical patches involved in the main calculation; thus, our protocol is scalable to the overall size of the calculation.

The first step of our injection protocol is to generate the ancilla state encoded in the  $[[4, 1, 1, 2]]$  subsystem code

[39]. This code is defined by four physical qubits (we index them by subscripts 0–3 in the following discussion) with two stabilizer operators

$$S_X = X_0X_1X_2X_3, \quad S_Z = Z_0Z_1Z_2Z_3. \quad (2)$$

The +1 eigenstate of these stabilizers defines a logical qubit with logical operators

$$L_X = X_0X_1, \quad L_Z = Z_0Z_2, \quad (3)$$

and gauge operators

$$G_X = X_0X_2, \quad G_Z = Z_0Z_1. \quad (4)$$

The code distance is 2, so it can detect a single error.

A circuit for preparing the ancilla state  $|m_\theta\rangle_L$  encoded in the  $[[4, 1, 1, 2]]$  subsystem code is shown in Fig. 9. Hadamard and CNOT operations encode the input state into the  $|+\rangle_L$  state first, and then the logical rotation gate  $R_{Z_0Z_2}(\theta) \equiv e^{-i\theta/2(Z_0Z_2)}$  acts on  $|+\rangle_L$ . The output state is therefore  $|m_\theta\rangle_L$  up to an irrelevant overall factor. We assume that the gate  $R_{Z_0Z_2}(\theta) \equiv e^{-i\theta/2(Z_0Z_2)}$  can be directly performed here.

The generated ancilla state may suffer from errors in practice; thus, we measure syndromes of the  $[[4, 1, 1, 2]]$  subsystem code. Naively, we must measure the weight-4 stabilizer operator defined in Eq. (2). Because of the gauge DOF, however, we can measure them as products of weight-2 gauge operators, whose measurements do not collapse the logical qubit. This property reduces noise in the ancilla state since it avoids critical weight-2 hook errors propagating from the measurement qubits and reduces the depth of the measurement circuit. As shown in Fig. 10, we assume that four measurement qubits interact with the two nearest physical qubits (black circles labeled as M0–M3). This arrangement can be smoothly embedded in the rotated surface code as discussed later. The measurement circuit based on this arrangement is shown in Fig. 11. To detect measurement errors, the measurement circuit is performed twice, and we discard the prepared state if the measured syndromes satisfy one of the following conditions (post-selection): (i) one (or both) of the syndromes is equal to  $-1$  or (ii) one (or both) of the bare (the XOR operation is not performed) syndromes of the first round takes the

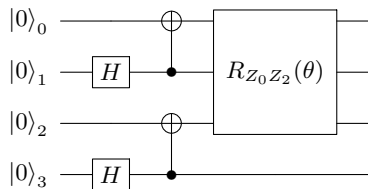


FIG. 9. Quantum circuit for the injection of the ancilla state encoded in the  $[[4, 1, 1, 2]]$  subsystem code.

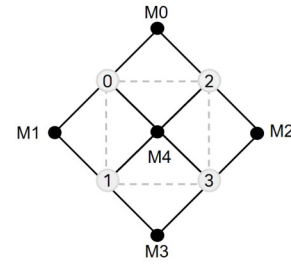


FIG. 10. Arrangement of physical and measurement qubits encoded in the  $[[4, 1, 1, 2]]$  subsystem code. White circles labeled as 0–3 and black circles labeled as M0–M3 represent physical qubits and measurement qubits, respectively. The solid line shows the connectivity between measurement qubits and physical qubits.

value  $-1$ , although the syndromes are equal to  $+1$ . The state injection circuit is repeated until this postselection is passed.

Once we obtain the ancilla state  $|m_\theta\rangle_L$  that passes the postselection, we expand it to the rotated surface code with an arbitrary code distance. After the measurement circuit in Fig. 11, the gauge DOF are fixed to the eigenstate of  $G_Z = Z_0Z_1$ . In other words, the postselected state is stabilized by

$$S_X = X_0X_1X_2X_3, \quad S'_Z = Z_0Z_1, \quad S''_Z = Z_2Z_3, \quad (5)$$

which is the smallest rotated planar surface code with  $d = 2$ . Therefore, its expansion to an arbitrary code distance patch can be immediately achieved in a standard way in lattice surgery [29]. We show an example of the expansion to the  $d = 5$  patch in Fig. 12.

The ancilla state  $|m_\theta\rangle_L$  is prepared on a certain corner of the target patch, and other physical qubits in the target patch are initialized to  $|0\rangle$  (blue circles in Fig. 12) or

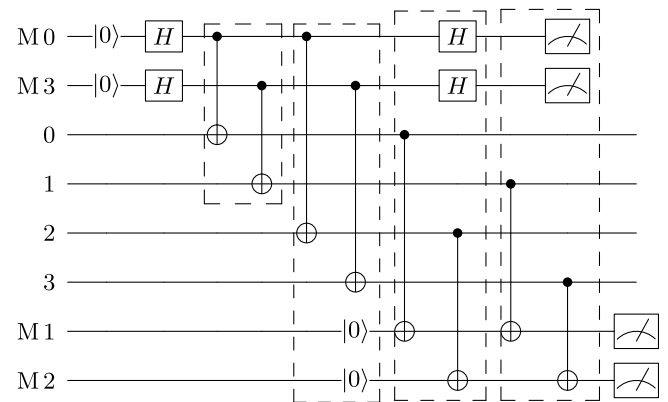


FIG. 11. Syndrome measurement circuit of the  $[[4, 1, 1, 2]]$  subsystem code using the gauge operators. Numbers 0–3 and M0–M3 represent physical qubits and measurement qubits, respectively. Meter symbols indicate  $Z$  measurements. Gates grouped by dashed lines are implemented simultaneously.

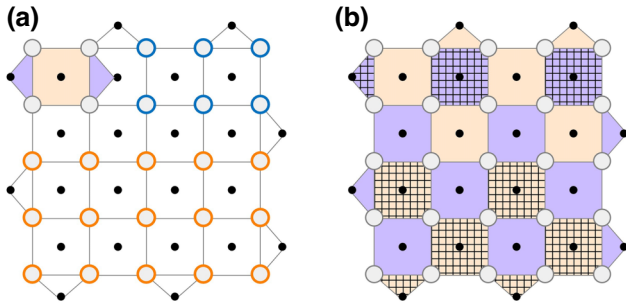


FIG. 12. Expansion of the ancilla state to the  $d = 5$  surface code patch. (a) The ancilla state is prepared at the upper-left corner of the target patch. Other physical qubits are initialized to  $|0\rangle$  (blue circles) or  $|+\rangle$  (orange circles). (b) Expansion is achieved by measuring stabilizer operators of the entire patch twice. After the syndrome measurement, some stabilizers (shown by the hatched pattern) have fixed eigenvalues determined by the initial configuration in an error-free case. These deterministic syndromes enables us to detect errors occurring on data qubits.

$|+\rangle$  (orange circles in Fig. 12). The success rate of the postselection can be increased by performing the injection protocol in parallel using the empty space in the target patch. In this case, one picks up a successful ancilla state after the parallel injection and refreshes and initializes other physical qubits in the target patch. Then the syndrome measurement of the entire patch is performed twice. To obtain the ancilla state as clean as possible, we discard the expanded state if one of the following conditions is satisfied: (i) at least one of the syndromes is equal to  $-1$  or (ii) at least one of the bare syndromes whose value is determined by the initial configuration [shown by the hatched pattern in Fig. 12(2)] takes an unexpected value, although all syndromes are equal to  $+1$ . The state that passes the second postselection is clean in detectable  $O(p)$  error, and it can be consumed in the gate teleportation circuit in Fig. 7 or the multi-qubit Pauli rotation in Fig. 8.

Under the circuit-level noise model introduced in Sec. VIA, the logical error probability of the prepared ancilla state  $|m_\theta\rangle_L$  behaves as follows:

$$P_{Z_L}(p) = 2p/15 + O(p^2), \quad (6)$$

$$P_{X_L}(p) = O(p^2), \quad (7)$$

whose details are discussed in the Appendix. Compared with the previous state injection protocols, our protocol achieves greater precision even in more general situations. We now briefly discuss the difference between other protocols and our protocol. In typical state injection protocols, one first prepares the ancilla state on a single physical qubit, and then it is injected into the encoded logical qubit. Since the first step directly suffers from the noisy qubit initialization and noisy single-qubit operation, the injected state has a large logical error rate proportional to  $p$ . For

example, the injection protocol proposed in Ref. [40] and its improved version of the rotated planar surface code [41] show  $P_L = 46p/15 + O(p^2)$  and  $P_L = 34p/15 + O(p^2)$ , respectively. In contrast, our protocol first generates an encoded qubit  $|+\rangle_L$ , and then applies the logical  $Z$  rotation gate  $R_{Z_0 Z_2}(\theta)$  on the encoded qubit to generate the ancilla state. The  $[[4, 1, 1, 2]]$  subsystem code is  $d = 2$  and most logical errors in our protocol occur at  $O(p^2)$ . Moreover, possible  $O(p)$  logical errors are absorbed in part by the redundant gauge DOF and the circuit structure. Thus, our protocol achieves smaller error probability than the above-mentioned protocols. The noisy non-Clifford gate can also be performed by use of the code deformation [24], but it has a large logical error rate  $P_L \approx 30p$  [24]. Another recently proposed protocol is transversal injection [42], in which physical qubits are transversally initialized in a certain state before the encoding, and then a random state is injected depending on the initialization. By performing a postselection, Gavriel *et al.* [42] report that the logical error rate of the injected state behaves as  $P_L \approx 0.39p$ . Our protocol is advantageous for injecting a certain target state with high accuracy because the injected state has no randomness and achieves greater accuracy. Other approaches use distance-2 codes utilizing repetition code [43] and weight-2 hook propagation [44]. However, the former method is weak for  $O(p)$  bit-flip errors and the latter method still suffers from the propagation of a single-qubit error that results in a large logical error rate when the single-qubit error is not negligible. Because our protocol is robust with regard to  $O(p)$  bit-flip errors and does not assume that the single-qubit errors are negligible, it is more versatile.

