mathematics

IMPACT FACTOR 2.2

CITESCORE 4.6

Article

About Implementation of Magic State Injection in Heavy-Hexagon Structure

Hansol Kim, Wonjae Choi and Younghun Kwon

Special Issue

Recent Advances in Quantum Information and Quantum Computing

Edited by

Dr. Younghun Kwon

MDPI

https://doi.org/10.3390/math13233874

mathematics

MDPI

Article

About Implementation of Magic State Injection in Heavy-Hexagon Structure

Hansol Kim, Wonjae Choi and Younghun Kwon *

Department of Applied Physics, Hanyang University (ERICA), Ansan 15588, Republic of Korea; khshk18@hanyang.ac.kr (H.K.); marchenw@hanyang.ac.kr (W.C.)
* Correspondence: yyhkwon@hanyang.ac.kr

Abstract

Implementing fault-tolerant quantum computing necessitates the realization of logical non-Clifford gates, which requires the preparation of specific quantum states known as magic states. However, IBM's heavy-hexagon structure, which has limited qubit connectivity, presents challenges in adapting quantum error correction codes such as the surface code. Several methods have been proposed to address these challenges by adapting the surface code to the heavy-hexagon architecture. In this study, we implement the magic state injection process within two distinct implementations of surface codes (standard and rotated methods) suitable for the heavy-hexagon structure and compare their logical error rates. Furthermore, we propose initialization methods to enhance the performance of magic state injection in the heavy-hexagon structure, thereby efficiently achieving logical non-Clifford gates with reduced error rates.

Keywords: quantum computing; quantum error correction; magic state injection; fault-tolerant quantum computing; heavy-hexagon structure; flag qubit

MSC: 81P68

Academic Editor: João Nuno Prata

Received: 7 November 2025
Revised: 28 November 2025
Accepted: 2 December 2025
Published: 3 December 2025

Citation: Kim, H.; Choi, W.; Kwon, Y. About Implementation of Magic State Injection in Heavy-Hexagon Structure. *Mathematics* **2025**, *13*, 3874. https://doi.org/10.3390/math13233874

1. Introduction

Recent progress in the fields of Quantum Computation and Quantum Information has been remarkable [1,2]. It is based on practical development in quantum algorithms [3–8], quantum communication, and quantum hardware [9,10]

In particular, within the field of quantum computation, recent experimental demonstrations of quantum error correction have brought us closer to realizing practical fault-tolerant quantum computing [11–21]. However, a critical challenge remains the implementation of logical non-Clifford gates. While logical Clifford gates can be implemented relatively simply through transversal operations, logical non-Clifford gates cannot. Currently, the leading approach for implementing logical non-Clifford gates involves utilizing specific states known as logical magic states [22–27]. The preparation of logical magic states, referred to as magic state injection, is therefore crucial. Developing efficient and low-error methods for magic state injection is essential for achieving universal fault-tolerant quantum computing [28–36].

One of the most promising quantum error correction codes, the surface code, requires each qubit to interact with up to four neighboring qubits [37–42]. However, the heavy-hexagon architecture used by IBM has limited qubit connectivity, making direct adaptation of surface code-type quantum error correction challenging [43,44]. To address

this limitation, several approaches for adapting the surface code to the heavy-hexagon architecture have been proposed [45–54]. These various adaptations differ significantly in their capabilities to detect and correct errors, consequently affecting the performance of the magic state injection process.

In this study, we carry out the magic state injection process using two adaptations of the surface code for the heavy-hexagon structure (standard and rotated methods), comparing the resulting logical error rates [55–60]. Our results demonstrate that utilizing the rotated method initially yields a higher success probability compared to the standard method; however, we observed that the logical error rate becomes higher when the physical error rate is reduced below a certain threshold. Additionally, we identified specific error patterns responsible for this increased logical error rate and proposed methods to reduce these errors. Through these efforts, we anticipate that it will be possible to reduce logical errors during magic state injection in the heavy-hexagon structure, ultimately achieving non-Clifford gates with lower error rates.

2. Surface Code on Heavy-Hexagon Structure

The heavy-hexagon structure features limited qubit connectivity of two or three, which limits qubit interactions compared to standard lattice structures. Consequently, implementing quantum error correction codes, such as the surface code, on this architecture is significantly challenging. In this study, we investigate two specific adaptations of the surface code designed for the heavy-hexagon structure.

2.1. Surface Code on Standard Heavy-Hexagon Structure (Standard Method)

The first method directly applies the surface code to the standard heavy-hexagon structure (standard method). In this approach, one to three flag qubits are introduced to connect data qubits with syndrome qubits (Figure 1). Errors occurring on data qubits propagate through flag qubits to syndrome qubits, allowing detection of data qubit errors by measuring the syndrome qubits. The flag qubits are shared across two distinct syndrome measurement processes (X and Z), necessitating their execution in two separate subrounds.

During these measurement cycles, errors originating from flag qubits propagate differently based on their type. Z-type errors on flag qubits propagate to syndrome qubits, causing measurement errors, while X-type errors propagate to data qubits, resulting in additional errors on the data qubits depending on the stabilizer type. To prevent the propagation of these flag qubit errors into subsequent rounds, it is crucial to measure each flag qubit at the end of every subround. Furthermore, the measurement results from the flag qubits can be utilized in the decoding process, reducing the logical error rate.

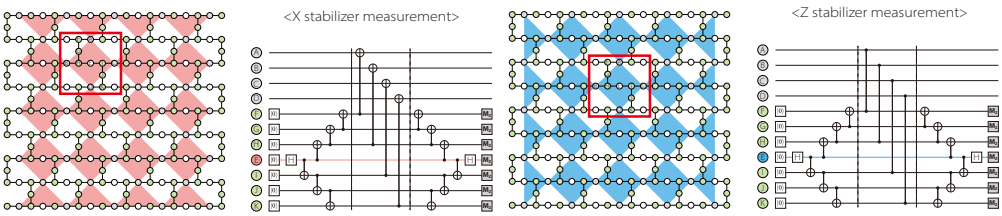

Figure 1. Adaptation of surface code to standard heavy-hexagon structure. The stabilizer measurement circuit was constructed using 1 to 3 flag qubits to connect data qubits and syndrome qubits. In the first subround, X stabilizers were measured, and in the second subround, Z stabilizers were measured. By measuring each flag qubit, errors occurring on the flag qubits could be detected. The red box indicates the unit stabilizer measurement. The gray, white, and green circles represent data qubits, syndrome qubits, and flag qubits, respectively.

