



NORTHWESTERN UNIVERSITY

Computer Science Department

Technical Report
Number: NU-CS-2026-07

March, 2026

Defining and Evaluating the Boundaries of Acceptable Defectiveness: Case Studies of Uniform and Heterogeneous Distributions of Noise on the Surface Code

Jacob Palmer

Abstract

Past research on superconducting qubits shows that these devices exhibit considerable variation and thus cannot be realistically captured under a uniform, homogeneous noise model. Our work aims to define the boundaries of acceptable defectiveness (BADs), the upper boundary of defectiveness with respect to lattice-wide variation or outlier defects, beyond which the logical computation becomes effectively unusable for a targeted application or process. We present a simulation framework that enables rapid experimentation with the performance of rotated surface codes under a parametrizable noise model. Using this tool, memory experiment circuits were generated and sampled for distances 3 to 17, with various individual defects and deviated Gaussian noise distributions. The results suggest that there are boundaries of acceptable defectiveness in which a defective qubit, with a physical error rate less than or equal to 0.75, can be left in the lattice with negligible impact on logical error rate given sufficient code distances and proper placement in the lattice. When modeling noise as a Gaussian distribution, the logical error rate shows minimal impact as deviation increases. Our research shows how heterogeneity affects logical error rates and, using our framework, supports defining quantitative design targets for hardware with imperfect or non-uniform qubit qualities.

Keywords

Quantum Computing, Quantum Error Correction, Surface Codes

NORTHWESTERN UNIVERSITY

Defining and Evaluating the Boundaries of Acceptable Defectiveness:
Case Studies of Uniform and Heterogeneous Distributions of Noise on the Surface Code

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

MASTER OF SCIENCE

Computer Science

By

Jacob Palmer

EVANSTON, ILLINOIS

June 2026

© Copyright by Jacob Palmer 2026

All Rights Reserved

Abstract

Past research on superconducting qubits shows that these devices exhibit considerable variation and thus cannot be realistically captured under a uniform, homogeneous noise model. Our work aims to define the boundaries of acceptable defectiveness (BADs), the upper boundary of defectiveness with respect to lattice-wide variation or outlier defects, beyond which the logical computation becomes effectively unusable for a targeted application or process. We present a simulation framework that enables rapid experimentation with the performance of rotated surface codes under a parametrizable noise model. Using this tool, memory experiment circuits were generated and sampled for distances 3 to 17, with various individual defects and deviated Gaussian noise distributions. The results suggest that there are boundaries of acceptable defectiveness in which a defective qubit, with a physical error rate less than or equal to 0.75, can be left in the lattice with negligible impact on logical error rate given sufficient code distances and proper placement in the lattice. When modeling noise as a Gaussian distribution, the logical error rate shows minimal impact as deviation increases. Our research shows how heterogeneity affects logical error rates and, using our framework, supports defining quantitative design targets for hardware with imperfect or non-uniform qubit qualities.

Acknowledgment

First, I would like to thank my advisor, Professor Kate Smith, whose passion for the subject and endless support on this project sparked my desire to continue my studies in this field. I'd also like to thank Professor Nikos Hardavellas for serving on my thesis committee and for the incredibly helpful feedback regarding this thesis and future directions on this work.

Additionally, I'd like to thank both Malcolm Carroll and Andrew Cross from IBM for providing invaluable feedback on earlier drafts of this research.

I'd like to extend my gratitude to my friend, Elizabeth Kuntz, whose feedback and patience in reading and listening to countless drafts of this work have kept me sane. Finally, I'd like to thank my family. Without the support of both my parents and my sister, I would not have had the ability or dedication to be where I am today.

List of Abbreviations

- **BAD** → Boundaries of Acceptable Defectiveness
- **LER** → Logical Error Rate
- **MWPM** → Minimum Weight Perfect Matching
- **QEC** → Quantum Error Correction
- **SC** → (Rotated) Surface Code

Dedication

To Tracey Hayden, whose teachings and encouragement have instilled a lifelong wonder and curiosity for the sciences.

Table of Contents

Abstract	3
Acknowledgment	4
List of Abbreviations	5
Dedication	6
Table of Contents	7
List of Figures	9
1 Introduction	10
1.1 Contributions	10
1.2 Thesis Organization	12
2 Background	13
2.1 Classical Error Correction	13
2.2 Classical Bits to Quantum Qubits	15
2.3 Quantum Error Correction	18
2.4 The Surface Code	19
2.4.1 Homogeneous vs. Heterogeneous Assumptions of Noise on Surface Code Lattices	20
2.5 Previous Work	21
3 Simulation Framework	23
3.1 Building Circuits as Stim Strings	24
3.2 Types of Distributions	25

- 3.2.1 Homogeneous Distributions 25
 - 3.2.2 Heterogeneous Distributions 26
 - 3.2.3 Defective Qubits 27
- 3.3 Operational Noise 27
- 4 Boundaries of Acceptable Defectiveness 29**
- 5 Case Studies and Analysis 30**
 - 5.1 Case Study 1: Homogeneous Noise 32
 - 5.2 Case Study 2: Homogeneous Noise with Outlier Defect 35
 - 5.3 Case Study 3: Heterogeneous (Gaussian) Noise 38
 - 5.4 Case Study 4: Heterogeneous (Gaussian) Noise with Outlier Defect 42
- 6 Discussion 45**
- 7 Extensions and Future Directions 48**
- 8 Conclusion 50**
 - References 51
 - Vita 56

List of Figures

2.1	Bloch sphere visualization	17
2.2	$d=5$ rotated surface code lattice	20
3.1	Defect locations on a $d=5$ SC lattice	25
3.2	Example heatmap of Gaussian noise distribution across a $d=5$ lattice	26
5.1	Case Study 1 - Noise distribution heatmap	32
5.2	Case Study 1 - Logical performance for $d=(3, 5, 7, 9, 11, 13, 15, 17)$	33
5.3	Case Study 2 - Noise distribution heatmap	35
5.4	Case Study 2 - Logical performance for $d=(3, 5, 7, 9)$	36
5.5	Case Study 2 - Logical performance for $d=(11, 13, 15, 17)$	37
5.6	Case Study 2 - Logical error rates vs. defective qubit error rate	38
5.7	Case Study 3 - Heterogeneous noise distribution heatmap	39
5.8	Case Study 3 - Logical performance for $d=(3, 5, 7, 9)$	40
5.9	Case Study 3 - Logical performance for $d=(11, 13, 15, 17)$	41
5.10	Case Study 4 - Noise distribution heatmap	42
5.11	Case Study 4 - Logical performance for $d=(3, 5, 7, 9)$	43
5.12	Case Study 4 - Logical performance for $d=(11, 13, 15, 17)$	44

CHAPTER 1

Introduction

Quantum error correction (QEC) enables fault-tolerant quantum computing by utilizing multiple physical qubits to redundantly encode a single abstract logical qubit, thereby making the information more resilient to errors such as decoherence or gate noise [8]. A leading method for QEC is the surface code (SC), a topological QEC code in which physical qubits are arranged in a 2D lattice and operations are performed between neighboring qubits [8]. The SC is favorable for near-term QEC demonstrations because of its high noise tolerance, and defective qubits in the 2D lattice can be managed by deforming the lattice around the defect at the cost of additional physical qubit overhead [2, 28, 16]. Existing SC research explores (1) how logical error rate (LER) is affected by physical error rate and (2) how physical overheads are impacted by defects. However, analysis often considers physical qubit performance, specifically in regard to superconducting qubit architectures, in a rigid, homogeneous manner wherein physical qubits are characterized as either uniform in their error rates or completely unusable. As a result, critical insights into the impact and upper bounds of the usability of qubits that range in their defectiveness are currently sparse. A better understanding of how SC tolerates heterogeneous error landscapes is necessary to improve the mapping of QEC to realistic quantum systems.