Although our implementation has a small logical error rate, logical errors still occur at  $O(p)$ . These remnant errors can be mitigated if we consider the evaluation tasks for the expectation value of some observable. Fortunately, some proposed algorithms for the early-FTQC era, such as resource-efficient phase estimation [45], are based on certain expectation values, so the error mitigation technique can be applied to obtain accurate results. One possible mitigation technique applicable to the STAR architecture is probabilistic error cancellation (or quasiprobability decomposition) [46,47]. We consider the case where the noise channel is known as a simple phase-flip channel with error probability  $P$ ,

$$\mathcal{E}(\rho) = (1 - P)\rho + PZ\rho Z. \quad (8)$$

In this case we can explicitly construct an “inverse error channel”  $\mathcal{E}^{-1}$  as

$$\mathcal{E}^{-1}(\rho) = \frac{1 - P}{1 - 2P}\rho - \frac{P}{1 - 2P}Z\rho Z, \quad (9)$$



and we can rewrite the noise-free (identity) channel as

$$\mathcal{I} = \mathcal{E}^{-1}\mathcal{E} = \gamma((1-P)\mathcal{E} - P\mathcal{Z}\mathcal{E}), \quad (10)$$

where  $\gamma = 1/(1-2P)$  and  $\mathcal{Z}$  is a Pauli  $Z$  channel. Therefore, an error-free expectation value of a certain operator  $M$  can be estimated by the noisy counterpart as

$$\langle M \rangle_{\mathcal{I}} = \gamma((1-P)\langle M \rangle_{\mathcal{E}} - P\langle M \rangle_{\mathcal{Z}\mathcal{E}}), \quad (11)$$

where  $\langle M \rangle_{\mathcal{N}} = \text{tr}(M\mathcal{N}(\rho))$ . By performing Monte Carlo sampling on an additional  $\mathcal{Z}$  channel with probability  $P$ , we can approximate Eq. (11) by averaging those samples with correct overall factors of  $\pm\gamma$ . The variance of the expectation value is amplified by a factor of  $\gamma^2$  as seen in Eq. (11), so we need to generate  $\gamma^2$  times more samples to suppress amplified statistical fluctuations.

Returning to our analog rotation gate, its leading-order logical error channel is well described by the phase-flip channel with  $P_L = 2p/15$ . If the  $O(p^2)$  contribution is negligible, we can completely mitigate the phase-flip channel by the probabilistic error cancellation. Even if the  $O(p^2)$  logical error contribution remains, we can still mitigate the dominant  $O(p)$  contribution by the inverse bit-flip channel with  $P_L = 2p/15$ . The fact that the leading-order logical error is a simple phase-flip channel is another benefit of our injection protocol. Note that the total step number of the RUS process varies in each sample, so we cannot directly mitigate the errors of each RUS step. Instead, we consider the entire RUS process as a single noisy operation and mitigate its error. The logical  $Z$  error probability of the entire RUS process is given as

$$P = \sum_{n=1}^{\infty} \frac{1}{2^n} P_{Z,n}, \quad (12)$$

where  $P_{Z,n}$  is an error probability when the RUS process is completed by  $n$ th step:

$$\begin{aligned} P_{Z,n} &= \sum_{m=1}^{\lfloor \frac{n+1}{2} \rfloor} \binom{n}{2m-1} P_{Z,1}^{2m-1} (1-P_{Z,1})^{n-2m+1} \\ &= nP_{Z,1} + O(P_{Z,1}^2), \end{aligned} \quad (13)$$

$$P_{Z,1} = 2p/15. \quad (14)$$

Since  $O(P_{Z,1}^2) = O(p^2)$  can be ignored, Eq. (12) becomes

$$P = \sum_{n=1}^{\infty} \frac{1}{2^n} P_{Z,n} \approx P_{Z,1} \sum_{n=1}^{\infty} \frac{n}{2^n} = 2P_{Z,1}. \quad (15)$$

Therefore, we can mitigate the phase-flip error of the entire RUS process by the probabilistic error cancellation with  $P = 2P_{Z,1}$  by an additional sampling overhead of  $\gamma^2 =$

$(1/(1-2P))^2 \approx e^{8P_{Z,1}}$  ( $P \ll 1$ ). When we perform  $N$  analog rotations in a circuit and want to mitigate their noise, we can immediately extend the discussion by assuming that  $N$  noise channels are independent. In such a case, the sampling overhead is modified to  $\gamma^{2N} \approx e^{8P_{Z,1}N}$ . This indicates that the sampling overhead grows exponentially with increasing number of noisy gates. One typical bound of the realistic number of noisy gates is  $N \approx 1/P$ , where the sampling overhead is  $e^4 \approx 55$ . Note that this overhead gives the worst-case estimation since some observables are affected by noise only in the relevant causal cone [48].

Finally, we briefly discuss how the restrictions of real devices affect our protocol. In real quantum devices, there are several restrictions on the connectivity between physical qubits and native gate sets. Regarding the connectivity restriction, for example, the superconducting qubits can interact only with their nearest neighbors. If we consider this restriction in a qubit arrangement such as that in Fig. 10, only qubits connected by solid black lines interact with each other; thus, the CNOT operation and  $R_{Z_0Z_2}(\theta)$  in the circuit in Fig. 9 cannot be performed directly. In this situation, one can resolve the connectivity problem by additionally inserting SWAP gates in the circuit. We show a simple example to perform our state injection circuit with SWAP gates in Fig. 13.

Inserted SWAP gates also introduce additional two-qubit errors, but they occur on certain pairs of measurement and physical qubits, so they do not lead to logical errors at  $O(p)$ . Therefore, the performance of our protocol at  $O(p)$  can be maintained. Regarding the restriction of native gates, our assumption that the  $R_{Z_0Z_2}(\theta)$  gate can be directly applied may become invalid in some cases. For typical ion-trap devices and superconducting devices, this assumption is valid since the  $X \otimes X$  rotation and the  $Z \otimes X$  rotation (the cross-resonance gate), respectively, can be directly implemented. If such gates are not supported, indirect implementations of the  $R_{Z_0Z_2}(\theta)$  gate shown should be used. One straightforward example is the circuit shown in upper part of Fig. 14. In this example, the logical error rate of the injected state degrades to  $P_L = 9p/15 + O(p^2)$  under the circuit-level noise model since the single  $Z$  errors and their propagation through the second CNOT gate lead

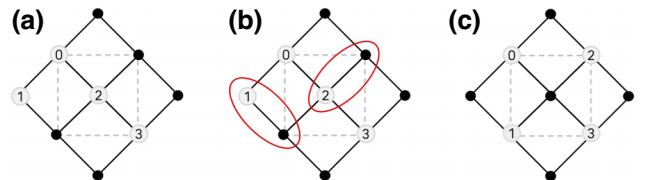


FIG. 13. Example of the insertion of SWAP gates. (a) Initial physical qubit arrangement. The circuit in Fig. 9 is performed in this arrangement. (b) Physical qubits 1 and 2 are then swapped to the neighboring measurement qubits (red circled pairs). (c) Final arrangement after SWAP operations.

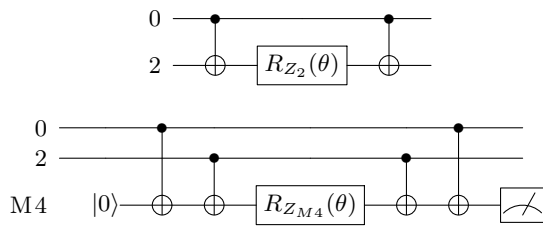


FIG. 14. Examples of a circuit for indirectly performing the  $R_{Z_0Z_2}(\theta)$  operation. Top:  $R_{Z_0Z_2}(\theta)$  can be performed with use of a single-qubit rotation  $R_{Z_2}(\theta)$  and CNOT operations. The logical error rate degrades to  $P_L = 9p/15 + O(p^2)$  in this circuit since there are more error patterns generating the logical Z error. Bottom: An alternative  $R_{Z_0Z_2}(\theta)$  operation circuit using another ancilla qubit, M4. The final Z measurement of the ancilla qubit M4 is necessary to detect  $O(p)$  X errors that flip  $\theta$  to  $-\theta$ . The logical error rate becomes  $P_L = 7p/15 + O(p^2)$  in this circuit.

to additional undetectable logical Z errors. If an ancilla qubit is available for this operation, we can use another circuit, where the logical error rate behaves as  $P_L = 7p/15 + O(p^2)$  [Fig. 14 (Lower)]. The ancilla qubit is measured by the Z basis after the operation to detect X errors that flip the rotation angle  $\theta$  to  $-\theta$ . If any X error is detected, the injection protocol will be restarted. Such variations for the  $Z \otimes Z$  rotation can also be useful to reduce the coherent error of the rotation gate, which may cause, for example, logical over-rotation. Although the detailed discussion of the coherent error depends on the physical realization of qubits and gates, one can, for example, perform the  $Z \otimes Z$  rotation by using the circuit in the upper part of Fig. 14 with the virtual Z rotation gate. In this case, the virtual Z rotation is free from the coherent error, and other coherent noise in CNOT operations can be twirled and partially removed by postselection. In the resource estimation discussed in Sec. VI, we directly perform the circuit in Fig. 9 without specifying any device restriction.

## V. LOGICAL QUBIT ARRANGEMENT

As mentioned in Sec. III B, mainly two schemes are used to perform quantum computations by lattice surgery: (i) original quantum circuits are computed by use of explicit logical Clifford gates and single-qubit rotation gates  $R_Z(\theta)$  and (ii) quantum circuits are converted to alternative forms comprising multi-qubit Pauli rotation gates and multi-qubit Pauli measurements beforehand and then the converted circuits are computed. The former case is suitable for calculating quantum circuits that have a high parallelism of the rotation gates since the rotation gate acts on only a single logical qubit and can be performed in parallel. A major drawback is that we must implement costly logical Clifford operations, which need  $2d$  or  $3d$  rounds of syndrome measurement and an additional ancilla patch. In the latter case, on the other hand, the number of physical qubits required

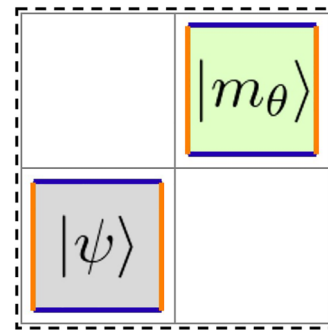


FIG. 15. Data unit that carries a data logical qubit and ancilla state for rotation gates. Actual data are carried by the logical patch labeled as  $|\psi\rangle$ , and the other patch labeled as  $|m_\theta\rangle$  is the ancilla state for the rotation gate.

is small because it does not need an ancilla region for the explicit Clifford gates. Instead, multi-qubit Pauli rotation gates are difficult to parallelize because of the anticommutation relations between them. Therefore, we can say that the latter scheme is suitable for calculating quantum circuits that contain the rotation gates sparsely or have a small parallelism of the rotation gates.