2.2. Surface Code on Rotated Heavy-Hexagon Structure (Rotated Method)

The second method applies the surface code to a rotated heavy-hexagon structure (rotated method). This method addresses the limited connectivity issue through a process known as stabilizer folding. This method measures weight-4 stabilizer operators by dividing data qubits into pairs, checking parity separately for each pair, and merging this information at syndrome qubits. This approach allows simultaneous measurement of multiple weight-4 stabilizers using chains of data qubits with connectivity of two. Additionally, CNOT gates used for parity checking facilitate simultaneous folding of adjacent X and Z stabilizers, increasing the efficiency of the measurement process. However, due to connectivity constraints, stabilizers cannot all be measured simultaneously and thus are divided into two subsets, each measured in separate subrounds. Due to the rotated lattice structure, additional data qubits are required along the right and lower boundaries compared to the standard surface code. To eliminate extra degrees of freedom introduced by these additional data qubits, weight-1 stabilizer operators are employed. These boundary data qubits are not entangled with other data qubits but are measured exclusively to reduce the degrees of freedom, enabling the application of the surface code to the rotated hexagonal lattice. To implement stabilizer folding, data qubits must be connected by CNOT gates. However, applying this configuration to the heavy-hexagon architecture introduces intermediate qubits between pairs of data qubits. Consequently, indirect CNOT operations between data qubits are required, achieved through two distinct methods. The first method utilizes SWAP gates implemented via intermediate bridge qubits, transferring the state of data qubits indirectly; notably, the initial state of bridge qubits does not affect this operation. The second method implements the flag approach by utilizing measurement operations: intermediate flag qubits are prepared in specific states (0 or +) depending on the direction of the desired CNOT operation, connected via CNOT gates, and subsequently measured with proper basis to indirectly realize the required CNOT operation (Figure 2). Both SWAP and flag approaches share identical arrangements of data and syndrome qubits, differing only in their operational approach—the flag approach requires fewer CNOT gates but includes additional measurements.

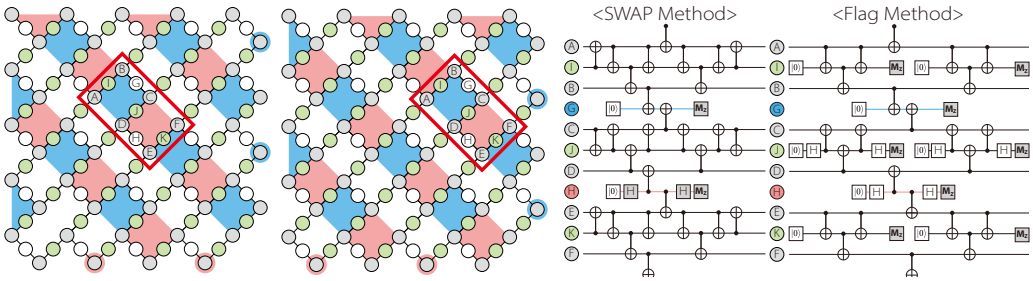

Figure 2. Adaptation of surface Code to rotated heavy-hexagon structure. The stabilizer measurement process was adapted to the rotated heavy-hexagon structure using the stabilizer folding method, enabling stabilizer measurement despite limited connectivity. Depending on how CNOT gates connecting data qubits were configured, two types of stabilizer circuits were employed: the SWAP-based and flag-based methods. The SWAP-based method utilized additional qubits known as bridge qubits, while the flag-based method used flag qubits. The full stabilizer operator was divided into two groups and measured across two subrounds. The red box indicates the unit stabilizer measurement. The gray, white, and green circles represent data qubits, syndrome qubits, and flag qubits, respectively.

2.3. Comparison of Two Methods

When applying the surface code to a rotated heavy-hexagon structure, one advantage is the reduction in the required number of physical qubits and gates (see Table 1). However, since this approach involves rotating the heavy-hexagon structure, it is impossible to utilize all qubits when directly adapting it to existing IBM hardware. Consequently, when

comparing the number of physical qubits required for error correction codes in the standard heavy-hexagon structure, the difference in required qubits becomes smaller. For a surface code with distance d, the standard method requires $10d^2 - 8d - 1$ physical qubits. In the rotated method, the actual number of physical qubits used for the same distance is $5d^2 - 2d - 2$, leading to a significant difference of approximately $\sim 5d^2$. However, when including the additional physical qubits present in the heavy-hexagon structure but not actively used, the required physical qubits increase to $10d^2 - 13d + 4$. As a result, the difference in the number of physical qubits required to implement a surface code of the same distance in standard and rotated structures reduces to approximately $\sim 5d$. Thus, we compare the two methods under the assumption that they require a similar number of physical qubits for the same distance (Figure 3).

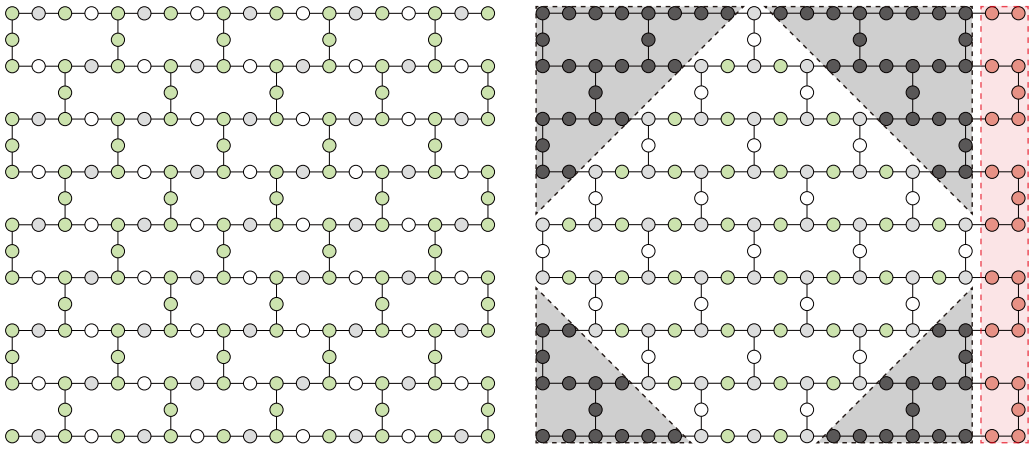

Figure 3. Comparison of the number of qubits required to implement a distance-5 surface code using the two methods. (**Left**) Standard method. (**Right**) Rotated method. The rotated method achieves the same distance surface code with fewer physical qubits, but due to the rotated structure, some physical qubits remain unused (black regions). As a result, the actual difference in the required qubit patch size is relatively small (red regions). The gray, white, and green circles represent data qubits, syndrome qubits, and flag qubits, respectively.