1.1 Contributions

The primary goal of this work is to better understand how the SC responds when the physical error rate is not held constant across a lattice, such as with normally distributed noise or individual

defects, and to provide a framework for analyzing performance in the presence of heterogeneity. Using a QEC simulation framework that supports distributed error models and is based on Stim, we investigate the impacts of defective outlier qubits and normally distributed noise on overall LER across code distances ranging from 3 to 17. Our contributions are as follows:

1. For the first time, we define the concept of *boundaries of acceptable defectiveness* (BADs) as a way to examine and quantify the performance and thresholds for imperfect noise models. The community lacks quantitative guidance with respect to the question of “how defective is too defective” for qubit error distance from the mean. Our research shows the utility of treating the physical error rate as a distribution when selecting the best SC solution to achieve an algorithm’s target logical error. Further, our simulation results show (1) the existence of BADs in both homogeneous and heterogeneous noise models and (2) these BADs are not static and differ greatly depending on the assumed noise model (i.e., the quality of qubits within a lattice) and targets for LER. Real quantum hardware exhibits vast variation in noise, and thus, in the pursuit of grounding our simulated models in reality, identifying the BADs allows a more accurate understanding of how performance scales given any arbitrary noise model.
2. We developed a novel circuit generator framework using Stim for rapid experimentation of SC performance under any arbitrary and unique noise model. Our code, which will be open source, creates quantum circuits with granular control over each qubit’s fidelity, enabling comparisons across any configuration of noise models. It also provides a novel generation of heterogeneous noise models based on deviations from a mean physical error rate. This tool has a high potential to find utility in QEC resource estimation and design space exploration based on superconducting qubits. This code and demo notebooks used to generate the figures within this paper can be found here on GitHub.
3. We model the performance of the SC under heterogeneous noise models, providing additional insights into the relationship between logical performance on imperfect quantum de-

vices and increasingly deviant physical noise distributions.

1.2 Thesis Organization

This thesis is an extended version of a pre-print co-authored with Kaitlin Smith [23]. The remainder of the thesis is structured as follows: Chapter 2 provides a brief background on classical and quantum error correction as well as the surface code and a discussion of previous work. Chapter 3 delves into the circuit generation tool developed as part of this work. Chapter 4 formally introduces the boundaries of acceptable defectiveness framework. Next, Chapter 5 contains the results simulating four different types chosen noise models with further discussion on these results found in Chapter 6. Finally, Chapters 7 and 8 conclude this thesis with propositions for future work and conclusion to the paper.

CHAPTER 2

Background

2.1 Classical Error Correction

To understand the necessity and unique complexity introduced in quantum error correction, it is first helpful to review some basic principles of classical error correction. At its simplest, error correction codes involve the utilization of multiple bits to encode single units of information [25].

All error correction codes share the same structure:

1. **Source** → The logical information that we want to encode
2. **Encode** → Encode the information over multiple bits redundantly using the instructions determined by the specific error correction code, called “codewords”
3. **Noise** → Noise can be introduced during computation, causing erroneous, unintended changes to the encoded information
4. **Decode** → Detect if any errors occurred and decode the information accordingly

A simple example of such a code is the three-bit repetition code, in which we encode 0 and 1 a series of three bits, like such:

$0 \rightarrow 000$

$1 \rightarrow 111$

These three-bit strings are called *codewords*, which are valid sequences of bits that encode our original data with additional redundancy that can be then be used in error correction and detection during data processing and storage. For instance, to encode 0, we copy the logical information into three individual bits, creating $\{000\}$. Here, $\{000\}$ and $\{111\}$ are our codewords. For decoding, we can use majority voting on the output to determine if an error occurred in our logical information. On classical devices, a bit-flip error occurs when a bit erroneously changes from a 0 to 1 (or vice versa). In this thesis, we will later refer to errors that occur to the individual bits that comprise our encoded logical information as *physical errors*. In the three-bit code example, a bit-flip can occur on any of the three bits that comprise our encoded data. If our data is initially encoded as $\{000\}$, then upon decoding we could determine the following about possible errors that occurred:

- $\{000\} \rightarrow 0$: (*No Physical Error or Logical Error*) No errors occurred since our source matches the logical source
- $\{100\}, \{010\}, \{001\} \rightarrow 0$: (*1 Physical Error; No Logical Error*) A singular bit-flip error ($1 \rightarrow 0$; $0 \rightarrow 1$) occurs, but we correctly decode this as 0 by majority vote.
- $\{011\}, \{110\}, \{101\} \rightarrow 1$: (*2 Physical Errors; Logical Error*) Two bit-flip errors occur, causing the logical information to be decoded incorrectly as 1.
- $\{111\} \rightarrow 1$: (*3 Physical Errors; Logical Error*) All three bits are subject to a bit-flip error, which results in the valid codeword for the opposite state from our logical source

This three-bit repetition code can handle at most 1 error, since any more than one bit-flip causes the information to be decoded to the opposite state. When three bit-flips occur, the result is $\{111\}$, which is itself a valid codeword for 1. This is what is defined as a *logical error*, or when the errors occur in such the information is decoded to an incorrect, but valid, logical state. The number of errors required to transform one valid codeword into another valid codeword is called the code distance. The code distance, d , can be determined by the minimum number of errors the code can correct, t , as follows:

$$d = 2t + 1$$

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor$$

Most often, codes are described by $[[n, k, d]]$ using the number of bits in a codeword (n), the number of logical bits we are encoding (k), and the distance of the code (d). In the case of the three-bit repetition code, it would be described as $[3, 1, 3]$ since we use 3 bits to encode a single logical bit, and 3-bit flip errors result in the transformation of one valid codeword to another. What should be made clear here is that the ability to accurately correct errors depends entirely on the error correction code being employed and the specific environmental factors in which noise, and therefore errors, are introduced. If we can guarantee that the probability for an error to occur, p , is incredibly low ($p \ll 0$), then the likelihood of at least two bit-flips (p^2) occurring, resulting in a logical error, could be considered negligible. In this case then, perhaps the three-bit repetition code with majority voting would suffice. However, for most practical cases, the three-bit code is rarely sufficient enough for robust classical error correction and certainly not for correcting errors in quantum computers. Yet still, the core concepts and descriptors introduced here are key in understanding the backbone of QEC.

2.2 Classical Bits to Quantum Qubits

In transitioning from the realm of classical to quantum, it is first important to establish the differences in the fundamental errors we are attempting to correct. As mentioned above, classical computers operate on binary bits of 1's and 0's, and the only error that can occur is a logical bit-flip in which a 0 becomes a 1 or vice versa. Moving into the quantum realm, we now work with quantum bits, or qubits, as the unit of information with a quantum computer. Qubits are two-level quantum systems, with a measurement in the computational basis yielding a 1 or 0, much like a classical bit. However, prior to measurement, the qubit's state can exist in an infinitesimal

continuum, or superposition, of these two states. An individual qubit's state, $|\psi\rangle$, is commonly represented in terms of this computational basis of the states $|0\rangle$ and $|1\rangle$ as follows:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (2.1)$$

Where α and β are complex numbers that represent the probability amplitudes of each of our orthonormal computational basis states [25]. Per Born's rule, the likelihood of measuring a qubit in a specific state is proportional to the square of the probability amplitude of the associated state. Given that the resulting value is representative of a probability of measuring a possible quantum state, the sum of the squared amplitudes associated with each basis of a quantum state is equal to 1:

$$|\alpha|^2 + |\beta|^2 = 1 \quad (2.2)$$

What should be evident at this point is that we are no longer strictly working with deterministic outcomes (e.g., the ON/OFF state of a transistor) but instead with probabilistic outcomes governed by complex wave functions. For instance, starting with a one-qubit system initialized in the ground state, $|\psi\rangle = |0\rangle$, we can create a state in superposition by applying a Hadamard (H) operation:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H \otimes |\psi\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle$$

Upon measuring $|\psi\rangle$, the states $|0\rangle$ and $|1\rangle$ will be measured with an equal probability of 50% (since $|\alpha|^2 = |\beta|^2 = |\frac{1}{\sqrt{2}}|^2 = 0.5$). As with classical bits, a bit-flip error, or often referred to as an X-error, can occur causing a $|0\rangle$ to become a $|1\rangle$, for instance.