The arrangement of the logical qubit patches strongly depends on these schemes. Moreover, a trade-off relationship holds between the efficiency of the number of logical qubits and the execution time of the operations in general. To minimize unnecessary overheads, one needs to find an optimal arrangement for an input quantum circuit. Additionally, the input circuit possibly must be converted into a suitable form before the determination of the optimal arrangement. A logical qubit arrangement optimizer and circuit compiler are mandatory for maximizing the computational power of the STAR architecture, but this is beyond the scope of this paper. Here we illustrate only some typical arrangements of the logical qubit in both schemes. The development of a circuit compiler and arrangement optimizer for the STAR architecture will be one of the most important future studies.

In scheme (i), the ancilla state  $|m_\theta\rangle$  should be prepared in parallel for each data logical qubit to maximize the merit of the parallelism of rotation gates. Given that the gate teleportation circuit in Fig. 7 needs at least three logical patches because of the logical CNOT operation, it is better to group four logical patches as a unit, which carry a data logical qubit and an ancilla qubit for parallel rotation gates,



FIG. 16. Example of the qubit arrangement in scheme (i) with  $n = 6$ . Data units are arranged in a row.

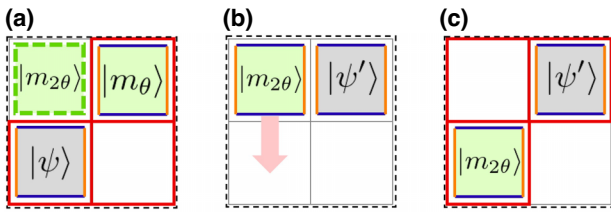


FIG. 17. RUS protocol within the data unit. (a) The gate teleportation circuit in Fig. 7 is computed with use of three patches, where  $|m_\theta\rangle$  and  $|\psi\rangle$  are the control qubit and the target qubit of the CNOT gate, respectively (red square). During the computation, we can prepare the ancilla state  $|m_{2\theta}\rangle$  for the next RUS step in the lower-right patch (dashed green square). If the output state  $|\psi'\rangle$  is not correct, (b) the prepared ancilla state moves to the next patch and then (c) the next step of the RUS protocol begins. We can also prepare  $|m_{4\theta}\rangle$  using the upper-left patch.

as shown in Fig. 15 (note that the same structure was proposed in Ref. [49]). Figure 16 exemplifies the arrangement based on this unit. This example requires at least  $4n$  logical patches to allocate  $n$  data logical qubits.

We can perform the RUS protocol within the unit as shown in Fig. 17. Because a single patch is free during the gate teleportation circuit, we can prepare the ancilla state needed for the next RUS step with a small overhead [dashed green square in Fig. 17(a)]. Additionally, the patch rotation after the logical  $H$  operation (Fig. 5) can be done with use of two patches in the unit. The logical CNOT operation can be directly applied to neighboring units with additional patch movements. Moreover, remote CNOT operations between distant units can be realized with use of the ancilla region. We show examples of those logical CNOT operations in Figs. 18 and 19. Note that one cannot perform some remote CNOT operations in parallel in the architecture in Fig. 16 because their ancillae cannot overlap. To minimize such conflicts, it is better to optimize the mapping of the quantum circuit.

The requirement of the  $4n$  logical patch discussed above may be somewhat large. Fortunately, we can use the multi-qubit Pauli measurement-based rotation circuit in Fig. 8 instead of the gate teleportation circuit in Fig. 7 to reduce the number of logical qubit patches. If we use the circuit in Fig. 8 for a single-qubit rotation, we must measure the

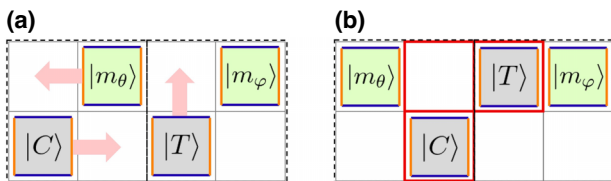


FIG. 18. Direct logical CNOT operation between neighboring data units. (a) The logical patches in these units first move to the appropriate positions and then (b) the CNOT operation is performed (red square).

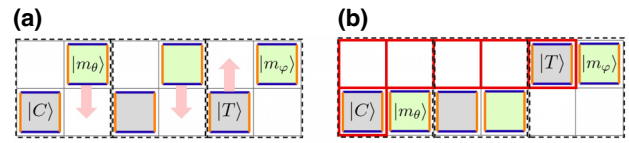


FIG. 19. Remote logical CNOT operation. (a) Logical patches move to the appropriate positions. (b) The CNOT operation is then performed with use of a long ancilla patch (red square).

$Z \otimes Z$  operator over the target logical state and the ancilla state, but it can be immediately performed by the  $Z$  boundary merging and splitting of these states [30]. Furthermore, unlike the circuit in Fig. 7, the target logical state does not move after the rotation; thus, no unnecessary overhead is needed to bring it back to the correct place. Therefore, in this case, we can perform the RUS protocol by a unit of three logical patches contacting each other on  $Z$  boundaries as shown in Fig. 20. During a single rotation performed by two of the three patches, the other patch can prepare the ancilla state for the next rotation. Because the target logical patch remains at the same position after the rotation, the next rotation step can immediately start. Figure 21(a) provides a typical arrangement in this case, which requires  $3n$  logical patches to allocate  $n$  data logical qubits. One can perform CNOT operations between logical patches and patch deformations in the same way as in the previous arrangement in Fig. 16 using the ancilla region. Note that if the overhead of the state injection does not need to be hidden, then the number of the logical patches can be further reduced to  $2n$ , as shown in Fig. 21(b). Although we consider mainly the case in which all data logical patches have identical unit structures, they can be mixed to minimize the computational overhead.

The prototypical logical qubit arrangement in scheme (ii) was proposed in detail in Ref. [30], and we only briefly

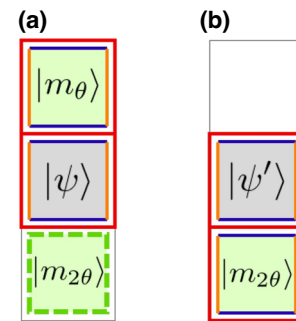


FIG. 20. RUS protocol based on the multi-qubit Pauli rotation circuit in Fig. 8. (a) A single  $Z$  rotation circuit is implemented by a  $Z \otimes Z$  measurement between the target state and the ancilla state (red square). During the operation, the other logical patch can be used to prepare the next ancilla state (dashed green square). (b) After the first RUS step, the output state  $|\psi'\rangle$  remains; thus, the next RUS step can start immediately.

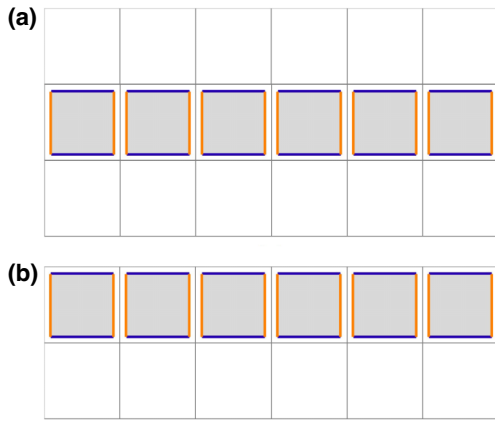


FIG. 21. Typical qubit arrangement in scheme (i) based on the circuit in Fig. 8 with  $n = 6$ . (a) Arrangement requiring  $3n$  patches. Each column of three patches constructs a unit to implement the RUS protocol, and they are arranged in a row. An ancilla region is used not only to parallelize  $Z$  rotations but also to perform other operations, such as logical CNOT operations and patch deformations. (b) Minimum arrangement requiring  $2n$  patches. Although this arrangement cannot hide the overhead of the state injection during the RUS protocol, it requires only  $2n$  patches to allocate  $n$  data logical qubits.

introduce it here. Since the early-FTQC device has a limited number of physical qubits, the compact block and the intermediate block [30] are suitable for our purpose. Figure 22 shows typical examples of each cases. In these examples, we assume that the multi-qubit Pauli rotation gates are sequentially performed. We allocate two additional patches for the injection of the ancilla state  $|m_\theta\rangle$  to hide the overhead of the state injection behind the execution time of a single RUS step by consuming and generating the ancilla states consecutively. The minimum construction using the compact and intermediate blocks

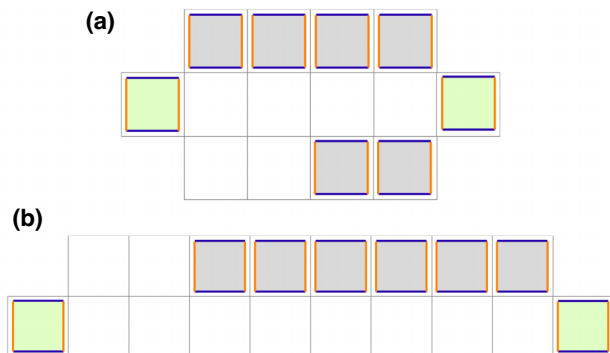


FIG. 22. Example of the qubit arrangement in scheme (ii), based on the data blocks discussed in Ref. [30]. (a) Compact block for  $n = 6$ . Gray patches represent data logical qubits. We allocate two patches to prepare the ancilla state (green) to reduce its additional overhead. (b) Intermediate block for  $n = 6$ .

requires  $1.5n + 5$  and  $2n + 6$  logical patches, respectively, to allocate  $n$  data logical qubits.

## VI. PERFORMANCE OF THE STAR ARCHITECTURE

To estimate the performance of our proposal quantitatively, we perform numerical simulations on the error correction of the surface code patch (related to the orange-square part in Fig. 1) and the ancilla state injection (related to the blue-square part in Fig. 1). In this section, we show the results of these simulations and estimate the computational resources available for early-FTQC devices on the basis of these results. We also briefly discuss possible applications of the STAR architecture on the basis of the estimation.