Table 1. Comparison of between the two surface code adaptation schemes. Since the rotated method does not perfectly fit into the heavy-hexagon lattice, some physical qubits remain unused. The total number of qubits, including these unused ones, is $10d^2 - 13d + 4$. The numbers of CNOT and readout gates are based on distance d per round.

Method	Stabilizer Measurement	Number of Physical Qubits	Number of CNOT Gates	Number of Readout Gates
Standard	Flag qubit	$10d^2 - 8d - 1$	$32d^2 - 44d + 12$	$14d^2 - 18d + 4$
Rotated (SWAP)	Stabilizer folding	$5d^2 - 2d - 2$ ($10d^2 - 13d + 4$)	$30d^2 - 72d + 42$	$2d^2 - 2d$
Rotated (flag)	Stabilizer folding	$5d^2 - 2d - 2$ ($10d^2 - 13d + 4$)	$24d^2 - 58d + 34$	$14d^2 - 34d + 20$

3. Magic State Injection

3.1. Injection Method

Magic state injection is the process of preparing a logical magic state, which is essential for implementing non-Clifford gates. These magic states cannot be constructed using only Clifford operations within the computational basis. Therefore, a method known as magic state injection has been proposed, which involves preparing a magic state at the physical qubit level and injecting it into a logical state. Magic state injection is performed by preparing the physical qubits that constitute the logical qubit, initializing each qubit into an appropriate quantum state, and then performing a projection measurement. To achieve this, a specific physical qubit must be selected and prepared in the desired magic state. The state preparation of this physical qubit and the stabilizers of the error correction

code determine how the surrounding physical qubits are initialized. The magic state injection process consists of two main stages. In the first stage, each physical qubit is initialized appropriately, followed by a projection measurement. A post-selection process then eliminates states that contain errors resulting from the projection process, allowing the extraction of a logical magic state. However, as the logical state's distance increases, the probability of obtaining an error-free state through post-selection decreases. Therefore, in the second stage, the logical state obtained in the first stage is expanded to a larger distance logical state. The detailed procedure is described in Appendix A and Figure 4.

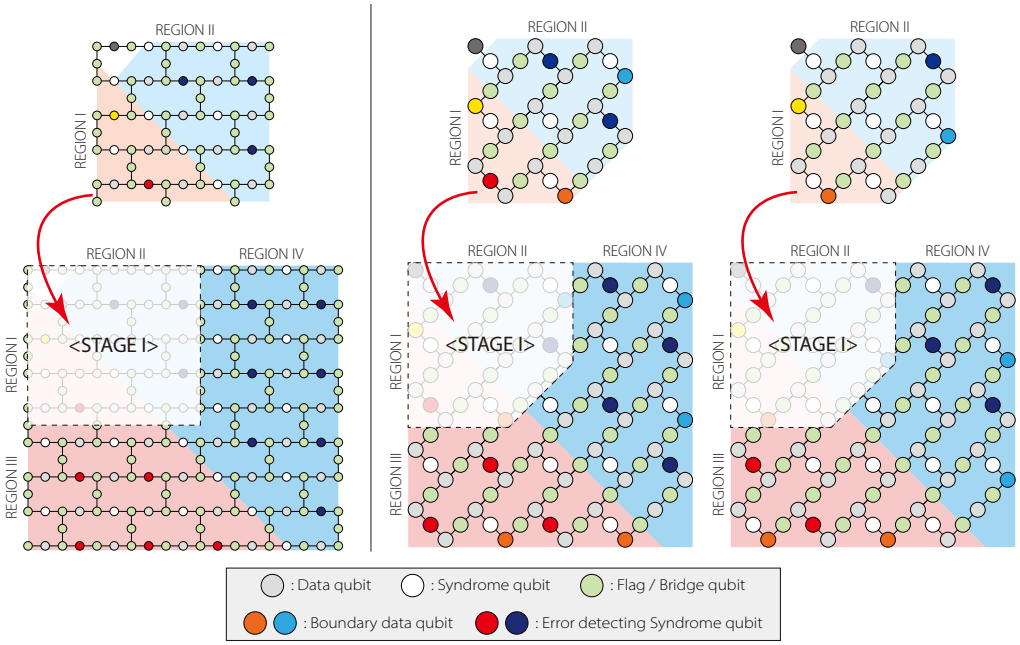

Figure 4. Magic state injection process using two adaptation methods of surface codes for heavy-hexagon structures. In both methods, the process consists of two stages. In the first stage, the state of the data qubit is projected onto the eigenspace of the stabilizer. Errors occurring during the projection process are checked, and if errors are detected, the state is discarded, and the preparation is repeated. If no errors are detected, the process proceeds to the second stage, where the logical state's distance is increased. The gray, white, and green circles represent data qubits, syndrome qubits, and flag qubits, respectively.

3.2. Injection Error

Errors in the magic state injection process that lead to logical errors can be categorized into two types. The first type occurs when errors occur during quantum state preparation or projection measurement and go undetected. The second type occurs during the process of increasing the distance of the projected quantum state. If the error correction code is sufficiently robust, the second type of error can be significantly reduced as the physical error rate decreases. However, the first type of error is inherent to the magic state injection process and cannot be completely eliminated.

The first type of error can originate from various sources. If an error occurs while preparing the magic state on the physical qubit, it cannot be corrected. Additionally, depending on the arrangement of the initial states of the data qubits, errors in certain data qubits may become undetectable. In our previous study, we examined the impact of such undetectable data qubits (blind qubits) on logical errors in the magic state injection process using the standard method. One of the key factors influencing logical errors in the magic state injection process is the initialization method. Figure 5 illustrates the initialization strategies used in previous studies to mitigate such errors. In our previous study, four different initialization methods were explored: down triangle, right triangle, down square,

and right square. However, the square methods were not consistently applicable for preparing boundary data qubits; thus, in this study, we focus only on the down triangle and right triangle methods. We previously analyzed the effect of different initialization methods on error rates in the standard heavy-hexagon structure. The results indicated that the right triangle method generally yielded lower error rates due to the positioning of blind qubits and the influence of flag qubits. However, in the rotated heavy-hexagon structure, the effects of initialization differ due to its altered configuration. In the rotated structure, certain errors introduced during stabilizer folding can simultaneously affect two data qubits. These errors commute with all stabilizers measured in that subround, making them undetectable within the same subround. Consequently, such errors must be detected in the subsequent subround through another stabilizer that shares one of the defective data qubits. In a usual error correction process, except for boundary data qubits, each data qubit is associated with multiple stabilizers, allowing for error detection and correction. However, during the magic state injection process, error detection relies on the assumption that all data qubits forming a stabilizer are prepared in its eigenstate. Thus, depending on the initialization states of other data qubits, there may be no additional stabilizer available to detect errors. As a result, when specific types of errors occur at the initialization boundaries, they may remain undetected. If such an error arises in the first subround and no stabilizer in the second subround is available to detect it, the error propagates directly, contributing to a logical error.