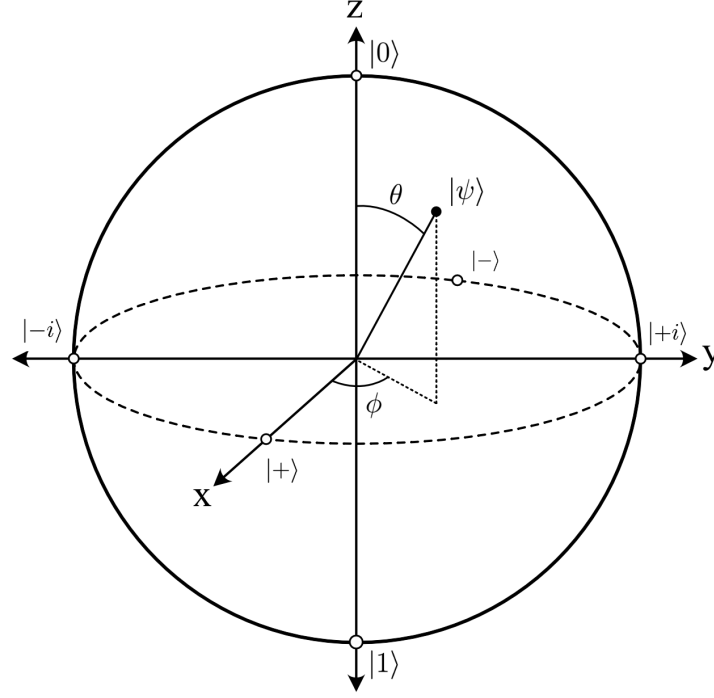


Figure 2.1: The Bloch sphere is visualization of a two-level quantum system, such as a qubit, and is frequently used in quantum information science to visualize how quantum operations, such as the Hadamard operation, impact the qubit's state, $|\psi\rangle$.

However, what might not be evident is one additional source of added complexity when it comes to correcting quantum errors, which is a phase-flip. Another way of representing the quantum state, in which phase is given a more notable representation, is via the Bloch sphere, which visualizes a quantum state as a point on the surface or within a three-dimensional unit sphere. The general geometric formula for a quantum state can thus be written as:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle \quad (2.3)$$

In this formula, the relative phase is determined by $e^{i\phi}$. As such, a phase-flip, or Z error, changes the phase of the qubit. For instance, if we have a two-qubit system in the state $|\psi\rangle = \alpha |00\rangle + \beta |11\rangle$, a Z-error would result in a phase flip on the $|11\rangle$, which changes the sign change + to -, like such:

$$Z \otimes |\psi\rangle = Z \otimes (\alpha |00\rangle + \beta |11\rangle) = \alpha |00\rangle - \beta |11\rangle$$

While the precise importance of relative phase in regards to quantum computing will be omitted for brevity, it's important to know that we now must consider both errors in relating to bit-flips and phase-flips.

2.3 Quantum Error Correction

Quantum computers, by utilizing phenomena like superposition and entanglement, offer the possibility of solving problems previously thought to be intractable, such as prime factorization [26]. These same features that motivate the development of quantum computers also make quantum information incredibly difficult to preserve and precisely manipulate. As the development of classical computers advanced, the optimization of transistor technology has resulted in incredible resilience to errors at the lowest-level. However, to assume that simply developing better qubit technologies alone will reduce the error rates to a level comparable to classical devices is far too hopeful. For one, regardless of qubit modality (e.g., trapped ion, superconducting, neutral atom, etc.), physical qubits and the operations performed on them are incredibly noisy, due in part to the level of precision needed to manipulate quantum environments with sufficient fidelity. As the number of qubits and circuit depth increase, these errors compound and accumulate much more significantly than in the classical realm. This difficulty in manipulating inherently sensitive information motivates the development of quantum error correction (QEC) codes, which provide a means to protect quantum information so that reliable, large-scale computation can be performed despite the imperfect, noisy physical hardware.

Despite the high resilience of modern transistors, the regime of classical error correction serves as the basis for its quantum dual. However, classical error correction relies heavily on copying information and intermediate checks of stored information, two concepts that are seemingly incompatible with two core principles in quantum mechanics - the “No Cloning Theorem” and quantum

state collapse upon measurement. The “No Cloning Theorem” [22] states that, given two quantum states, there is no way to identically copy the information in one qubit onto a second qubit. Secondly, while a qubit can exist in any infinitesimal superposition of $|0\rangle$ and $|1\rangle$, the process of measuring that qubit results in a collapse to a discrete value, destroying the quantum state in the process. These two concepts, that quantum information cannot be simply copied and that measurement is destructive, are a vital component in understanding the why behind quantum error correction.

The core tenet of QEC, like its classical counterpart, is to protect quantum information by redundantly encoding logical information across many entangled physical qubits, resulting in a robust abstract logical qubit. To clarify, the physical qubit is the hardware-level object that makes up the qubit while the logical qubit is the abstract computational unit that we want to preserve. The logical qubit is not stored in any one physical qubit, but instead is delocalized across a variable number of physical qubits. As information is encoded across many entangled physical qubits, a series of repeated measurements on specific measurement qubits can be performed in such a way that detects if errors occurred in the logical qubit while preserving the overall quantum state.

2.4 The Surface Code

The surface code [8] (SC) is one of the most promising QEC codes for physical realization, especially with superconducting qubits, due to its implementation using a 2D nearest-neighbor qubit layout, and its high tolerance to noise [8]. Recent demonstrations on superconducting circuits have allowed logical qubits to be demonstrated and logical error rates (LER) to drop below threshold [1]. SCs can implement a universal set of gates, create logical entanglement with lattice surgery [18], and generate magic resource states with magic state distillation [19, 12]. In this work, we focus on the rotated SC (i.e., any further mention of SC is assumed to be this), requiring $2d^2 - 1$ physical qubits per logical qubit. Our goal is to study the SC’s ability to store logical qubit states over time while being subject to various noise models.

An illustration of a $d=5$ SC patch can be seen on the left side of Fig. 2.2. In this picture, d^2 data

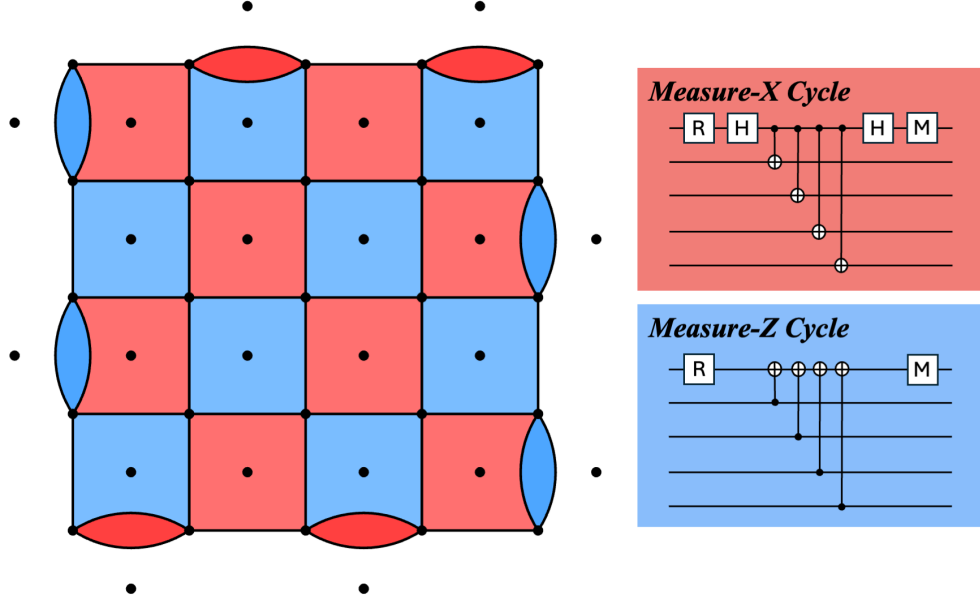


Figure 2.2: An example visualization of a $d=5$ rotated surface code lattice (left) where data qubits are found on the grid boundaries, and X (Z) measurement/stabilizer qubits are found within the red (blue) squares of the grid. The sequence of operations (right) for a single Measure-X and Measure-Z surface code cycle are also illustrated.

qubits are found on the edges of the grid while $d^2 - 1$ measurement qubits are found within the red and blue squares representing Measure-X and Measure-Z stabilizers, respectively. The right side of Fig. 2.2 shows the sequence of operations performed during the SC cycle that allows the data qubits (bottom four lines) to interact with the measurement qubit (top line). Over many cycles, the history of detection events gives insight into the errors happening during runtime. These syndromes are then used by the decoding algorithm, minimum-weight perfect matching (MWPM) [6] in our case, to locate the error within the logical qubit so that it can be corrected. A logical error is introduced into the SC when at least $d/2$ errors occur along a non-trivial path.