### A. Logical error probability of the rotated surface code patch

In the simulation of the error correction of the rotated surface code, we assume that Hadamard and CNOT gates, initialization to  $|0\rangle$ , and measurement in the  $Z$  basis are available as physical qubit operations, so that we use the depth-8 measurement circuit in Fig. 3. Noise processes are simulated by the circuit-level noise model, in which all operations on physical qubits suffer from errors: noisy qubit initialization and measurement flip to an orthogonal state with probability  $p$ , and noisy Hadamard and CNOT gates are simulated by ideal gate operations followed by the depolarizing noise channels

$$\mathcal{E}_{\text{single}}(\rho) = (1 - p)\rho + \frac{p}{3}(X\rho X + Y\rho Y + Z\rho Z), \quad (16)$$

and

$$\mathcal{E}_{\text{double}}(\rho) = \left(1 - \frac{16}{15}p\right)\rho + \frac{p}{15} \sum_{E \in \{I, X, Y, Z\}^{\otimes 2}} E\rho E, \quad (17)$$

respectively. Noisy identity gates are inserted whenever physical qubits are idle. We assume that all errors occur with common probability  $p$ . All measurement circuits are performed in parallel and performed  $d$  times to treat measurement errors. The last measurement round is performed ideally. For the decoding, we use PyMatching [50], an open-source PYTHON/C++ library, to implement the MWPM algorithm. We consider hook error edges in the construction of the decoder graph to decode errors correctly up to  $O(p^{\lfloor (d-1)/2 \rfloor})$ .

Logical error rates  $P_{L,i}$  ( $i = Z, X$ ) are determined by  $10^7$  Monte Carlo samples for each physical error rate  $p$ . For a resource estimation under a limited number of physical qubits available in the early-FTQC era, we consider small code distances of up to  $d = 9$  and a physical error rate of  $p \in [10^{-4}, 10^{-3}]$ . Figure 23 shows the resultant logical error rates obtained in our simulation.

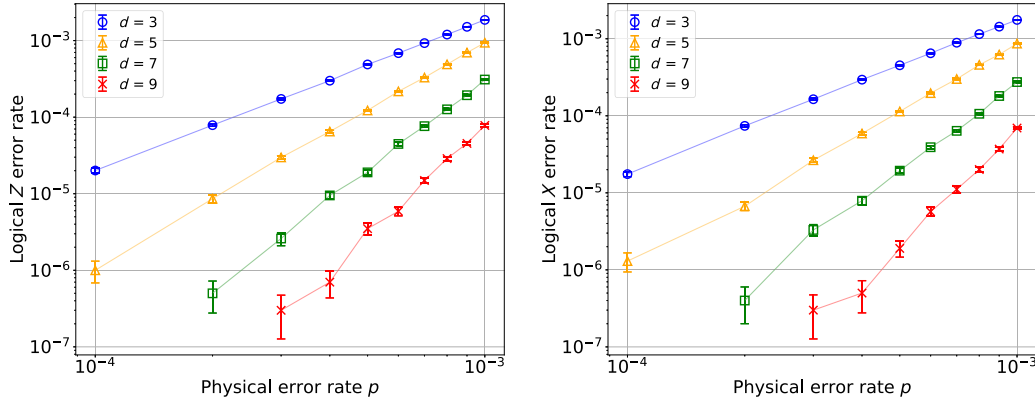


FIG. 23. Logical Z (left) and X (right) error rates of the rotated surface code patch. The error bars indicate  $\pm 1\sigma$  statistical errors estimated by a standard deviation of the Monte Carlo samples.

Because the data obtained seem to behave linearly in the log-log plot as seen in Fig. 23, we can expect that the  $p$  dependence of the logical error rate is well described by

$$P_{L,i}(p) = C_i \left( \frac{p}{p_{\text{th},i}} \right)^{\frac{d+1}{2}} \quad (i = Z, X), \quad (18)$$

where  $C_i$  and  $p_{\text{th},i}$  ( $i = Z, X$ ) are constant parameters, within the range of the physical error rate we consider. We determine those parameters by fittings using the numerical results for  $d = 7$  and 9. We show the optimized parameters and the behaviors of Eq. (18) with the optimized parameters in Table I and Fig. 24, respectively.

As expected, the numerical data are well fitted by the function of Eq. (18). We also observe that the threshold value  $p_{\text{th},X}$  obtained is larger than  $p_{\text{th},Z}$ , which is a well-known behavior resulting from the circuit asymmetry in Fig. 3 [31]. In the later resource estimation, we use Eq. (18) with the mean values of the optimized parameters (solid black lines in Fig. 24).

### B. Logical error probability of the ancilla state

In this study, we simulate the entire process of the state injection protocol discussed in Sec. IV B. Because the target patch after the expansion contains many physical qubits, the simulation is performed on the basis of the stabilizer formalism [51]. The stabilizer simulation does not support non-Clifford gates; thus, we take  $\theta = 0$ . We can justify this assumption as follows: First, in this setup, we cannot estimate the logical X error rate

TABLE I. Parameters of Eq. (18) optimized by the fitting. Statistical errors are estimated by the jackknife method with a bin size of  $10^4$ .

$C_Z$	$p_{\text{th},Z}$	$C_X$	$p_{\text{th},X}$
0.0679(76)	0.00385(10)	0.0819(97)	0.00416(12)

because the prepared state is now  $R_Z(0)|+\rangle = |+\rangle$ , on which the logical X operator does nothing. As discussed in the Appendix, however, the logical X error occurs only at  $O(p^2)$  and can be ignored if we are interested in the leading  $O(p)$  logical error rate. Second, the elimination of  $R_{Z_0Z_2}(\theta)$  ignores some error propagation processes occurring at  $O(p)$ , such as  $R_{Z_0Z_2}(\theta)X_0 = X_0R_{Z_0Z_2}(-2\theta) \times R_{Z_0Z_2}(\theta)$ , but those errors always occur together with a single Pauli X operator, which is detectable in the following postselection. Therefore, the elimination does not modify the leading  $O(p)$  logical error rate. We use the circuit-level noise model and the same assumption on the fundamental operations as in the simulation of the surface code patch discussed in Sec. VI A. Syndrome measurements are performed with the circuits in Figs. 3 and 11. The last syndrome measurement in the protocol is performed ideally. For the ancilla state that passes all the postselections, we measure the logical X operator and check whether the logical Z error occurs. We estimate the failure rate of the postselection and the logical Z error rate of the prepared ancilla state by counting these events for all Monte Carlo samples.

For the resource estimation, we consider the target surface code patch with  $d = 3, 5, 7$ , and 9 and a physical error rate of  $p \in [10^{-5}, 10^{-3}]$ . The failure rate and the logical Z error rate are estimated with use of  $8 \times 10^6$  ( $4 \times 10^6$ ) Monte Carlo samples with  $p \in [10^{-5}, 10^{-4}]$  ( $p \in [10^{-4}, 10^{-3}]$ ). Figures 25 and 26 show the numerical results for the logical Z error rate and the failure rate, respectively.

First, we discuss the resultant logical Z error rate. As discussed in the Appendix, under the circuit-level noise model, the leading-order behavior of the logical Z error rate is analytically given as  $P_{L,Z} = 2p/15 + O(p^2)$ . Our numerical result in Fig. 25 (top) shows that the logical Z error rate actually approaches the leading-order behavior when the physical error rate  $p$  becomes small. Moreover, from Fig. 25 (bottom), we confirm that the subleading

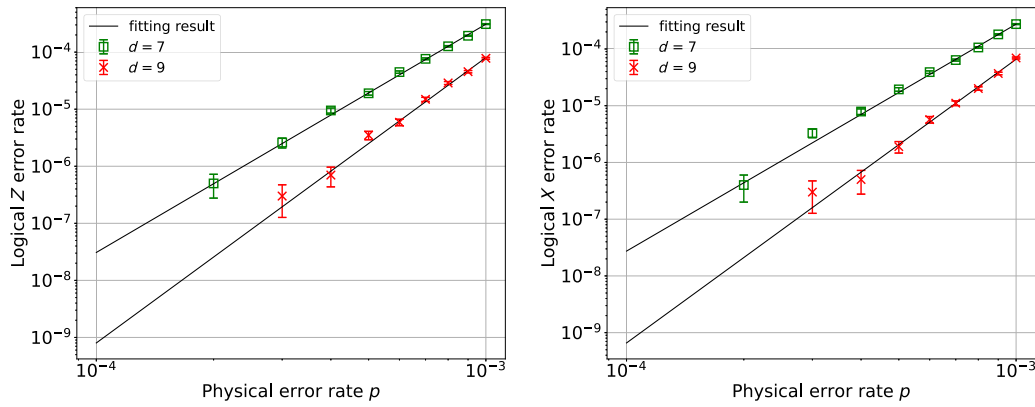


FIG. 24. Fitting results for the logical error rates with  $d = 7$  and  $9$ . The scaling of Eq. (18) with the mean values of the optimized parameters is shown as solid black lines.

contribution of  $O(p^2)$  can be ignored at a physical error rate below  $10^{-4}$ . Combining this observation with the fact that the logical  $X$  error rate in the ancilla state is  $O(p^2)$ , we can assume that the logical  $X$  error is also negligible below  $p = 10^{-4}$ . Next, we examine the failure rate of the postselection (Fig. 26). We observe that the failure rate increases

when the code distance becomes longer. This behavior is due to the longer distance code having more possible error configurations at  $O(p)$ , which are captured by the second postselection in our protocol. This large failure rate results in a large overhead for the state injection, and therefore we must reduce it using certain techniques. One solution is to repeat the protocol many times and reduce the effective failure rate. This can be achieved naively by parallel injection using multiple patches, although this approach requires an additional space cost. Alternatively, we may

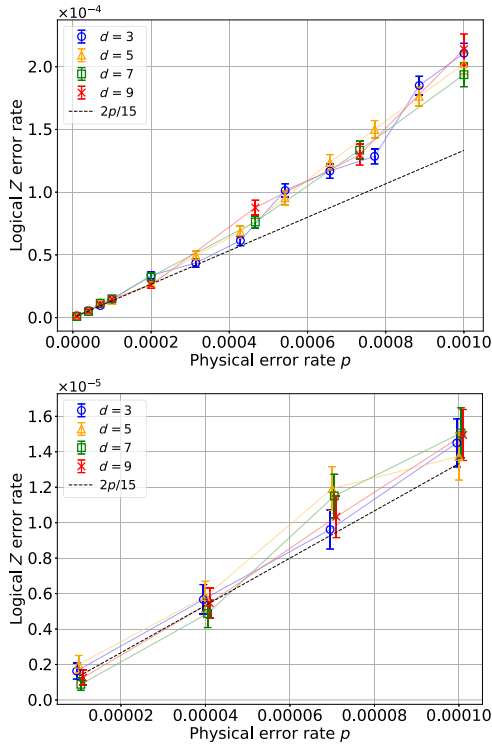


FIG. 25. Logical  $Z$  error rates of the ancilla state prepared in the surface code patches with  $d = 3, 5, 7,$  and  $9$ . The dashed line shows the leading-order behavior expected under the circuit-level noise model,  $P_{L,Z}(p) = 2p/15$ . The error bars indicate  $\pm 1\sigma$  statistical errors estimated by the standard deviation. Top:  $p$  dependence in the range  $p \in [10^{-5}, 10^{-3}]$ . Bottom: Enlarged view in the range  $p \in [10^{-5}, 10^{-4}]$ .