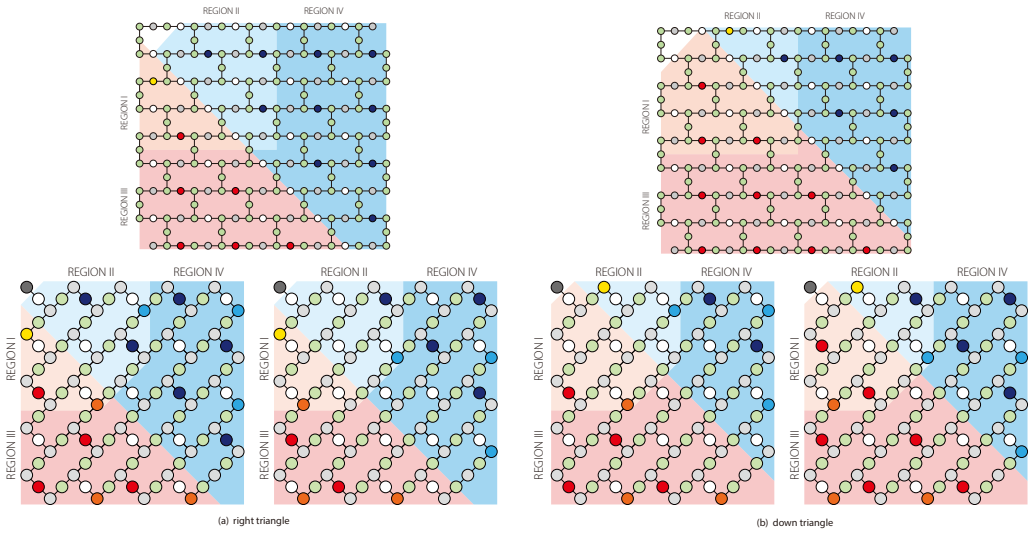

Figure 5. Initialization methods for performing magic state injection. (**a**) Right triangle method, (**b**) Down triangle method. If more qubits are located in the right region, the right triangle method is used; if more qubits are located in the lower region, the down triangle method is applied. Data qubits in the blue-shaded region are initialized to $|0\rangle$, whereas those in the red-shaded region are initialized to $|+\rangle$. Red and blue syndrome qubits represent those that can detect errors occurring in data qubits. Yellow qubits represent blind qubits whose errors cannot be detected. Orange and light blue qubits indicate weight-1 stabilizer data qubits. The gray, white, and green circles represent data qubits, syndrome qubits, and flag qubits, respectively.

Even in the standard surface code, errors occurring on specific qubits can introduce simultaneous errors in two data qubits. However, these errors can be detected through flag qubit measurements. By incorporating flag qubit measurement results into the postselection process, it is possible to identify such errors and discard erroneous states. In contrast, the SWAP-based approach does not include measurements on bridge qubits, and the flag-based method does not fully detect all two-qubit errors. As a result, it is challenging to completely eliminate errors introduced during the injection process. The occurrence of

undetected two-qubit errors depends on the initialization method. Since the initialization method determines how each data qubit is prepared, appropriately setting the initialization boundary enhances error detection. These undetected errors arise when an error in the first subround remains undetected in the second subround. Therefore, reducing such errors requires setting the initialization boundary so that fewer two-qubit errors occur in the first subround. For example, the right triangle boundary results in more undetectable two-qubit errors in the first subround. In contrast, the down triangle method shifts two-qubit errors to the second subround, leading to a lower logical error rate (See Figure 6 and Appendix B).

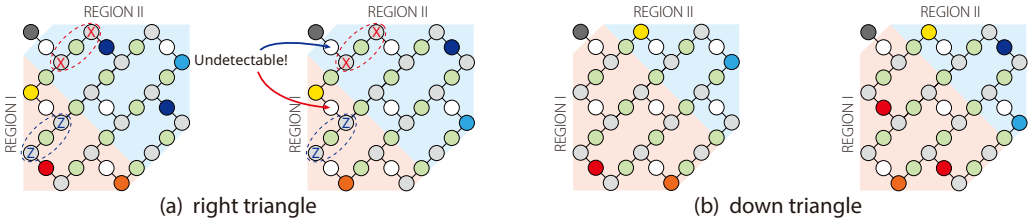

Figure 6. Some errors occurring during the injection process remain undetected in the first stage and contribute to logical errors. In the right triangle method, errors affecting two data qubits near the initialization boundary go undetected. In contrast, in the down triangle method, undetected errors are reduced because errors near the initialization boundary are not left undetected in the first subround. The dashed circles indicate simultaneous errors occurring on two data qubits. The gray, white, and green circles represent data qubits, syndrome qubits, and flag qubits, respectively.

4. Performance Comparison: Standard vs. Rotated Methods

We conducted simulations to compare the performance of magic state injection in surface codes adapted to two types of heavy-hexagon structures. The physical error rate range was set to 10^{-4}∼10^{-2} to encompass the critical threshold region of the surface code (approx. 10^{-3}∼10^{-2} [51,52]). Physical error rates exceeding 10^{-2} make effective error correction unfeasible. Conversely, the lower bound of 10^{-4} was selected to reflect the operational limits of current quantum hardware capabilities. A logarithmic scale was adopted to facilitate a detailed analysis of the logical error rate behavior in the deep error suppression regime (See Appendix C for detailed experimental setup).

First, we performed magic state injection using two different initialization methods: down triangle and right triangle. For each initialization method, we applied magic state injection to a surface code with distance 3 and then gradually increased the distance in steps of 2 up to distance 9, observing the changes in logical error rates in Figure 7. When comparing the SWAP-based and flag-based approaches for implementing the rotated method, we found that both methods exhibited nearly identical logical error rates. This is likely because our error model assumes equal error rates for CNOT gates and measurement operations. However, in terms of success probability, the flag-based approach demonstrated a slightly higher success probability than the SWAP-based approach. Since the error correction performance of these two methods depends on the specific gate error rates, their performance may differ under different error models.