2.4.1 Homogeneous vs. Heterogeneous Assumptions of Noise on Surface Code Lattices

In general, if physical operations on the SC are performed below a sufficiently low critical noise threshold then logical error rates are exponentially suppressed with increasing code distances [8]. This scaling can be expressed by the relation:

$$\varepsilon \propto \left(\frac{p}{p_{thr}} \right)^{\frac{d+1}{2}} \quad (2.4)$$

where ε is LER, p_{thr} is threshold for physical noise tolerance, and d is the distance of our code [1]. Prior work has found $p_{thr} = 0.0057$ [8, 1]. In this general form relation, p represents the physical error rate of all $2d^2 - 1$ physical qubits that encode our singular logical qubit in a given lattice. Thus, this relation assumes uniform and homogeneous physical noise exhibited across the lattice. However, results from real quantum devices have shown that superconducting quantum devices do not exhibit uniform noise throughout the processor, but instead vary greatly in qubit fidelity [20, 30].

2.5 Previous Work

The heterogeneous nature of superconducting qubit physical qubit error has been well-documented [3, 21, 4]. Prior work has proposed novel architectures, such as those based on chiplets [27], to promote device scaling with more uniform metrics, but it is unlikely that quantum systems comprised of imperfectly fabricated processors will ever demonstrate perfectly homogeneous noise. This raises many challenges in terms of how to design QEC strategies that best meet the needs of algorithms while managing realistic constraints of hardware.

Previous work with the SC has made progress in combating the reality of defective qubits by leveraging “super stabilizers” and boundary deformations in the SC patch. These techniques work by routing the stabilizer measurement circuits so that they avoid faulty components [17, 5, 32]. Current deformation methodologies incur a tradeoff: deform the lattice around the defect, thus removing the defect’s negative impact on the LER *but* incur additional resource overhead in the process. This increase in overhead is typically due to a higher physical qubit requirement, but some deformation techniques also incur increased complexity during compilation or runtime through specialized steps to facilitate computation within the deformed lattice.

Most work in SC defect management assumes a rigid categorization of qubit quality in the

sense that a qubit is either operating at the mean physical error rate or it is completely unusable and should be dropped out of the logical qubit mapping. However, more lenient bounds of acceptable defectiveness could exist, especially in the case where the distribution of physical qubit error rates is characterized by a standard deviation. For example, the work in [20] reports that the current variation in coherence time seen on existing superconducting qubits leads to hindered LER with the SC. This presents the question of how defects should be defined, and which qubits with higher error rates can be left in the lattice in its current state, given the QEC designer's choice of code distances. These sorts of estimates remain to be fully explored, with early efforts appearing in Carroll et al. [3], which examined how logical error rates for subsystem surface and compass codes on a heavy-hex lattice depend on non-identical infidelity distributions and the presence of outlier qubits with significantly higher physical error rates. Our work validates and extends this type of analysis for rotated surface codes, and uses it as a foundation for developing the boundaries of acceptable defectiveness in the presence of non-uniform qubit fidelity.

CHAPTER 3

Simulation Framework

The tool developed as part of this research is, at its core, a quantum circuit generator with granular per-qubit physical noise control on rotated SCs using Stim [11]. The circuits generated are memory experiments, in which the goal is to identify how well a logical observable is preserved over time. In essence, a memory experiment involves initializing a logical qubit in a known state, performing repeated rounds of syndrome extraction, and measuring the logical observable [9]. If no logical errors occur, the final measured logical observable at the end of the circuit will be the same as the initial logical observable. Memory experiments are a reliable benchmark for stabilizer codes, like the surface codes, in determining how logical computation is affected by characteristics like code distance and how physical noise affects our logical computation. Stim is specifically designed for rapid design and simulation of quantum stabilizer circuits, and the tool presented here uses Stim to generate dynamic noise memory experiments [10]. While other popular quantum circuit simulators exist [14, 7] Stim was selected because (1) the ability to adaptively create entire stabilizer circuits from string manipulation and (2) it offers very fast runtime allowing for sampling the logical error count from millions of shots in minutes. For example, in our distributed heterogeneous noise simulations in Section 5.3, the simulations (Fig. 5.8 and Fig. 5.9) consist of nearly $4 * 10^7$ total shots over the eight distances sampled for each, with each point representing a generated circuit with a unique noise distribution with $5 * 10^4$ shots. Thus, it was important that the simulation tool be reasonably fast in order to test a wide variety of different input parameters.

The circuits are generated with a variety of input parameters to facilitate testing variations

in both individual qubit noise (e.g., specific defect location and defect physical error rates) and overall lattice noise distributions (e.g., uniform or heterogeneous). Our code works in three parts (1) generating the parameterized Stim string representation of the Z-memory experiment circuit, (2) parsing the simulation data and performance results provided by Stim into usable DataFrames, and (3) directly plotting simulation data into easy-to-understand plots that allow for immediate comparison of performance across any number of code distances. Each of these circuits is built using either a uniform (homogeneous) or normally distributed / defect-injected (heterogeneous) noise distribution.

3.1 Building Circuits as Stim Strings

For a given distance d , the generators first assign each of the $n = 2d^2 - 1$ physical qubits a coordinate pair (X, Y) within a grid representing the 2D lattice. The set of values for both the x-coordinate and the y-coordinate starts at 0.5, increments by 0.5, and goes up to $(d + 0.5)$.

In SCs, each qubit is either a measurement (stabilizer) qubit or a data qubit [8]. Within our generators, data qubits are always located at integer coordinates (e.g., (3, 5)) and measurement qubits are always located at non-integer coordinates ± 0.5 in x or y direction from the data qubits (e.g., (.5, .5) or (1.5, 2.5)).

Because Stim strings represent the stabilizer circuits in a manner similar to quantum circuit diagrams [22], each of the n qubits is referred to by a wire index from 0 to $n-1$. Our generators map coordinate pairs to the appropriate wire index within the Stim string, allowing for the generators to accept coordinates as inputs for specific noise models and then appropriately translate between the index and coordinate references for any given qubit. Additionally, the generators will map special location keywords to coordinates on the lattice, as seen in Fig. 3.1. Since the distance directly determines the grid space, these special keywords facilitate placing defects at the same relative location in the lattice, simplifying cross-distance performance analysis.

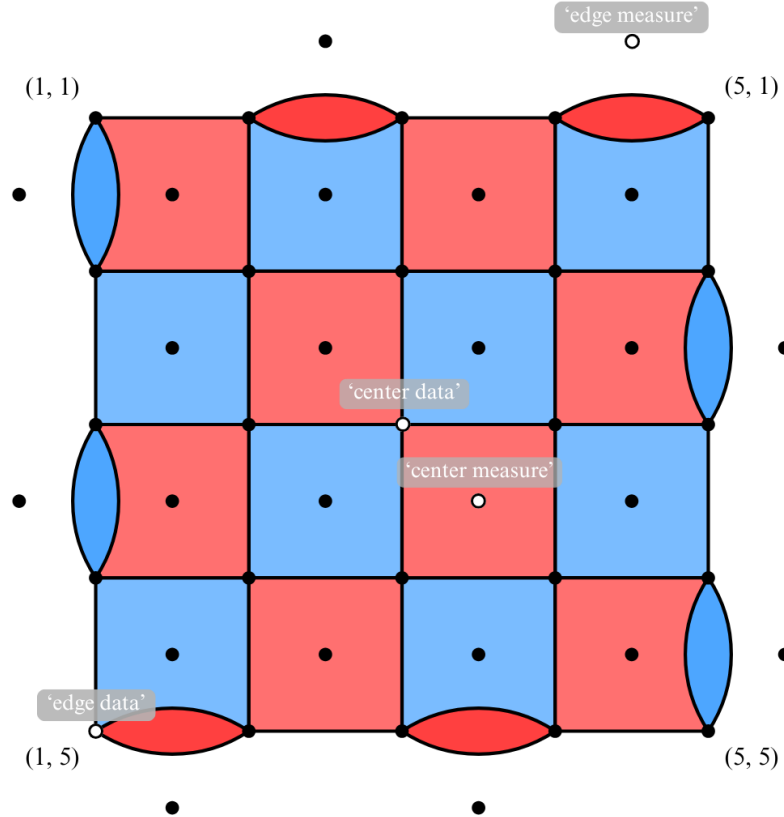


Figure 3.1: For the simulations and selection of defect locations, the generators optionally allow for specific qubit locations to be designated in order to maintain similarity across various code distances. The qubit locations are marked by the special case strings, ‘center data’, ‘center measure’, ‘edge data’, ‘edge measure’, which are then converted to an (X, Y) pair by calculating their respective coordinate for any given distance, d . For example, for a $d=3$ lattice, the center data qubit would be located at $(2, 2)$ while the same location on a $d=5$ lattice would be $(3, 3)$. Instead of needing to specify the exact coordinate pair for each distance, the ‘center data’ string can be provided to the circuit generator.