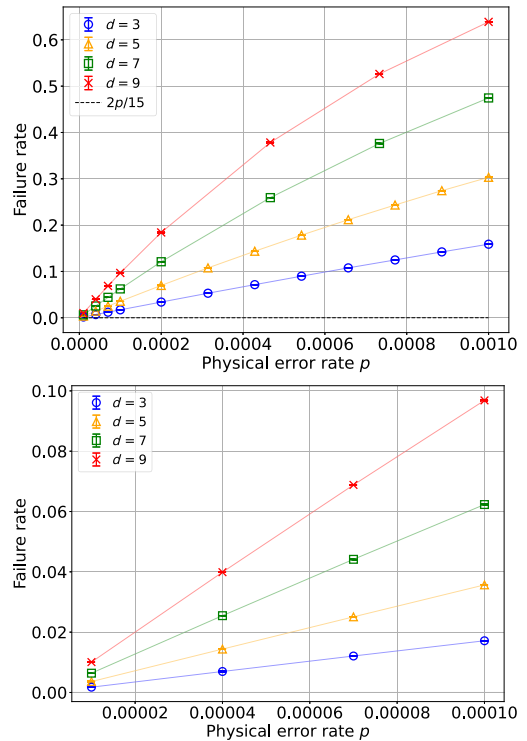


FIG. 26. Failure probability of the postselection in our protocol. Top:  $p$  dependence in the range  $p \in [10^{-5}, 10^{-3}]$ . Bottom: Enlarged view in the range  $p \in [10^{-5}, 10^{-4}]$ .

reduce the effective failure rate without any additional space cost by parallelizing the protocol along the “time direction.” To this end, one should first notice that the ancilla injection protocol can be performed within four rounds of the syndrome measurement (strictly, the total depth of the entire circuit can be  $2 + 7 + 6 + 2 \times 8 = 31$  when we maximally overlap the circuits in Figs. 3, 9, and 11). If we consider the RUS protocol shown in Fig. 17, we have a time interval of  $2d$  rounds of the syndrome measurement during a single RUS step, and then we can perform the state injection protocol roughly  $2d/4 = d/2$  times. With  $p = 10^{-4}$  and  $d = 9$ , we have a failure rate of approximately 10% as observed in Fig. 26 (right), but  $9/2 \approx 4$  runs of the protocol effectively reduce the failure rate to 0.01%.

### C. Resource estimation

In this section, we estimate a computational resource for early-FTQC devices on the basis of the results of the numerical simulations. Here we assume that a target device has  $N = 10^4$  physical qubits with physical error probability  $p = 10^{-4}$ .

Initially, we consider the number of logical qubits we can allocate. This number depends on the scheme for calculating a given circuit, as discussed in Sec. V. Since scheme (ii) in Sec. V is more efficient regarding the space cost, we consider it first. In the minimum construction using the compact block, we need at least  $1.5n + 5$  logical patches to allocate  $n$  data logical qubits. A single rotated surface code patch with code distance  $d$  needs approximately  $2d^2$  physical qubits, and therefore  $(1.5n + 5) \times 2d^2$  physical qubits are needed in total. If  $N = 10^4$ , we can allocate approximately 64 (approximately 37) data logical qubits for the  $d = 7$  ( $d = 9$ ) surface code patch in this setup. The same estimation for scheme (i) in Sec. V results in approximately 51 (approximately 30) data logical qubits for  $d = 7$  ( $d = 9$ ) if we use the smallest arrangement in Fig. 21(b).

Next, we estimate the number of gate operations we can perform on those data logical qubits. Regarding the logical Clifford operation, by our assuming that the error channel for the logical Clifford gate is an independent logical  $Z$  and  $X$  error channel, its logical error rate per  $d$  rounds of the syndrome measurement can be given as  $P_{L,\text{round}} \approx P_{L,Z} + P_{L,X}$ , where  $P_{L,Z}$  and  $P_{L,X}$  are the logical error rates obtained by the numerical simulation discussed in Sec. VI A. Using the fitting result for the logical error rates given in Table I, we can estimate  $P_{L,\text{round}}$  as approximately  $5.82 \times 10^{-8}$  (approximately  $1.46 \times 10^{-9}$ ) for the  $d = 7$  ( $d = 9$ ) logical patch with  $p = 10^{-4}$ . The available number of Clifford gates can be estimated as  $N_{\text{Clifford}} \approx 1/P_{L,\text{round}}$ , leading to  $N_{\text{Clifford}} \approx 1.72 \times 10^7$  ( $N_{\text{Clifford}} \approx 6.85 \times 10^8$ ) for the  $d = 7$  ( $d = 9$ ) logical patch with  $p = 10^{-4}$ . This number is sufficiently large, and  $d = 7$  or  $d = 9$

may be sufficient for most applications in the early-FTQC era. Note that, in practice, the estimated  $N_{\text{Clifford}}$  may be divided by a certain  $O(1)$  factor because some logical operations need more measurement rounds than  $d$  rounds (e.g., the explicit logical CNOT operation needs  $2d$  measurement rounds). However, this factor is not expected to change the estimation drastically, so we consider only the estimated value of  $N_{\text{Clifford}}$  above as representative. The available number of analog rotation gates can be estimated similarly. For  $p = 10^{-4}$ , we can ignore the  $O(p^2)$  contributions of the logical error rate as observed in the numerical simulation discussed in Sec. VI B. Therefore, a single rotation gate has a logical error rate of  $P_{L,\text{rotation}} = 2p/15 \approx 1.3 \times 10^{-5}$ . Since the actual rotation gate needs two RUS steps on average, we can estimate the available number of the rotation gates as  $N_{\text{rotation}} \approx 1/(2P_{L,\text{rotation}}) = 3.75 \times 10^4$ . As discussed in Sec. IV, the remnant error of  $N_{\text{rotation}}$  analog rotations can be mitigated by the probabilistic error cancellation with an additional sampling overhead  $\gamma^{2N_{\text{rotation}}} \approx e^{8P_{L,\text{rotation}}N_{\text{rotation}}} \approx 55$ . Note that since the sampling overhead of probabilistic error cancellation grows exponentially as approximately  $e^{8P_{L,\text{rotation}}N_{\text{rotation}}}$ , the number of rotation gates cannot increase drastically from the estimation above in general. In some cases where one does not need to remove the effect of noise completely, however, one can perform more rotation gates: for example, tasks that allow some margin of error and evaluation tasks for local observables that require only the error mitigation of the relevant noisy gates included in the causal cone [48].

In summary, assuming that  $N = 10^4$  and  $p = 10^{-4}$ , the STAR architecture based on the  $d = 7$  ( $d = 9$ ) surface code patch can perform quantum circuits using 64 (37) data qubits, which comprise  $1.72 \times 10^7$  ( $6.85 \times 10^8$ ) Clifford gates and  $3.75 \times 10^4$  arbitrary rotation gates. Notably, we can perform more than  $10^4$  arbitrary rotations and many error-corrected Clifford gates on 64 logical qubits with a relatively lenient requirement, namely,  $N = 10^4$  and  $p = 10^{-4}$ . Computations of this size cannot be simulated by classical supercomputers and state-of-the-art classical algorithms [52,53]. Even if we choose the  $d = 9$  case, which classical supercomputers can simulate, it still provides a useful test bed for small-scale FTQC experiments through the direct comparisons with classical simulations.

### D. Comparison with existing NISQ and FTQC architectures

To clarify an advantage of our architecture, we compare its performance with the performance of naive NISQ architectures and existing FTQC architectures.

A typical performance metric for the NISQ architecture is the quantum volume  $V_Q$  [54], which quantifies the typical size of a correctly executable circuit. To measure  $V_Q$ , we use a benchmark circuit with  $m$  qubits and  $d$  layers, as shown in Fig. 27. Each layer comprises a permutation

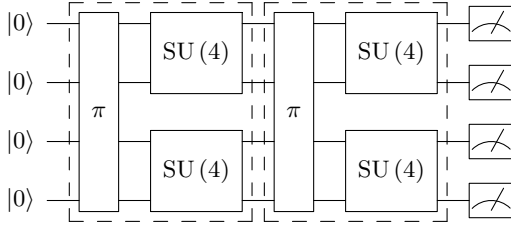


FIG. 27. Benchmark circuit used to measure the quantum volume with  $m = 4$  and  $d = 2$ . Operations grouped by dashed lines form a single layer.

(depicted by  $\pi$  in Fig. 27) and two-qubit unitary gates [depicted by  $SU(4)$  in Fig. 27]. The permutations and two-qubit unitaries in the benchmark circuit are randomly chosen and an output distribution is determined by our averaging over these randomly generated circuits. By considering the heavy output generation problem [54] on the output distribution, one can judge whether the circuit is implemented successfully: if the heavy output probability is more than two thirds, the circuit is considered reasonably executed; otherwise the computation failed.  $V_Q$  is defined by the maximum size  $m_{\max}$  of the square-shaped ( $m = d$ ) benchmark circuit that is successfully implemented,  $\log_2 V_Q = m_{\max}$ . To compare the performance between the STAR architecture and the naive NISQ architecture, we consider, for example, a quantum device comprising  $10^4$  physical qubits with a square grid connectivity and error rate  $p = 10^{-4}$ . According to Ref. [54], the size  $m(=d)$  of the square-shaped circuit that can be executed correctly satisfies

$$m^2(1.29\sqrt{m} - 0.78)p < 1, \quad (19)$$

when the single-qubit error rate is negligible versus the two-qubit error rate. The  $\sqrt{m}$  factor comes from the restriction of the square grid connectivity. By inserting the physical error rate  $p = 10^{-4}$  into Eq. (19), we can estimate the quantum volume as  $\log_2 V_Q = 37$  [55]. Regarding the STAR architecture, on the other hand, the permutations can be performed ideally and only the  $SU(4)$  gates suffer from errors. To quantify the error rate of the  $SU(4)$  gate, we use the fact that the  $SU(4)$  gate  $U$  can be decomposed as  $U = K_1 A(\alpha, \beta, \gamma) K_2$ , where  $K_i (i = 1, 2)$  are tensor products of single-qubit unitary gates and  $A(\alpha, \beta, \gamma) = \exp[i(\alpha X \otimes X + \beta Y \otimes Y + \gamma Z \otimes Z)]$  [54]. Each of the single-qubit unitary gates as well as  $A(\alpha, \beta, \gamma)$  is implemented by three analog rotations. Therefore, the entire  $SU(4)$  gate  $U$  can be implemented by  $3 \times 4 + 3 = 15$  analog rotations. Thus, the size  $m$  of the square-shaped circuit that can be executed correctly satisfies

$$m^2 \times (15/2) \times 2.6 \times 10^{-5} < 1, \quad (20)$$

and an allowed maximum size is  $m = 71$ . Since the STAR architecture can prepare 64 logical qubits on  $10^4$  physical qubits, it can achieve  $\log_2 V_Q = 64$ , which is substantially larger than that of the naive NISQ architecture. This result indicates that the STAR architecture reaches an advanced stage of quantum computation by successfully integrating error-corrected Clifford gates and noisy analog rotations when the physical error rate is sufficiently small. Note that this advantage gradually decreases if the physical error rate approaches the threshold value near  $p_{\text{th}} = 0.4\%$  because the suppression of the logical error of the Clifford gates becomes poor. Therefore, to overcome the difficulty associated with the NISQ architecture in the early-FTQC era, it is important to achieve a sufficiently small physical error rate below the threshold.