Figure 8 presents a performance comparison between the standard and rotated methods utilizing right triangle and down triangle initialization. Consistent with our previous findings, the right triangle initialization yields a lower logical error rate for the standard method. In contrast, our current results demonstrate that for the rotated method, the down triangle initialization results in superior performance. This inversion is attributed to the down triangle method's ability to mitigate undetectable errors affecting two data qubits, particularly near the boundary where the first subround lacks sufficient error-detecting stabilizers (see Section 3.2).

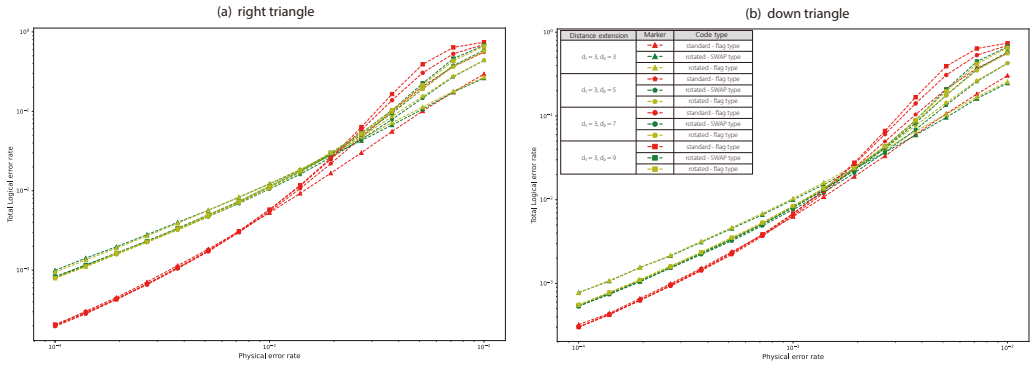

Figure 7. Comparison of performance between the standard method and rotated method using (**a**) right triangle and (**b**) down triangle initialization. The standard method exhibits a lower error rate in the low physical error rate regime.

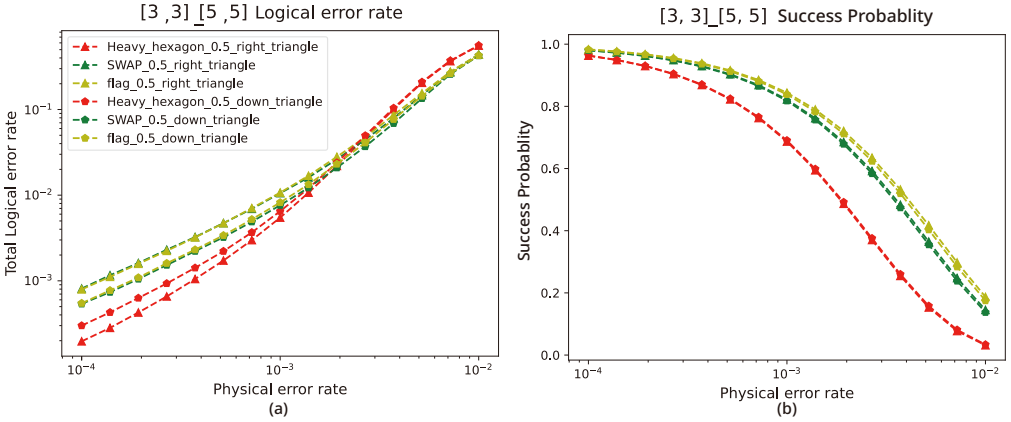

Figure 8. Comparison of (**a**) logical error rate and (**b**) success probability for the initialization methods using down triangle and right triangle. The rotated method showed a lower error rate with down triangle initialization, whereas the standard method had a lower error rate with right triangle initialization. Additionally, the success probability of the standard method was observed to be lower than that of the rotated method.

Furthermore, we observe that in the low physical error rate regime ($\leq 10^{-3}$), the standard method exhibits a lower logical error rate than the rotated method. This difference arises because the stabilizer folding technique in the rotated method increases the incidence of simultaneous two-qubit errors, which are challenging to detect during the first injection stage. These results demonstrate that in the physical error rate regime sufficiently below the threshold, the standard method outperforms the rotated method despite its higher overhead in physical qubits and gates. However, this advantage comes with a trade-off: the success probability of the standard method is lower than that of the rotated method. Consequently, selecting the optimal method necessitates a comprehensive evaluation of physical qubit requirements, achievable logical error rates, and success probabilities.

5. Conclusions and Discussion

We conducted simulations to evaluate magic state injection in two adapted surface code methods for the heavy-hexagon structure. The results indicate that the performance depends on the physical error rate. Specifically, in the low physical error rate regime, the standard method exhibits a lower logical error rate. However, in the high physical error rate regime, the rotated method exhibits a lower logical error rate compared to the standard method. Regarding the initialization method, the standard method achieves a lower error rate with the right triangle initialization, whereas the rotated method performs better with the down triangle initialization. Furthermore, the rotated method demonstrates a higher

success probability compared to the standard method.

In the magic state injection process, not all stabilizers can detect errors, meaning that errors can only be identified through a subset of stabilizers. However, because the rotated method utilizes fewer physical qubits, there are cases where errors remain undetected by the stabilizers used in magic state injection. This issue may be mitigated by adjusting the initialization of data qubits. Additionally, further modifications to the initialization process—such as optimizing the placement of the injection qubit or refining methods to increase the distance—could enable magic state injection with an even lower logical error rate in the rotated method.

Author Contributions: Conceptualization, H.K., W.C. and Y.K.; methodology, H.K.; software, H.K.; validation, H.K., W.C. and Y.K.; formal analysis, H.K.; investigation, H.K.; resources, Y.K.; data curation, H.K.; writing—original draft preparation, H.K.; writing—review and editing, H.K., W.C. and Y.K.; visualization, H.K.; supervision, Y.K.; project administration, Y.K.; funding acquisition, Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Creation of the Quantum Information Science R&D Ecosystem (Grant No. 2022M3H3A106307411) through the National Research Foundation of Korea (NRF) funded by the Korean government (Ministry of Science and ICT).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Magic State Injection Process