3.2 Types of Distributions

3.2.1 Homogeneous Distributions

The homogeneous generator takes as input a physical error rate p that will be set as the standard physical error rate for all qubits in the lattice. This represents the idealized uniform assumption of noise across a lattice.

3.2.2 Heterogeneous Distributions

As discussed earlier, a primary motivation for this research is to provide methods for exploring, simulating, and analyzing noise models that more accurately resemble real superconducting devices [20, 1]. To achieve this, the heterogeneous generator creates a circuit that can be used to model the performance of a non-perfect lattice with individual qubits' physical error rate spanning any arbitrary distribution of noise values, rather than a static p .

In this first iteration of our code, the heterogeneous generator defaults to a truncated Gaussian distribution. For each qubit in a lattice, its physical error rate is randomly sampled from a truncated normal distribution created based on two input parameters: a mean physical error rate, p_μ , and either standard deviation, p_σ , or a scalar, α , which determines p_σ as such:

$$p_\sigma = \alpha * p_\mu \quad (3.1)$$

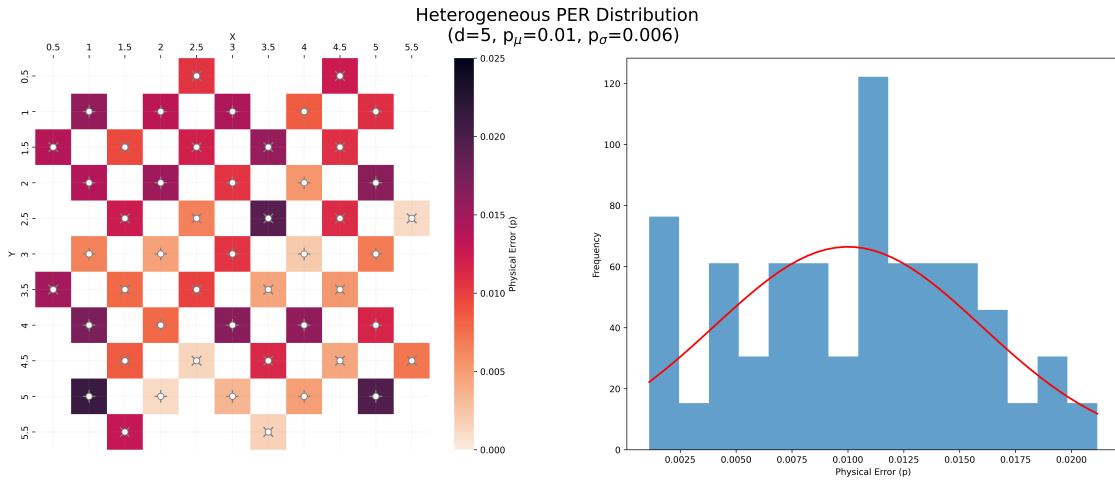


Figure 3.2: A heatmap and distribution representing a randomly distributed heterogeneous noise distribution on a $d=5$ code. The distribution on the right is shown to demonstrate that, although the truncated normal distribution results in slight differences, the samples still form a distribution around p_μ and p_σ . For our experiments in Case 3 (Section 5.3) and Case 4 (Section 5.4), we utilize an α parameter and the actual sampled distribution mean to ensure that any shifts from the input mean are reflected (i.e., when $p_\sigma > p_\mu$ at sufficiently small p_μ values). In this figure, with a $p_\sigma = 0.006$ and $p_\mu = 0.01$, the α parameter would be $\alpha = 0.6$, given that $p_\sigma = \alpha * p_\mu$. Within all heatmaps shown, data qubits are denoted by $+$ and measurement qubits are denoted with $\mathbf{X}$.

Each time a circuit is generated with a given input p_μ and p_σ , a distribution following these

parameters will be randomly sampled from, creating unique distributions of heterogeneous noise between circuit generations, as seen in Fig. 3.2 for a $d=5$ code. This unique distribution for each generated circuit is, in part, a modest reflection of how the noise exhibited by superconducting qubits can greatly fluctuate during or between runtime, but introduces some variability in our simulation’s performance results that leads to some trends being non-monotonic, something that is discussed in Section 5 and Section 7 later on.

It is important to note that noise distribution on current superconducting devices does not necessarily follow a Gaussian model [29]. Thus, as a preliminary method for analyzing the impact of increasingly distributed noise in a lattice, analyzing how the SC reacts under a Gaussian distribution is an ideal starting point, and our parameterized modeling allows for other distributions to be explored in future work.

3.2.3 Defective Qubits

Both homogeneous and heterogeneous circuit generators accept a number of parameters, including distance, rounds, maximum shots. More importantly, the generator methods allow individual defect(s) to be arbitrarily created at any part in the lattice by providing a qubit’s coordinate (x, y) pair within the lattice (as seen in Fig. 3.1) and a physical error rate for each defect, p_{def} .

3.3 Operational Noise

For this initial foray into heterogeneous noise models, we adopt a simplified notion of noise. Specifically, our memory experiments primarily use depolarizing-channel (D) noise, a standard quantum noise model in which a qubit is depolarized with probability p [31]. Applying depolarizing channel to a qubit means that a qubit’s state is left unaltered with probability $1 - p$, and an equal probability of $\frac{p}{3}$ that either a Pauli-X, Pauli-Z, or Pauli-Y will be applied. In most contexts, p is treated as the overall physical noise level of a qubit, encompassing more than just depolarization [8]. In our simplified model, however, noise enters only through Pauli-X (bit-flip) errors on state preparation and measurement and depolarizing errors on CX gate operations. In reality, quantum

noise can be introduced by a wide variety of sources, such as interactions with the environment, uncontrolled interactions with other qubits, imperfect operations, or leakage [24]. Additionally, noise can be introduced permanently on specific qubits as a result of manufacturing defects on the superconducting devices [2]. Implementing a more complex noise space is an area that could be greatly extended upon and is discussed further in our future work section.

In our generators, noise is introduced right before circuit elements (e.g., gates and measurements) as a noise operator with a given probability of being applied, p . Each qubit is assigned an individual error rate, p_i , where i is the index of the qubit. p_i determines the error rate for all operation involving this qubit. These operations include all multi-qubit operations (“DEPOLARIZE2”), single-qubit operations (“DEPOLARIZE1”), and measurements (“X-ERROR”) on the qubit. The p_i is scaled based on single or two-qubit operations. Generally, multi-qubit operations are more prone to errors than operations involving only a single qubit. Thus, it should be reflected that CX operations involving the defective qubits would exhibit a higher error rate than their single-qubit operations. Thus, for two-qubit depolarization noise associated with CX operations, the error rate is set as a mean of the p_i corresponding to the two qubits involved in the operation. Since two-qubit operations are more prone to error than single-qubit operations [1, 20], two-qubit noise has an additional scaling factor, s , on the $q_i + q_i$ mean, in the form:

$$p_{CX} = \left(\frac{p_{q1} + p_{q2}}{2} \right) * s \quad (3.2)$$

This scaling factor is set as $s=1.2$, representing a 20% increase over single qubit operations. For single-qubit depolarization and Pauli-X, the value for each operation on a qubit q_i is set directly to its corresponding p_i value. The process of error decoding is handled by the Stim package, which utilizes the minimum weight perfect matching (MWPM) algorithm provided by PyMatching [13] as the decoding algorithm. In MWPM, the most likely error chain is identified using the syndrome detection events simulated by Stim to minimize the graph’s total edge weight. MWPM is a popular polynomial-time algorithm for decoding SC errors due to the SC having a nearest-neighbor connection which results in graph-like errors.

CHAPTER 4

Boundaries of Acceptable Defectiveness

At this point, we once again return to the idea of the “*Boundaries of acceptable defectiveness*”, or BADs. For each code distance, the boundary of acceptable defectiveness (BAD) defines the maximum physical noise tolerance after which the LER surpasses the ε_{target} . Within the figures for each of our case experiments, the BAD for each code distance is precisely at the intersection point between the target ε_{target} and the distance’s performance curve. Once the physical noise level has surpassed the BAD, we can consider the lattice to be effectively too noisy and the computation unusable. Broadly speaking, the BAD can be viewed as a function of a processor’s distance, unique noise characteristics, and the selected ε_{target} :

$$BAD(p) = f(\text{distance, noise configuration, } \varepsilon_{target})$$

Importantly, these boundaries of acceptable defectiveness for a given distance are not held constant across different models of noise. Thus, using these boundaries provides a meaningful way to quantitatively compare performance across various noise models. Using our tool, the boundaries of acceptable defectiveness (BADs) are provided by determining the intersection point of the logical performance curve and the targeted logical error rate. More formally, we can define I as the piecewise linear interpolant through the set of points for each sample, and let L be the horizontal line representing the targeted logical error rate. Then, for a given sample, we define the BAD as the solution of $L(x) = I(x)$.