Next, we compare the STAR architecture with the existing FTQC architectures. Presently, the best space-efficient FTQC architecture is the one reported in Ref. [30] and was introduced in Sec. III B. To compare it with the STAR architecture, we first note that the STAR architecture can perform an analog multi-qubit Pauli rotation with a logical error rate  $\epsilon$  of  $2.6 \times 10^{-5}$  within 18 clocks on average (here we call an execution time of  $d$  rounds of the syndrome measurements “1 clock”) since the compact block consumes one magic state in 9 clocks in the worst case [30]. In the following discussion, we estimate the resources needed to achieve the same performance of the analog multi-qubit Pauli rotation using the existing FTQC architecture. On this basis, we estimate the available computational power of the existing FTQC architecture under the restriction of a quantum device that comprises  $10^4$  physical qubits with a square grid connectivity and error rate  $p = 10^{-4}$ .

The Clifford gates are implemented by lattice surgery in both architectures; thus, we fix the code distance to  $d = 7$  to make their performances even. Using the existing state injection protocol proposed in Ref. [40], a bare magic state is obtained with error rate  $46p/15 \approx 3.1 \times 10^{-4}$ . Because we must implement tens or hundreds of  $T$  gates to perform analog rotation, this accuracy is insufficient and we must distill the magic state. Using the typical 15-to-1 distillation protocol [30], we obtain a clean magic state with accuracy of  $35 \times (3.1 \times 10^{-4})^3 \approx 1.0 \times 10^{-9}$ , the precision of which is sufficient for our purpose. Therefore, to achieve an accuracy  $\epsilon$  of  $2.6 \times 10^{-5}$  for a single analog rotation, the remaining task is to decompose the analog rotation gate into the sequence of Clifford gates and  $T$  gates with accuracy  $\epsilon$ . According to the state-of-the-art algorithm [28], the required number of  $T$  gates to achieve approximation accuracy  $\delta$  roughly behaves as  $N \approx 3 \log_2(1/\delta)$  [56]. By substituting  $\delta = \epsilon = 2.6 \times 10^{-5}$ , we obtain  $N \approx 46$ . Thus, the existing FTQC architecture needs at least 46 clocks to implement the analog rotation, which is approximately 2.6-fold slower than the STAR architecture. This clearly show



that the direct analog rotation is inherently advantageous for the fast operation.

Furthermore, in the FTQC architecture, a trade-off relationship holds between the execution time of the non-Clifford gate and the number of physical qubits because of the slow supply of the magic state by a single distillation block. First, we consider the case in which a single  $T$  gate takes 1 clock to implement. This case is related to the fast block in Ref. [30], which needs  $2n + \sqrt{8n} + 1$  patches to allocate  $n$  logical qubits. In addition to the data block, we need a magic state factory that can supply one magic state per clock. According to Ref. [30], a single 15-to-1 distillation block can supply one magic state in 11 clocks using at least 11 patches. To achieve the required magic state supply rate, we must implement 11 distillation protocols in parallel; thus, at least  $11 \times 11 = 121$  patches are required. If we consider the  $d = 7$  surface code, we can prepare at most  $10^4 / (2 \times 7^2) \approx 102$  patches and cannot even allocate the magic state factory. To reduce the physical qubit overhead, we can use other data blocks, such as an intermediate block or a compact block, at the expense of operation speed. The intermediate block (compact block) takes 5 clocks (9 clocks) to implement a single  $T$  gate [30], and the number of distillation blocks in the factory can be reduced to three (two), respectively. The magic state factory requires  $11 \times 3 = 33$  ( $11 \times 2 = 22$ ) patches; thus, we can allocate logical qubits by using the remaining  $102 - 33 = 69$  ( $102 - 22 = 80$ ) patches. Because the intermediate block (compact block) requires  $2n + 4$  ( $1.5n + 3$ ) patches to allocate  $n$  logical qubits, we can allocate  $n = 32$  ( $n = 51$ ) logical qubits. The intermediate block architecture can easily be simulated by existing classical supercomputers, and it is difficult to provide useful quantum advantages with the  $10^4$  physical qubit device. Although the compact block architecture enters a classically intractable region, its available logical qubits are fewer than in the STAR architecture ( $n = 64$ ) and its execution time is 23 times slower than that of the STAR architecture. Therefore, the STAR architecture is advantageous in terms of the logical qubit number and execution time. We summarize the performance of the existing FTQC architecture and the STAR architecture in Table II.

Although the STAR architecture always has advantages against the FTQC architecture in terms of the execution speed and the number of logical qubits, the executable number of rotation gates is restricted by the inverse of the physical error rate. Therefore, the development of the low-error physical qubit is important again. In addition to hardware improvement, algorithmic improvement is also mandatory to achieve useful applications with a small number of rotation gates. If fully fledged FTQC becomes available in the future, it will be necessary to use the STAR architecture and the FTQC architecture differently depending on the application. For example, quantum circuits in which the number of arbitrary rotations is not so large can

TABLE II. Comparison between the STAR architecture and the existing FTQC architecture [30] on an early-FTQC device that consists of  $10^4$  physical qubits with a square grid connectivity and error rate  $p = 10^{-4}$ .

Architecture	Number of logical qubits	Non-Clifford gate execution time (clocks)
STAR compact ( $d = 7$ )	64	18
FTQC fast ( $d = 7$ )	0	46
FTQC intermediate ( $d = 7$ )	32	230
FTQC compact ( $d = 7$ )	51	414

be efficiently calculated with use of the STAR architecture, while in the case of quantum circuits that comprise an extremely large number of gate operations, the calculation is performed with high accuracy with use of the FTQC architecture.

In summary, the STAR architecture can outperform a naive application of the NISQ architecture and the FTQC architecture to the early-FTQC device. Its advantage versus the NISQ architecture is attributed mainly to the error correction of the Clifford gates. Furthermore, compared with the existing FTQC architecture, we can say that the combination of the direct implementation of the analog rotation gate and the careful state injection protocol makes the STAR architecture faster and smaller with minimum compromising accuracy.

### E. Possible applications

Finally, we briefly discuss possible applications of the STAR architecture. Here we show only some naive examples and typical calculation sizes based on the resource estimation. A detailed examination of the useful applications is an important future issue.

One promising application of the STAR architecture is a quantum many-body simulation because the time-evolution operator can be implemented easily by analog rotation gates. For example, we consider a two-dimensional Hubbard model with  $N = N_x N_y$  sites. By using the Jordan-Wigner transformation and snake-shaped indexing [57], we can write the Hamiltonian in terms of Pauli operators as

$$H = -\frac{t}{2} \sum_{\langle ij \rangle} (X_i X_j + Y_i Y_j) Z_{ij}^{\leftrightarrow} + \frac{U}{4} \sum_{i=0}^{N-1} Z_i Z_{2N-1-i} - \frac{U}{4} \sum_{i=0}^{2N-1} Z_i, \quad (21)$$

$$Z_{ij}^{\leftrightarrow} = \prod_{k=i+1}^{j-1} Z_k, \quad (22)$$

where  $t$  and  $U$  are parameters of the system and  $P_i$  ( $P = X, Y, Z$ ) are Pauli operators acting on the  $i$ th degree of freedom. The notation  $\langle i, j \rangle$  indicates adjacent sites in the  $N_x \times N_y$  lattice. The first term in Eq. (22) contains long Pauli chains that require many CNOT operations in the Trotter decomposition. A notable merit of our architecture is that we do not need to be afraid of these Pauli chains thanks to the error-corrected CNOT gates. This is generally true for other Hamiltonians containing long Pauli chains.

We estimate how many Trotter steps our architecture can perform in the two-dimensional Hubbard model simulation. Equation (22) consists of  $8N + N + 2N = 11N$  terms; thus, its time evolution of a single Trotter step requires  $11N$  arbitrary rotation gates. If we choose the  $d = 7$  ( $d = 9$ ) architecture and fully allocate data logical qubits for  $N$  sites, we can simulate  $N = 64/2 = 32$  ( $N = 37/2 = 18$ ) sites. The actual number of rotation gates per Trotter step is  $11 \times 32 = 352$  ( $11 \times 18 = 198$ ). Therefore, we can simulate real-time dynamics with  $3.75 \times 10^4/352 \approx 107$  ( $3.75 \times 10^4/198 \approx 189$ ) Trotter steps for this system.

With use of the iterative phase estimation [58] or recent resource-efficient algorithms for the early-FTQC era [45, 59], phase estimation for unitary operators acting on 63 or 37 qubits can be achieved. This phase estimation can be applied to determine the ground state energy of the quantum system reachable in the STAR architecture if it allows sufficiently large Trotter steps. In this context, discretization errors must be minimized in the Trotterization. We may, for example, use local variational quantum compiling [60] for this purpose.