The magic state injection process was carried out in two stages. In the first stage, a magic state was prepared on the top-left physical qubit, while all other physical qubits were initialized in either the $|0\rangle$ or $|+\rangle$ state. The initialization method in our approach determined how the data qubits were prepared in specific quantum states. In this study, data qubits in the right region were initialized to $|0\rangle$, while those in the lower region were initialized to $|+\rangle$. Previously, we investigated four initialization methods: down triangle, right triangle, down square, and right square. However, for the rotated structure, applying the square methods led to issues in preparing the initial states of boundary data qubits. Therefore, in this study, we considered only the down triangle and right triangle methods. In the first stage, data qubit initialization was followed by projection measurement. For a distance-3 qubit patch, we divided the system into two regions (Region I and Region II), where data qubits in Region I were initialized to $|0\rangle$, and those in Region II were initialized to $|+\rangle$. The top-left data qubit, assigned for magic state preparation, did not belong to either region. After state initialization, two rounds of stabilizer measurement were performed. The first stabilizer measurement collapses the data qubit states into the eigenspace of the stabilizers. Errors occurring during this process can be detected by surrounding data qubits, which are already eigenstates of the relevant stabilizers. However, this alone is insufficient for detecting all errors arising during injection. Thus, an additional stabilizer measurement and parity comparison were performed to detect errors. Through this injection process, we could detect errors. However, since the initial state was not a logical state, error correction was not possible. Instead, we employed a post-selection process, discarding erroneous states and proceeding only with states where no errors were detected. The second stage involved expanding the logical magic state obtained in the first stage to a larger-distance magic state. This was accomplished by defining Region

III and Region IV using the same criteria as in the first stage. Data qubits in Region III were initialized to $|0\rangle$, and those in Region IV were initialized to $|+\rangle$. After initialization, stabilizer measurements were performed. Since we obtained a logical state in the first stage, error correction could now be applied. After expansion, stabilizer measurements were repeated for as many rounds as the extended distance. For comparison, we also performed a number of rounds equal to the initial distance before expansion. To increase the distance of the logical state, a larger stabilizer group that commutes with the stabilizers of the logical state obtained in the first stage had to be constructed. In the standard heavy-hexagon structure, as the distance increases, the expanded stabilizer group naturally includes the stabilizer group of the smaller distance. However, in the rotated heavy-hexagon structure, the weight-1 stabilizers located at the right and lower boundaries anti-commute with the stabilizer group of the expanded code. These boundary data qubits remain in their initially prepared states or states affected by Pauli errors without forming entanglement with other data qubits. As a result, when increasing the distance of the magic state in the rotated heavy-hexagon structure, the boundary data qubits from the previous stage undergo projection measurement and collapse. This collapse process prevents the use of some stabilizer measurement results from the previous stage. Nevertheless, by excluding these weight-1 stabilizers, and considering only the remaining data qubits, it is still possible to represent a specific logical state, allowing for distance expansion. Therefore, in our method, we performed projection measurement in the second stage without including the stabilizer measurement results of the boundary data qubits from the first stage into the decoding process.

Appendix B. Error Detecting on Magic State Injection

In the first stage, stabilizer measurements project the prepared state onto the eigenspace of a low-distance surface code. Various types of errors can occur in this process, but not all of them can be detected and corrected. Since the magic state is prepared on a physical qubit, errors in that qubit cannot be detected. Additionally, depending on the initialization method, certain data qubits become blind qubits, making error detection impossible for those qubits. When applying the surface code to low-connectivity architectures such as the heavy-hexagon structure, stabilizer measurements cannot be performed simultaneously for all stabilizers and must be divided into multiple subrounds. Consequently, some errors occurring in the first subround can only be detected in the second subround. While such error propagation does not impact error rates in conventional error correction for logical states, it becomes problematic in magic state injection, where errors must be identified within the first two measurement rounds. Certain errors occurring in the first subround may only be detectable in the second subround, and if those stabilizers fail to detect errors, they contribute to logical errors (Figure A1). Thus, we must analyze cases where errors in the first subround remain undetected in the second subround. A representative case occurs when a stabilizer meant to detect an error, specifically one occurring in the first subround, has data qubits in non-eigenstates, leading to random syndrome measurement outcomes. Fortunately, most data qubits are linked to two stabilizers, making it possible to prevent undetectable errors through appropriate initialization. However, circuit-level errors affecting two data qubits simultaneously may go undetected due to the greater separation between their detecting stabilizers. In the standard method, such two-qubit errors arise when X errors occur on flag qubits, which can be mitigated by including flag qubit measurement results in the post-selection process. However, in the rotated method, this additional detection is not always possible, leading to an increase in undetectable errors during magic state injection.

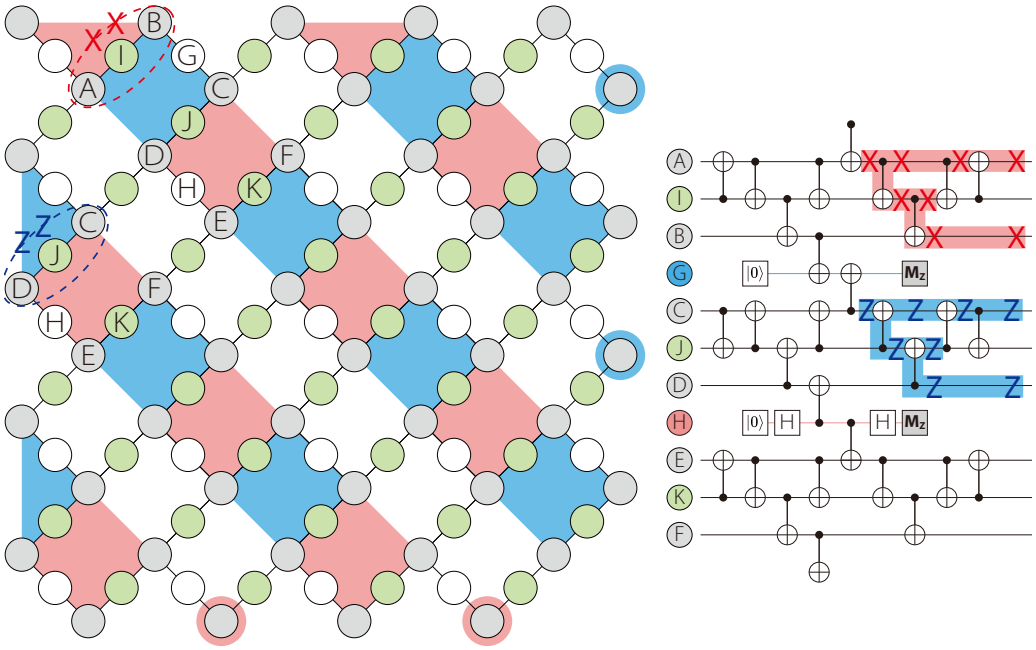

Figure A1. An example of undetectable two-qubit errors in the rotated method. In the rotated method, specific types of errors can propagate simultaneously to two data qubits, as shown in the circuit on the right. Since these errors are undetectable within the current subround, they must be identified in subsequent subrounds. However, during magic state injection, certain stabilizers are utilized for state projection; consequently, if such errors occur at the initialization boundary, they may remain undetected and directly contribute to a logical error. The dashed circles indicate simultaneous errors occurring on two data qubits. The gray, white, and green circles represent data qubits, syndrome qubits, and flag qubits, respectively.