CHAPTER 5

Case Studies and Analysis

To demonstrate the performance disparity between heterogeneous and homogeneous noise assumptions, we examined how LER scales across code distances 3-17 when assuming a homogeneous vs. heterogeneous noise model. In each case, the logical error rate per round, ε_{target} , is the metric used to evaluate performance. Every circuit generated for these memory experiments contains 3 rounds of syndrome extraction followed by a measurement in the Z-basis. A shot refers to one complete run of the circuit. The logical error rate for each shot is considered as the average number of errors that occur per round in the measurement circuit:

$$\varepsilon_{round} = \left(\frac{errors}{shots * rounds} \right) \quad (5.1)$$

To facilitate comparison and benchmarking across the four cases, we establish a target logical error rate, ε_{target} , of 0.005. The ε_{target} to use as a baseline depends primarily on the fidelity required to support a specific algorithm or application on a particular quantum device. With Shor's algorithm, for instance, past literature has estimated a $\varepsilon_{target} \in [10^{-15}, 10^{-14}]$ in order to reach sufficient fidelity to achieve an acceptable chance of success [8]. For our purposes, we assume that a $\varepsilon_{target} = 0.005$ is considered to be sufficient fidelity for the generated memory experiments. While a $\varepsilon_{target} = 0.005$ is significantly higher than in practical use cases such as Shor's algorithm, it reflects a realistic logical error rate achievable with emerging hardware as we approach the early fault-tolerant era [1]. A primary benefit of our proposed BADs framework is that it enables performance analysis to contextualize targeted applications within given hardware configurations

and constraints. As such, these case studies serve as both an analysis reflecting current estimates of superconducting hardware *and* an example of how the BADs framework can be applied to benchmark performance for any given application.

Four primary cases were selected under increasingly complex models of noise: (1) a completely uniform homogeneous lattice, (2) a uniform homogeneous lattice with a single defect in the center data qubit, (3) a heterogeneous lattice, and (4) a heterogeneous lattice with a single defect in the center data qubit. In Cases 2, 3, and 4, we vary a single case-specific parameter and measure performance across code distances, allowing the impact of that parameter on performance scaling with distance to become evident. For Cases 2 and 4, this parameter is the defective qubits physical noise p_{def} ; in Case 3, it is a deviation scalar α that determines how far the physical noise of all qubits deviates from the mean. For each distance and each value of the case-specific parameter, we generate 20 distinct physical error rates p (mean physical error rates, p_μ , in Cases 3 and 4) logarithmically spaced between 10^{-3} and 10^{-1} . For each combination of d , p (p_μ), and case parameter, one circuit is constructed and the performance is simulated over 50,000 shots. The logical error rate plotted at each point is the average logical error rate per round over those 50,000 shots, so each curve shows how a single sampled processor with a fixed parameter behaves as the overall noise level varies.

5.1 Case Study 1: Homogeneous Noise

Case 1 covers the assumption of uniform noise in which all physical qubits within a lattice unilaterally express the same p noise. A heatmap for this uniform homogeneous noise model can be seen in Fig. 5.1. The performance results for Case 1 are shown in Fig. 5.2 for $d=[3, 5, 7, 9, 11, 13, 15, 17]$. In past literature, this most closely resembles the ideal and common notion of how

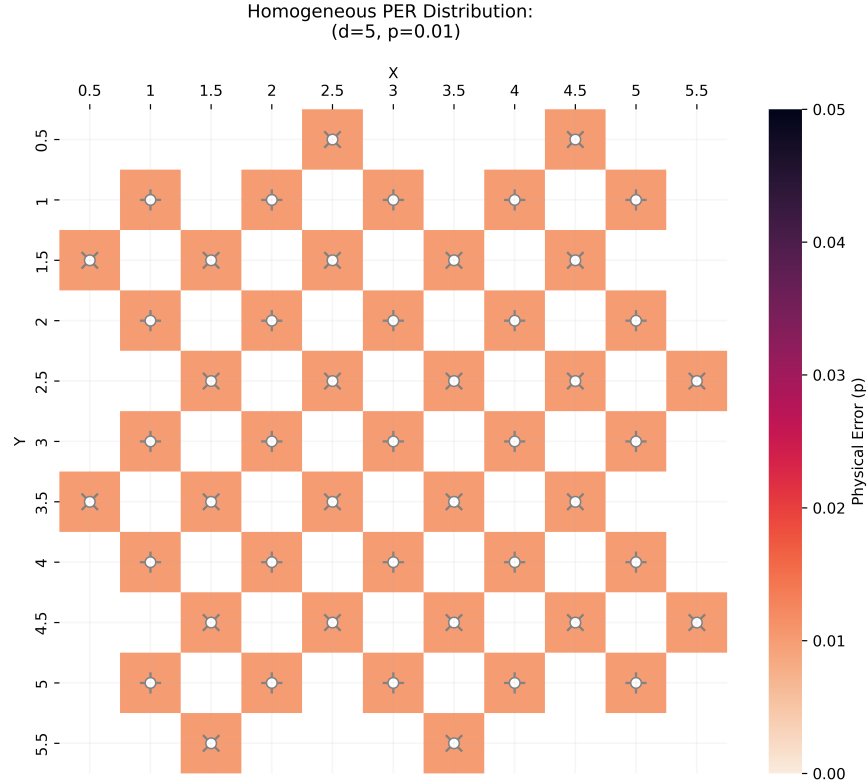


Figure 5.1: (*Case Study 1*) Heatmap for a homogeneous noise model on a $d=5$ code with no defects or deviations. In this heatmap, data qubits are denoted by $+$ and measurement qubits are denoted with X .

noise is represented in SC. As shown in Fig. 5.2, all distances (with the exception of $d=3$) cross the ε_{target} . For the highest distance sampled, $d=17$, the BAD is at $p \approx 0.006$. Any $d=17$ lattice with a $p \leq 0.006$ would, in this theoretical model where $\varepsilon_{target} = 0.005$, be sufficient for computation. However, if our goal is to minimize the code distance and thus our physical qubit overhead, we would select the minimum viable code distance for which our device's expressed p is below the BAD for that distance. In this uniform homogeneous model, the $d=5$ code has a BAD at approx-