Another promising application is the QAOA [61] for solving binary optimization problems. For example, we consider the maximum-cut problem for a graph with  $N$  nodes. The problem Hamiltonian is given as

$$H_C = -\frac{1}{2} \sum_{i \neq j} (1 - Z_i Z_j). \quad (23)$$

To obtain the ground state of  $H_C$ , we consider the QAOA ansatz state

$$|\gamma, \beta\rangle = e^{-i\beta_{p-1}H_B} e^{-i\gamma_{p-1}H_C} \dots e^{-i\beta_0 H_B} e^{-i\gamma_0 H_C} \times H^{\otimes N} |0\rangle^{\otimes N}, \quad (24)$$

where

$$H_B = \sum_{j=0}^{N-1} X_j \quad (25)$$

and  $\gamma = (\gamma_0, \dots, \gamma_{p-1})$  and  $\beta = (\beta_0, \dots, \beta_{p-1})$  are optimization parameters. They are optimized to minimize an expectation value  $\langle \gamma, \beta | H_C | \gamma, \beta \rangle$ . The ansatz state contains  $p(N + ((N(N-1))/2))$  arbitrary rotations in total. If we choose the  $d = 7$  ( $d = 9$ ) architecture and set  $N = 64$

( $N = 37$ ), we can take the depth of the ansatz as  $p = 3.75 \times 10^4/2080 \approx 18$  ( $3.75 \times 10^4/703 \approx 53$ ). Note that higher-order binary optimization problems can be directly solved in the STAR architecture without any reduction to quadratic unconstrained binary optimization problems, because the Clifford gates are almost error-free.

## VII. CONCLUSION

In this work, we propose a quantum computing architecture suitable for early-FTQC devices, the STAR architecture. In the STAR architecture, universal quantum computation is achieved by arbitrary rotation gates and error-corrected Clifford gates. Analog rotation gates are realized by the RUS protocol with appropriate ancilla states. To reduce logical errors of the rotation gates, we carefully design the ancilla state injection protocol by combining the  $[[4, 1, 1, 2]]$  subsystem code and postselection. Thus, our rotation gate achieves a small logical error rate of  $P_L = 2p/15 + O(p^2)$  under the circuit-level noise model, which is verified numerically. Clifford operations are performed by the standard lattice surgery protocol based on the rotated surface code, and we illustrate typical logical qubit arrangements. We also perform a numerical simulation on the surface code patch and determine a scaling behavior of the logical error rate. Finally, we estimate an available computational resource in the STAR architecture under the assumption of typical early-FTQC devices, where  $N = 10^4$  physical qubits can operate with a gate fidelity  $p$  of  $10^{-4}$ . According to this estimate, we can apply  $3.75 \times 10^4$  arbitrary rotation gates and  $1.72 \times 10^7$  Clifford gates on 64 logical qubits encoded in the  $d = 7$  rotated planar surface code. Classical computers cannot emulate such computations. Furthermore, the STAR architecture can surpass the naive NISQ architecture and the existing FTQC architecture. The STAR architecture may apply to some useful applications such as quantum many-body simulation, phase estimation, and the QAOA.

Some topics are not addressed in this paper. We summarize these topics to envision future directions of our proposal.

- (i) Optimization of the logical qubit arrangement and input quantum circuit. Regarding the logical qubit arrangement, we illustrate only some prototypical arrangements in this paper. In practical applications, however, the number of logical operations that can be performed simultaneously must be maximized to reduce computational time. Such parallelization highly depends on the structure of the quantum circuit we want to perform. Some existing ideas, such as introducing other “synthesis qubits” for parallel implementation of the diagonal non-Clifford operations [62], may be useful in this context. Developing a clever compiler that decomposes the input circuit

into the sequence of Clifford gates and  $R_Z(\theta)$  and determines the patch arrangement maximizing the gate parallelism based on the decomposed circuit should be future work.

- (ii) More concrete discussion of the possible applications of the STAR architecture. In this study, we only briefly mentioned some prototypical quantum computations that can be performed on the STAR architecture. By combining clever resource-reduction techniques such as local variational quantum compiling, the STAR architecture may give us some useful applications at the earlier stage of a large-scale quantum device.
- (iii) Improvements in our injection protocol. Although our injection protocol minimizes the remaining logical error, it still occur at  $O(p)$ . To perform more interesting computations, we must further reduce the logical error rate on the analog rotation. However, the distillation protocol on the arbitrary rotation ancilla state has not been known until now, and the task of reducing its logical error rate to  $O(p^2)$  is challenging. Developing a more sophisticated state injection and distillation protocol for the early-FTQC era is an interesting future direction. In this context, we note that another injection protocol was recently suggested by Choi *et al.* [63], which can achieve high-accuracy rotations for small angles. Since our injection protocol can achieve high accuracy for large angles, a combination of both protocols may improve the entire performance.

We hope that our proposal and the corresponding development of quantum algorithms will result in new insights into realizing practical quantum computers in the future.

### ACKNOWLEDGMENTS

We thank Jun Fujisaki and Mitsuki Katsuda for fruitful discussions. K.F. is supported by MEXT Quantum Leap Flagship Program Grants No. JPMXS0118067394 and No. JPMXS0120319794, JST COI-NEXT Grant No. JPMJPF2014, and JST Moonshot R&D Grant No. JPMJMS2061.

### APPENDIX: LEADING-ORDER LOGICAL ERROR PROBABILITY OF $|m_\theta\rangle_L$ UNDER THE CIRCUIT-LEVEL NOISE MODEL

In this appendix, we discuss a leading-order logical error probability of the ancilla state prepared in our protocol under the circuit-level noise model. We consider the same noise model as discussed in Sec. VI A: all physical operations suffer from error, which occurs with common probability  $p$ .

First, we consider logical errors occurring in the ancilla state injection circuit in Fig. 9. In this circuit, there can be

weight-2 logical errors with probability proportional to  $p$ , due to the error propagation of the CNOT operations and the two-qubit depolarizing channels. The error propagation of the CNOT operations causes weight-2 errors, such as

$$X_0X_1, \quad X_2X_3, \quad Z_0Z_1, \quad Z_2Z_3. \quad (\text{A1})$$

Note that errors on qubits 2 and 3 are identical to those on qubits 0 and 1 up to stabilizer operators. Since  $Z_0Z_1(Z_2Z_3)$  is the logical  $Z$  error on the gauge DOF, it is not critical for state injection. The other one,  $X_0X_1(X_2X_3)$ , is the logical  $X$  error on the logical qubit, but it does not destroy the logical state since the state is  $|+\rangle_L$  at that moment. The single  $Y$  errors before the CNOT operation also lead to other weight-2 errors, e.g.,  $Y_0 \otimes Z_1$  or  $X_0 \otimes Y_1$ , but they are detected as single  $X_0$  or  $Z_1$  errors and are removed by the postselection. In the same discussion, weight-2 errors produced by the two-qubit depolarizing channel in the noisy CNOT operation do not destroy the logical state. Regarding the noisy  $R_{Z_0Z_2}(\theta)$ , however, there are weight-2 errors that destroy the logical state. The two-qubit depolarizing channel after the ideal  $R_{Z_0Z_2}(\theta)$  provides weight-2 errors, such as

$$Z_0Z_2, \quad X_0X_2. \quad (\text{A2})$$

In these examples,  $X_0X_2$  is the logical  $X$  error on the gauge DOF and does not affect the logical state. On the other hand,  $Z_0Z_2$  is the logical  $Z$  error and changes the ancilla state to an orthogonal state as follows:

$$\begin{aligned} Z_L |m_\theta\rangle_L &= Z_L (e^{-i\theta/2}|0\rangle_L + e^{i\theta/2}|1\rangle_L) \\ &= e^{-i\theta/2}|0\rangle_L - e^{i\theta/2}|1\rangle_L \equiv |\bar{m}_\theta\rangle_L. \end{aligned} \quad (\text{A3})$$

The weight-2 errors that cause the logical  $Z$  error in the two-qubit depolarizing channel are  $Z_0Z_2$  and  $Y_0Y_2$ . Therefore, its probability of occurring is  $2p/15$ . There are other weight-2 errors, such as  $Y_0X_2$ , but they are detectable as a single-qubit error. In addition, there are some  $O(p)$  error propagation processes that cause an inverse rotation (equivalent to logical  $X$  error), such as  $R_{Z_0Z_2}(\theta)X_0 = X_0R_{Z_0Z_2}(-2\theta) \times R_{Z_0Z_2}(\theta)$ , but those errors are detectable as a single-qubit error. In summary, the logical error rate of the ancilla state generated by the circuit in Fig. 9 behaves as

$$P_{Z_L}(p) = 2p/15 + O(p^2), \quad (\text{A4})$$

$$P_{X_L}(p) = O(p^2). \quad (\text{A5})$$

Next, we examine possible logical errors during the syndrome measurement circuits in Figs. 3 and 11. In the circuit in Fig. 11, a single error on the measurement qubit leads at most to weight-1 errors on physical qubits, and they do not lead to undetectable logical errors. One possibility to realize  $O(p)$  weight-2 errors is the two-qubit

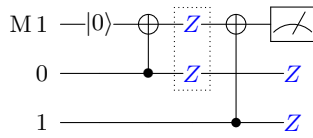


FIG. 28. Example of an  $O(p)$  weight-2 error in the measurement circuit of the  $[[4, 1, 1, 2]]$  subsystem code. The  $Z \otimes Z$  error occurring in the first CNOT gate (grouped by a dotted line) propagates to physical qubits and forms a weight-2 error.

error occurring in the first CNOT operation with error propagation through the second CNOT operation, as shown in Fig. 28. However, this weight-2 error acting on physical qubits is the gauge operator and is absorbed by the gauge DOF. Thus, the measurement circuit in Fig. 11 does not amplify the leading-order logical error rate of the ancilla state, Eqs. (A4) and (A5). Regarding the syndrome measurement circuit in Fig. 3, weight-2 errors can occur but they are orthogonal to the logical operators due to the order of CNOT operation. There is no other possibility to generate logical errors at  $O(p)$ . Therefore, the circuit in Fig. 3 does not amplify the leading-order behavior of the logical error rate as well.

In conclusion, the logical error rate of the ancilla state  $|m_\theta\rangle$  prepared in our protocol is dominated by the ancilla state injection circuit (Fig. 9) and behaves as  $P_L = 2p/15 + O(p^2)$ .