Appendix C. Experimental Setup

We performed circuit-level simulations to implement the magic state injection process. For the simulations, we used the Stim code and employed the minimum weight perfect matching algorithm for decoding. Each stabilizer measurement circuit was implemented using CNOT, Hadamard, measurement, and reset gates. In the standard method, an additional CZ gate was used for Z stabilizer construction. Using the Stim simulator, we conducted 10^7 trials for logical X and Z errors in each experiment and calculated the logical error rate using the following formula:

$$E_{total} = 1 - (1 - E_X)(1 - E_Z) \tag{A1}$$

In these simulations, we assumed a depolarizing error channel. The error sources included single-qubit errors, two-qubit errors, and readout errors, with reset errors treated as single-qubit errors. The two-qubit error rate and readout error rate were assumed to be equal, and this value was defined as the physical error rate parameter. Experiments were performed by varying this parameter. The single-qubit error rate was assumed to be 1/20 of the physical error rate.

References

1. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2010.
2. Bennett, C.H.; Brassard, G. Quantum cryptography: Public key distribution and coin tossing. *Theor. Comput. Sci.* **2014**, *560*, 7. [CrossRef]
3. Shor, P.W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Rev.* **1999**, *41*, 303–332. [CrossRef]

4. Grover, L.K. Quantum Computers Can Search Rapidly by Using Almost Any Transformation. *Phys. Rev. Lett.* **1998**, *80*, 4329–4332. [CrossRef]
5. Bae, J.; Kwon, Y. Generalized quantum search Hamiltonian *Phys. Rev. A* **2002**, *66*, 012314. [CrossRef]
6. Abrams, D.S.; Lloyd, S. Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors. *Phys. Rev. Lett.* **1999**, *83*, 5162–5165. [CrossRef]
7. Park, S.; Bae, J.; Kwon, Y. Wavelet quantum search algorithm with partial information *Chaos Solitons Fractals* **2007**, *32*, 1371–1374. [CrossRef]
8. Namkung, M.; Kwon, Y. Coherence and Entanglement Dynamics in Training Variational Quantum Perceptron. *Entropy* **2020**, *22*, 1277. [CrossRef]
9. Barends, R.; Kelly, J.; Megrant, A.; Veitia, A.; Sank, D.; Jeffrey, E.; White, T.C.; Mutus, J.; Fowler, A.G.; Campbell, B.; et al. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature* **2014**, *508*, 500–503. [CrossRef]
10. Kang, J.; Kim, C.; Kim, Y.; Kwon, Y. New design of three-qubit system with three transmons and a single fixed-frequency resonator coupler. *Sci. Rep.* **2025**, *15*, 12134. [CrossRef]
11. Krinner, S.; Lacroix, N.; Remm, A.; Paolo, A.D.; Genois, E.; Leroux, C.; Hellings, C.; Lazar, S.; Swiadek, F.; Herrmann, J.; et al. Realizing repeated quantum error correction in a distance-three surface code. *Nature* **2022**, *605*, 669. [CrossRef]
12. Zhao, Y.; Ye, Y.; Huang, H.-L.; Zhang, Y.; Wu, D.; Guan, H.; Zhu, Q.; Wei, Z.; He, T.; Cao, S.; et al. Realization of an error-correcting surface code with superconducting qubits. *Phys. Rev. Lett.* **2022**, *129*, 030501. [CrossRef]
13. Postler, L.; Heußen, S.; Pogorelov, I.; Rispler, M.; Feldker, T.; Meth, M.; Marciniak, C.D.; Stricker, R.; Ringbauer, M.; Blatt, R.; et al. Demonstration of fault-tolerant universal quantum gate operations. *Nature* **2022**, *605*, 675. [CrossRef]
14. Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. *Nature* **2023**, *614*, 676. [CrossRef] [PubMed]
15. Gupta, R.S.; Sundaresan, N.; Alexander, T.; Wood, C.J.; Merkel, S.T.; Healy, M.B.; Hillenbrand, M.; Jochym-O'Connor, T.; Wootton, J.R.; Yoder, T.J.; et al. Encoding a magic state with beyond break-even fidelity. *Nature* **2024**, *625*, 259–263. [CrossRef] [PubMed]
16. Bluvstein, D.; Evered, S.J.; Geim, A.A.; Li, S.H.; Zhou, H.; Manovitz, T.; Ebadi, S.; Cain, M.; Kalinowski, M.; Hangleiter, D.; et al. Logical quantum processor based on reconfigurable atom arrays. *Nature* **2024**, *626*, 58. [CrossRef]
17. Hetenyi, B.; Wootton, J.R. Creating entangled logical qubits in the heavy-hex lattice with topological codes. *Quantum* **2024**, *5*, 040334. [CrossRef]
18. Kim, Y.; Kim, H.; Kang, J.; Choi, W.; Kwon, Y. Effectiveness of the syndrome extraction circuit with flag qubits on IBM quantum hardware. *arXiv* **2024**, arXiv:2403.10217. [CrossRef]
19. Kim, Y.; Sevior, M.; Usman, M. Magic State Injection on IBM Quantum Processors Above the Distillation Threshold. *arXiv* **2024**, arXiv:2412.01446. [CrossRef]
20. Kim, Y.; Sevior, M.; Usman, M. Transversal cnot gate with multicycle error correction. *Phys. Rev. Appl.* **2025**, *23*, 024074. [CrossRef]
21. Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. *Nature* **2025**, *638*, 920–926. [CrossRef]
22. Eastin, B.; Knill, E. Restrictions on transversal encoded quantum gate sets. *Phys. Rev. Lett.* **2009**, *102*, 110502. [CrossRef] [PubMed]
23. Zeng, B.; Cross, A.; Chuang, I.L. Transversality versus universality for additive quantum codes. *IEEE Trans. Inf. Theory* **2011**, *57*, 62726284. [CrossRef]
24. Bravyi, S.; Kitaev, A. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Phys. Rev. A* **2005**, *71*, 022316. [CrossRef]
25. Horsman, D.; Fowler, A.G.; Devitt, S.; Van Meter, R. Surface code quantum computing by lattice surgery. *New J. Phys.* **2012**, *14*, 123011. [CrossRef]
26. Landahl, A.J.; Ryan-Anderson, C. Quantum computing by colorcode lattice surgery. *arXiv* **2014**, arXiv:1407.5103. [CrossRef]
27. Vuillot, C.; Lao, L.; Criger, B.; Almudéver, C.G.; Bertels, K.; Terhal, B.M. Code deformation and lattice surgery are gauge fixing. *New J. Phys.* **2019**, *21*, 033028. [CrossRef]
28. Bravyi, S.; Haah, J. magic state distillation with low overhead. *Phys. Rev. A* **2012**, *86*, 052329. [CrossRef]
29. O'Gorman, J.; Campbell, E.T. Quantum computation with realistic magic-state factories. *Phys. Rev. A* **2017**, *95*, 032338. [CrossRef]
30. Gidney, C.; Shutty, N.; Jones, C. Magic state cultivation: Growing T states as cheap as CNOT gates. *arXiv* **2024**, arXiv:2409.17595. [CrossRef]
31. Kim, H.; Choi, W.; Kwon, Y. Implementation of Magic State Injection within Heavy-Hexagon Architecture. *arXiv* **2024**, arXiv:2412.15751. [CrossRef]
32. Gidney, C. Cleaner magic states with hook injection. *arXiv* **2023**, arXiv:2302.12292. [CrossRef]
33. Chamberland, C.; Noh, K. Very low overhead fault-tolerant magic state preparation using redundant ancilla encoding and flag qubits. *npj Quantum Inf.* **2020**, *6*, 91. [CrossRef]
34. Li, Y. A magic state's fidelity can be superior to the operations that created it. *New J. Phys.* **2015**, *17*, 023037. [CrossRef]