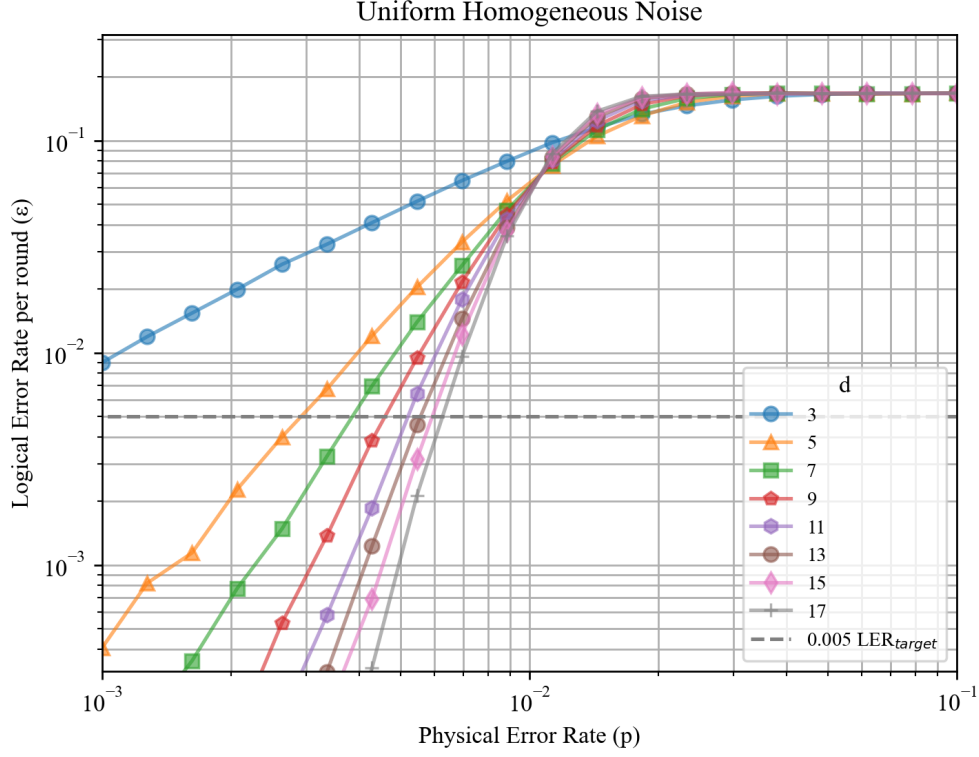


Figure 5.2: (*Case Study 1*) Performance results for a completely uniform, homogeneous noise model across distances $d=[3, 5, 7, 9, 11, 13, 15, 17]$. Each point in the plot represents a single simulated processor instance: for each fixed distance, we generate one circuit per $p \in \{10^{-3+\frac{2 \cdot k}{19}} | k \in \{0, 1, 2, \dots, 19\}\}$ (e.g., 20 points evenly spread in log-space ranging from 10^{-3} and 10^{-1}) and estimate the logical error rate at that point from 50,000 shots of that circuit. Case 1 represents the most ideal noise model assumptions when considering SCs in that all qubits within the lattice have an identical quality of p and no qubit deviates from this value. In Cases 2 and 3, the blue line in each performance result figure is the baseline for each distance when no deviation is expected. As such, each line in this figure is functionally equivalent to the blue baseline in the Case 2 and Case 3 performance results for the corresponding distance. Because errors are probabilistically applied, small differences in these baselines can occur across simulations.

imately $p \approx 0.003$. Thus, if deciding an optimal code distance for a physical device capable of performing computation with physical noise, $p \approx 0.003$, we should select the $d=5$ code. While achieving $p = 0.003$ on a real superconducting device might be out of reach with current hardware [15], analyzing the BADs for a given code distance with a realistic physical noise model allows for additional optimization considerations. In Case 1, performance requirements are met at all but the smallest distance, $d=3$. While achieving uniform noise across all physical qubits results in the highest overall fidelity and the most viable code distances, noise data from superconducting devices exhibit significant variations in noise within a given chip. Thus, this uniform model in Case

1, at least for superconducting devices, is naive and an unrealistic basis for accurately modeling SC robustness [15, 20].

5.2 Case Study 2: Homogeneous Noise with Outlier Defect

In this second scenario, we examine the impact of a single defect on the SC's performance under an otherwise uniform noise model. To test individual qubit impact within a homogeneous noise model, circuits were generated with a defect located at the center data qubit. A heatmap for the homogeneous noise model with an outlier defect center data qubit can be seen in Fig. 5.3. The performance results for Case 2 are shown in Fig. 5.4 for $d=[3, 5, 7, 9]$ and Fig. 5.5 for $d=[11, 13, 15, 17]$.

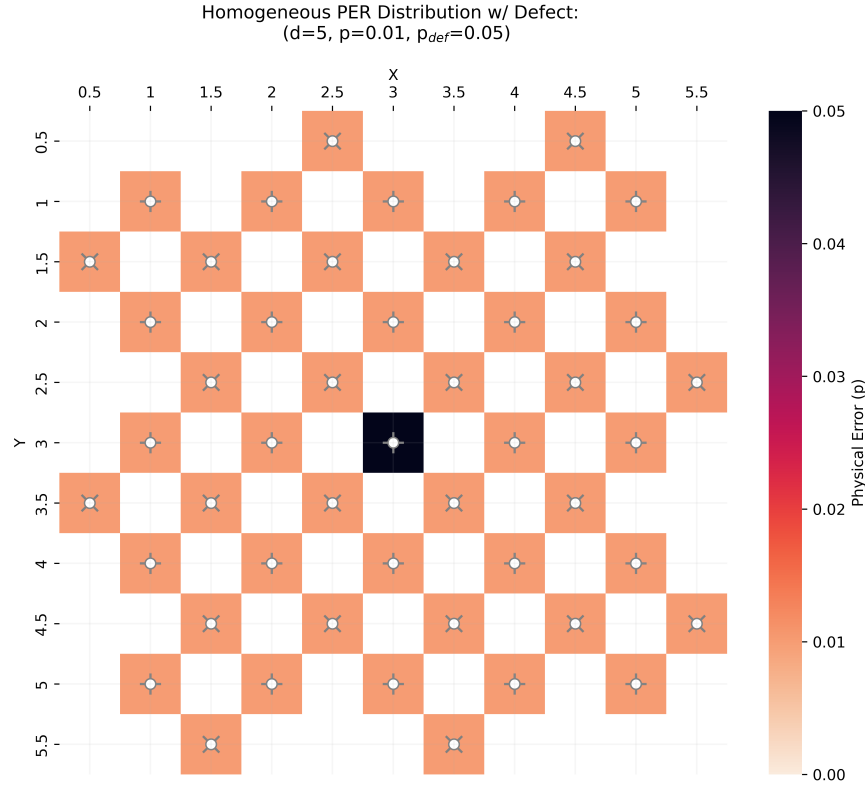


Figure 5.3: (*Case Study 2*) Heatmap for a homogeneous noise model on a $d=5$ code with a defect on the center data qubit. The p_{def} shown in this heatmap is 0.05 to show this distribution, but significantly higher values p_{def} are also used in the Case 2 experiments as well. In this heatmap, data qubits are denoted by + and measurement qubits are denoted with X.

In this experiment, all lattices across the sampled distances have a defect located at the data qubit in the center of the lattice with $p_{def} \in \{0.05, 0.25, 0.5, 0.75\}$. At the upper end of the p_{def} values sampled, the individual qubit is essentially completely unreliable, with roughly 75% of

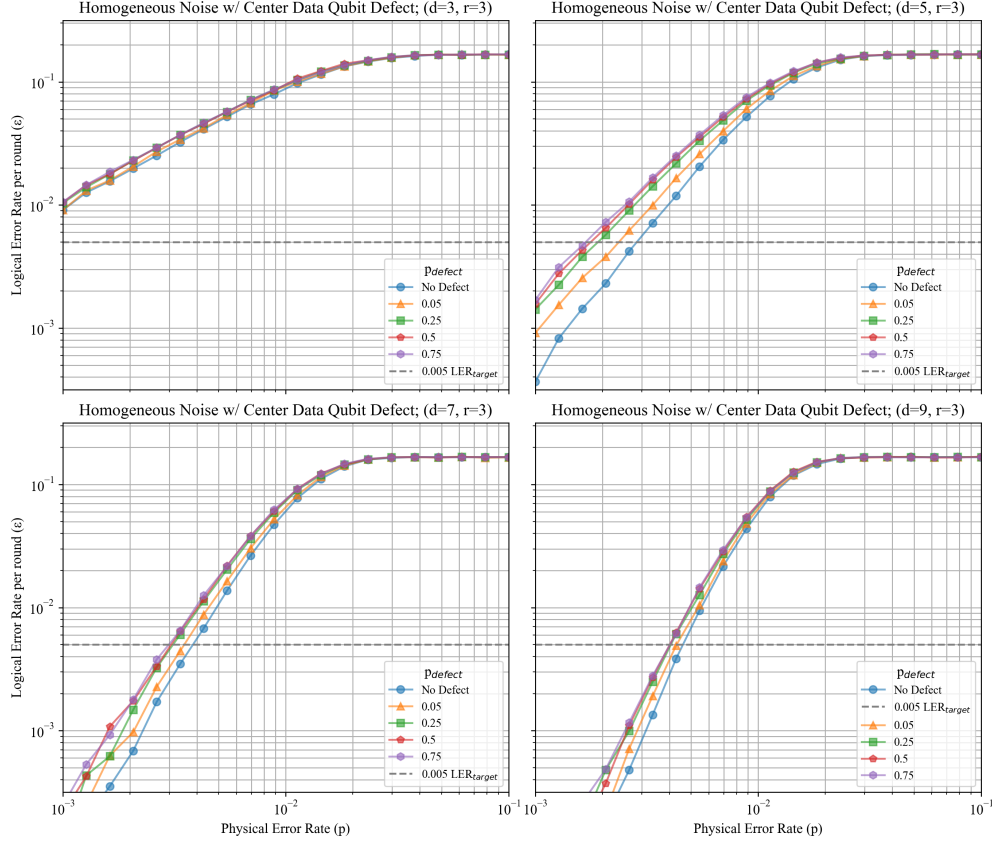


Figure 5.4: (*Case Study 2*) Performance of homogeneous noise model with outlier center data defect over $d=3$ through $d=9$. Each point in the plots represents a single simulated processor instance: for each fixed distance and p_{defect} , we generate one circuit per $p \in \{10^{-3+\frac{2+k}{19}} | k \in \{0, 1, 2, \dots, 19\}\}$ (e.g., 20 points evenly spread in log-space ranging from 10^{-3} and 10^{-1}) and estimate the logical error rate at that point from 50,000 shots of that circuit.

operations on this qubit resulting in an erroneous computation. The results of this experiment (Fig. 5.4, Fig. 5.5, and Fig. 5.6) show an incredibly strong resilience to the individual defect. Similar to that of the unilateral homogeneous model in Case 1 (Fig. 5.2), all but the smallest code distance $d=3$ (Fig. 5.4) were able to meet the $\varepsilon_{\text{target}}$. In the worst case with $p_{\text{def}} = 0.75$, the BAD for the $d=17$ code (Fig. 5.5) is $p \approx 0.0058$. In comparing this to the $d=17$ performance in Case 1 (Fig. 5.2), the maximally defective qubit only reduced our BAD by 3%. For the minimum viable code distance in Case 2, $d=5$ (Fig. 5.4), the BAD is located at $p \approx 0.0017$.