- [1] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Rev.* **41**, 303 (1999).
- [2] D. S. Abrams and S. Lloyd, Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors, *Phys. Rev. Lett.* **83**, 5162 (1999).
- [3] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Simulated quantum computation of molecular energies, *Science* **309**, 1704 (2005).
- [4] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* **103**, 150502 (2009).
- [5] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
- [6] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, Quantum supremacy using a programmable superconducting processor, *Nature* **574**, 505 (2019).
- [7] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, *et al.*, Quantum computational advantage using photons, *Science* **370**, 1460 (2020).
- [8] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, *et al.*, Strong quantum computational advantage using a superconducting quantum processor, *Phys. Rev. Lett.* **127**, 180501 (2021).
- [9] Q. Zhu, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, M. Gong, *et al.*, Quantum computational advantage via 60-qubit 24-cycle random circuit sampling, *Sci. Bull.* **67**, 240 (2022).
- [10] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins, *et al.*, Quantum computational advantage with a programmable photonic processor, *Nature* **606**, 75 (2022).
- [11] Y. Liu, X. Liu, F. Li, H. Fu, Y. Yang, J. Song, P. Zhao, Z. Wang, D. Peng, H. Chen, *et al.*, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Association for Computing Machinery, New York, 2021), p. 1.
- [12] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, *et al.*, Variational quantum algorithms, *Nat. Rev. Phys.* **3**, 625 (2021).
- [13] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, Hybrid quantum-classical algorithms and quantum error mitigation, *J. Phys. Soc. Jpn.* **90**, 032001 (2021).
- [14] S. Brandhofer, S. Devitt, T. Wellens, and I. Polian, in *2021 IEEE 39th VLSI Test Symposium (VTS)* (IEEE, 2021), p. 1.
- [15] Y. Zhao, Y. Ye, H.-L. Huang, Y. Zhang, D. Wu, H. Guan, Q. Zhu, Z. Wei, T. He, S. Cao, *et al.*, Realization of an error-correcting surface code with superconducting qubits, *Phys. Rev. Lett.* **129**, 030501 (2022).
- [16] S. Krinner, N. Lacroix, A. Remm, A. Di Paolo, E. Genois, C. Leroux, C. Hellings, S. Lazar, F. Swiadek, J. Herrmann, *et al.*, Realizing repeated quantum error correction in a distance-three surface code, *Nature* **605**, 669 (2022).
- [17] R. Acharya, I. Aleiner, R. Allen, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, J. Atalaya, R. Babbush, *et al.*, Suppressing quantum errors by scaling a surface code logical qubit, *Nature* **614**, 676 (2023).
- [18] S. Bravyi and J. Haah, Magic-state distillation with low overhead, *Phys. Rev. A* **86**, 052329 (2012).
- [19] C. Gidney and M. Ekerå, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits, *Quantum* **5**, 433 (2021).
- [20] N. Yoshioka, T. Okubo, Y. Suzuki, Y. Koizumi, and W. Mizukami, Hunting for quantum-classical crossover in condensed matter problems, *ArXiv:2210.14109* (2022).
- [21] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, Elucidating reaction mechanisms on quantum computers, *Proc. Natl. Acad. Sci.* **114**, 7555 (2017).
- [22] J. J. Goings, A. White, J. Lee, C. S. Tautermann, M. Degroote, C. Gidney, T. Shiozaki, R. Babbush, and N. C. Rubin, Reliably assessing the electronic structure of cytochrome P450 on today's classical computers and tomorrow's quantum computers, *ArXiv:2202.01244* (2022).
- [23] Y. Suzuki, S. Endo, K. Fujii, and Y. Tokunaga, Quantum error mitigation as a universal error reduction technique: Applications from the NISQ to the fault-tolerant quantum computing eras, *PRX Quantum* **3**, 010345 (2022).
- [24] C. Piveteau, D. Sutter, S. Bravyi, J. M. Gambetta, and K. Temme, Error mitigation for universal gates on encoded qubits, *Phys. Rev. Lett.* **127**, 200505 (2021).
- [25] K. Fujii, *Quantum Computation with Topological Codes: From Qubit to Topological Fault-Tolerance* (Springer, Singapore, 2015), Vol. 8.

- [26] B. Eastin and E. Knill, Restrictions on transversal encoded quantum gate sets, *Phys. Rev. Lett.* **102**, 110502 (2009).
- [27] X. Zhou, D. W. Leung, and I. L. Chuang, Methodology for quantum logic gate construction, *Phys. Rev. A* **62**, 052316 (2000).
- [28] N. J. Ross and P. Selinger, Optimal ancilla-free Clifford+T approximation of z-rotations., *Quantum Inf. Comput.* **16**, 901 (2016).
- [29] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, *New J. Phys.* **14**, 123011 (2012).
- [30] D. Litinski, A game of surface codes: Large-scale quantum computing with lattice surgery, *Quantum* **3**, 128 (2019).
- [31] A. M. Stephens, Fault-tolerant thresholds for quantum error correction with the surface code, *Phys. Rev. A* **89**, 022321 (2014).
- [32] D. Litinski and F. v. Oppen, Lattice surgery with a twist: Simplifying Clifford gates of surface codes, *Quantum* **2**, 62 (2018).
- [33] J. Edmonds, Paths, trees, and flowers, *Can. J. Math.* **17**, 449 (1965).
- [34] N. Delfosse and N. H. Nickerson, Almost-linear time decoding algorithm for topological codes, *Quantum* **5**, 595 (2021).
- [35] G. Duclos-Cianci and D. Poulin, Fast decoders for topological quantum codes, *Phys. Rev. Lett.* **104**, 050504 (2010).
- [36] K. Fujii, M. Negoro, N. Imoto, and M. Kitagawa, Measurement-free topological protection using dissipative feedback, *Phys. Rev. X* **4**, 041039 (2014).
- [37] J. Fujisaki, H. Oshima, S. Sato, and K. Fujii, Practical and scalable decoder for topological quantum error correction with an Ising machine, *Phys. Rev. Res.* **4**, 043086 (2022).
- [38] C. N. Self, M. Benedetti, and D. Amaro, Protecting expressive circuits with a quantum error detection code, *ArXiv:2211.06703* (2022).
- [39] D. Bacon, Operator quantum error-correcting subsystems for self-correcting quantum memories, *Phys. Rev. A* **73**, 012340 (2006).
- [40] Y. Li, A magic state's fidelity can be superior to the operations that created it, *New J. Phys.* **17**, 023037 (2015).
- [41] L. Lao and B. Criger, in *Proceedings of the 19th ACM International Conference on Computing Frontiers*, CF '22 (Association for Computing Machinery, New York, NY, USA, 2022), p. 113.
- [42] 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:2211.10046* (2022).
- [43] 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).
- [44] C. Gidney, Cleaner magic states with hook injection, *ArXiv:2302.12292* (2023).
- [45] Z. Ding and L. Lin, Even shorter quantum circuit for phase estimation on early fault-tolerant quantum computers with applications to ground-state energy estimation, *ArXiv:2211.11973* (2022).
- [46] K. Temme, S. Bravyi, and J. M. Gambetta, Error mitigation for short-depth quantum circuits, *Phys. Rev. Lett.* **119**, 180509 (2017).
- [47] S. Endo, S. C. Benjamin, and Y. Li, Practical quantum error mitigation for near-future applications, *Phys. Rev. X* **8**, 031027 (2018).
- [48] M. C. Tran, K. Sharma, and K. Temme, Locality and error mitigation of quantum circuits, *ArXiv:2303.06496* (2023).
- [49] L. Lao, B. van Wee, I. Ashraf, J. van Someren, N. Khammassi, K. Bertels, and C. G. Almudever, Mapping of lattice surgery-based quantum circuits on surface code architectures, *Quantum Sci. Technol.* **4**, 015005 (2018).
- [50] O. Higgott and C. Gidney, PyMatching v2, <https://github.com/oscarhiggott/PyMatching> (2022).
- [51] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, *Phys. Rev. A* **70**, 052328 (2004).
- [52] S. Bravyi and D. Gosset, Improved classical simulation of quantum circuits dominated by Clifford gates, *Phys. Rev. Lett.* **116**, 250501 (2016).
- [53] H. Pashayan, O. Reardon-Smith, K. Korzekwa, and S. D. Bartlett, Fast estimation of outcome probabilities for quantum circuits, *PRX Quantum* **3**, 020361 (2022).
- [54] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, Validating quantum computers using randomized model circuits, *Phys. Rev. A* **100**, 032328 (2019).
- [55] This value may be overly large since it ignores single-qubit errors. By adding the contribution of the single-qubit errors in the transpiled circuit of SU(4) gate obtained by QISKIT, we roughly obtain  $m = 26$ .
- [56] Precisely, Ross and Selinger [28] conjectured that  $N = K + 3 \log_2(1/\delta)$  with some constant  $K$  if  $\delta \rightarrow 0$ . Our estimation may be rough since it ignores the constant term and  $\delta = O(10^{-5})$  is not small. However, a similar algorithm proposed in Ref. [64] gives  $N = 3.067 \log_2(1/\delta) - 4.322$  around  $\delta \in 10^{-2}, 10^{-10}$ ; therefore, we expect that our estimation is reasonable up to  $O(1)$  constant deviation.
- [57] C. Cade, L. Mineh, A. Montanaro, and S. Stanisic, Strategies for solving the Fermi-Hubbard model on near-term quantum computers, *Phys. Rev. B* **102**, 235122 (2020).
- [58] M. Dobšiček, G. Johansson, V. Shumeiko, and G. Wendin, Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark, *Phys. Rev. A* **76**, 030306(R) (2007).
- [59] R. Kshirsagar, A. Katabarwa, and P. D. Johnson, On proving the robustness of algorithms for early fault-tolerant quantum computers, *ArXiv:2209.11322* (2022).
- [60] K. Mizuta, Y. O. Nakagawa, K. Mitarai, and K. Fujii, Local variational quantum compilation of large-scale Hamiltonian dynamics, *PRX Quantum* **3**, 040302 (2022).
- [61] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, *ArXiv:1411.4028* (2014).
- [62] M. E. Beverland, P. Murali, M. Troyer, K. M. Svore, T. Hoefler, V. Kliuchnikov, G. H. Low, M. Soeken, A. Sundaram, and A. Vaschillo, Assessing requirements to scale to practical quantum advantage, *ArXiv:2211.07629* (2022).
- [63] H. Choi, F. T. Chong, D. Englund, and Y. Ding, Fault tolerant non-Clifford state preparation for arbitrary rotations, *ArXiv:2303.17380* (2023).
- [64] V. Kliuchnikov, D. Maslov, and M. Mosca, Practical approximation of single-qubit unitaries by single-qubit quantum Clifford and T circuits, *IEEE Trans. Comput.* **65**, 161 (2016).