35. Singh, S.; Darmawan, A.S.; Brown, B.J.; Puri, S. High-fidelity magic state preparation with a biased-noise architecture. *Phys. Rev. A* **2022**, *105*, 052410. [CrossRef]
36. Lao, L.; Criger, B. Magic state injection on the rotated surface code. In Proceedings of the 19th ACM International Conference on Computing Frontiers, Turin, Italy, 17–22 May 2022.
37. Gottesman, D. *Stabilizer Codes and Quantum Error Correction*; California Institute of Technology: Pasadena, CA, USA, 1997.
38. Bravyi, S.B.; Kitaev, A.Y. Quantum codes on a lattice with boundary. *arXiv* **1998**, arXiv:quant-ph/9811052. [CrossRef]
39. Roffe, J. Quantum error correction: An introductory guide. *Contemp. Phys.* **2019**, *60*, 226. [CrossRef]
40. Kitaev, A.Y. Fault-tolerant quantum computation by Anyons. *Ann. Phys.* **2003**, *303*, 2–30. [CrossRef]
41. Fowler, A.G.; Mariantoni, M.; Martinis, J.M.; Cleland, A.N. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A* **2012**, *86*, 032324. [CrossRef]
42. Bravyi, S.; Englbrecht, M.; König, R.; Peard, N. Correcting coherent errors with surface codes. *npj Quantum Inf.* **2018**, *4*, 55. [CrossRef]
43. IBM. IBM Quantum Experience Devices. 2025. Available online: https://quantum-computing.ibm.com/ (accessed on 7 November 2025).
44. Zhang, E.J.; Srinivasan, S.; Sundaresan, N.; Bogorin, D.F.; Martin, Y.; Hertzberg, J.B.; Chow, J.M. High-performance superconducting quantum processors via laser annealing of transmon qubits. *Sci. Adv.* **2022**, *8*, eabi6690. [CrossRef]
45. Chao, R.; Reichardt, B.W. Fault-tolerant quantum computation with few qubits. *npj Quantum Inf.* **2018**, *4*, 42. [CrossRef]
46. Chao, R.; Reichardt, B.W. Flag fault-tolerant error correction for any stabilizer code. *PRX Quantum* **2020**, *1*, 010302. [CrossRef]
47. Lao, L.; Almudever, C.G. Fault-tolerant quantum error correction on near-term quantum processors using flag and bridge qubits. *Phys. Rev. A* **2020**, *101*, 032333. [CrossRef]
48. Chamberl, ; C.; Zhu, G.; Yoder, T.J.; Hertzberg, J.B.; Cross, A.W. Topological and subsystem codes on low-degree graphs with flag qubits. *Phys. Rev. X* **2020**, *10*, 011022.
49. Wu, A.; Li, G.; Zhang, H.; Guerreschi, G.G.; Ding, Y.; Xie, Y. Mapping Surface Code to Superconducting Quantum Processors. *arXiv* **2021**, arXiv:2111.13729. [CrossRef]
50. Gidney, C.; Newman, M.; Fowler, A.; Broughton, M. A fault tolerant honeycomb memory. *Quantum* **2021**, *5*, 605. [CrossRef]
51. Kim, Y.; Kang, J.; Kwon, Y. Design of Quantum error correcting code for biased error on heavy-hexagon structure. *arXiv* **2022**, arXiv:2211.14038. [CrossRef]
52. Benito, C.; López, E.; Peropadre, B.; Bermudez, A. Comparative study of quantum error correction strategies for the heavy-hexagonal lattice. *arXiv* **2024**, arXiv:2402.02185. [CrossRef]
53. Wootton, J.R. Measurements of Floquet code plaquette stabilizers. *arXiv* **2022**, arXiv:2210.13154. [CrossRef]
54. Carroll, M.S.; Wootton, J.R.; Cross, A.W. Subsystem surface and compass code sensitivities to non-identical infidelity distributions on heavy-hex lattice. *arXiv* **2024**, arXiv:2402.08203.
55. Gidney, C. Stim: A fast stabilizer circuit simulator. *Quantum* **2021**, *5*, 497. [CrossRef]
56. Higgott, O.; Gidney, C. Sparse Blossom: Correcting a million errors per core second with minimum-weight matching. *arXiv* **2023**, arXiv:2303.15933. [CrossRef]
57. Edmonds, J. Paths, trees, and Flowers. *Can. J. Math.* **1965**, *17*, 449–467. [CrossRef]
58. Kolmogorov, V. Blossom V: A new implementation of a minimum cost perfect matching algorithm. *Math. Program. Comput.* **2009**, *1*, 43–67. [CrossRef]
59. Fowler, A.G.; Whiteside, A.C.; Hollenberg, L.C. Towards practical classical processing for the surface code. *Phys. Rev. Lett.* **2012**, *108*, 180501. [CrossRef]
60. Fowler, A.G. Minimum weight perfect matching of fault-tolerant topological quantum error correction in average O(1) parallel time. *Quantum Inf. Comput.* **2015**, *15*, 145–158. [CrossRef]