Specifically, for $d=5$ in Case 2 with the BAD for $d=5$ ($p \approx 0.0029$) in Case 1 (Fig. 5.2), the presence of the defect with $p_{\text{def}} = 0.75$ resulted in a larger reduction to our acceptable boundary by about 50%. As with physical noise tolerance, the results here suggest that under a homoge-

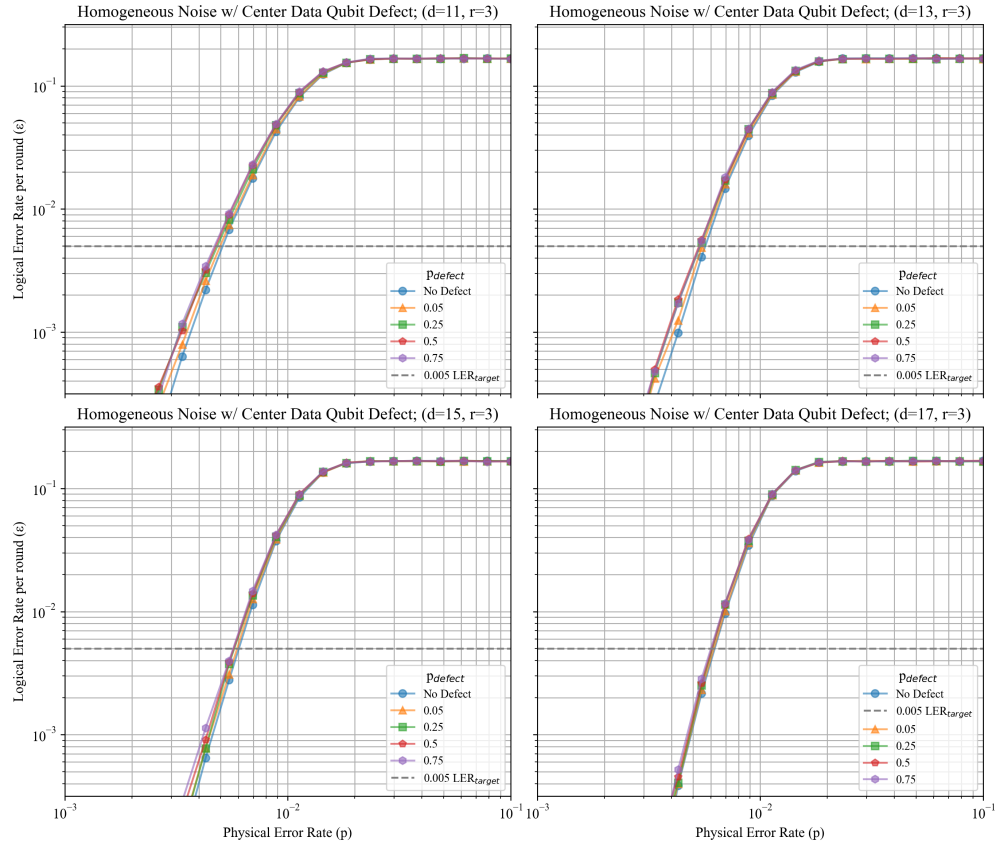


Figure 5.5: (*Case Study 2*) Performance of homogeneous noise model with outlier center data defect $d=11$ through $d=17$.

neous noise model, an increase in code distance also directly results in a significant increase in the resilience to individual defects with high levels of noise, which can be clearly seen in Fig. 5.6.

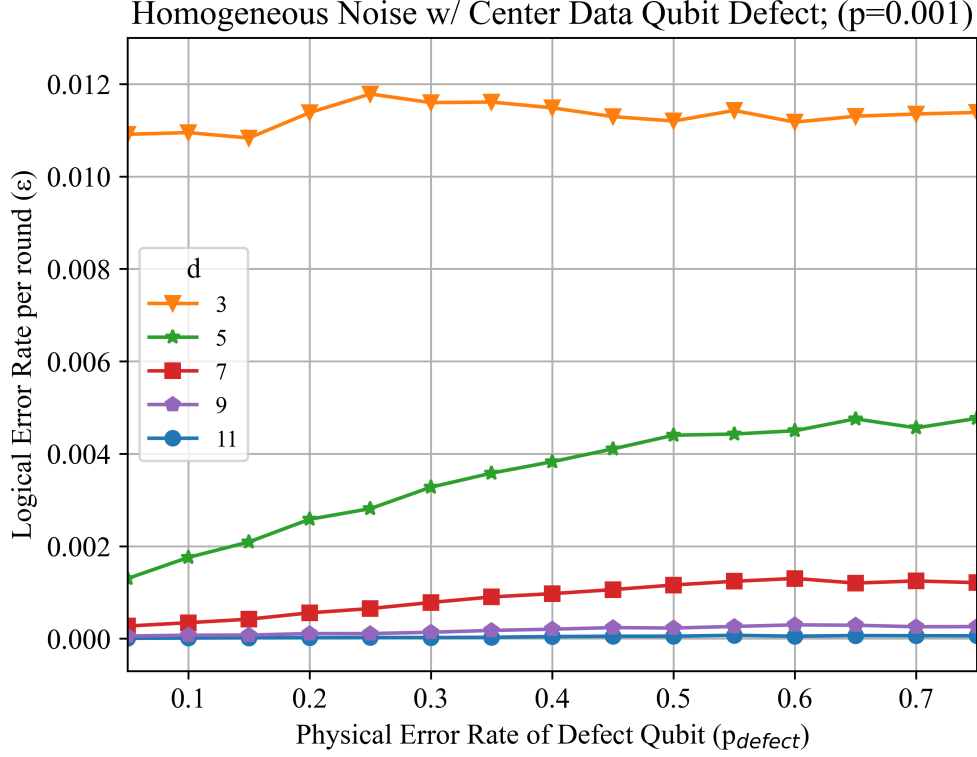


Figure 5.6: (*Case Study 2*) Logical error rate vs. defective physical qubit error rate for code distances $d=[3, 5, 7, 9, 11]$. To further examine how performance is affected across code distances, this figure shows a similar homogeneous noise model to that in Case 2, except that the noise for all other qubits is kept constant at $p = 0.001$. As we increase our code distance, the lattice becomes increasingly resilient to the noise introduced by the defective data qubit, as expected. At distances of 9 or above, the performance is essentially unaffected by the presence of the defective qubit. Thus, this further suggests that the impact of highly defective qubits becomes negligible at a sufficiently large code distance.

5.3 Case Study 3: Heterogeneous (Gaussian) Noise

In the third case, we expand our noise model to a heterogeneous normal distribution of physical errors and examine the impact on the logical performance. A heatmap demonstrating a potential configuration for a $d=5$ heterogeneous noise model can be seen in Fig. 5.7. The performance results for Case 3 are shown in Fig. 5.8 for $d=[3, 5, 7, 9]$ and Fig. 5.9 for $d=[11, 13, 15, 17]$.

For this experiment, we simulate heterogeneity by assigning each qubit a p_i randomly sampled from a truncated normal distribution with an input mean parameter, p_μ , and an input standard deviation, p_σ . All operations involving this qubit in this instance will use the p_i following the same operational noise assignment detailed in Section 3.3. To examine the impact of this Gaussian

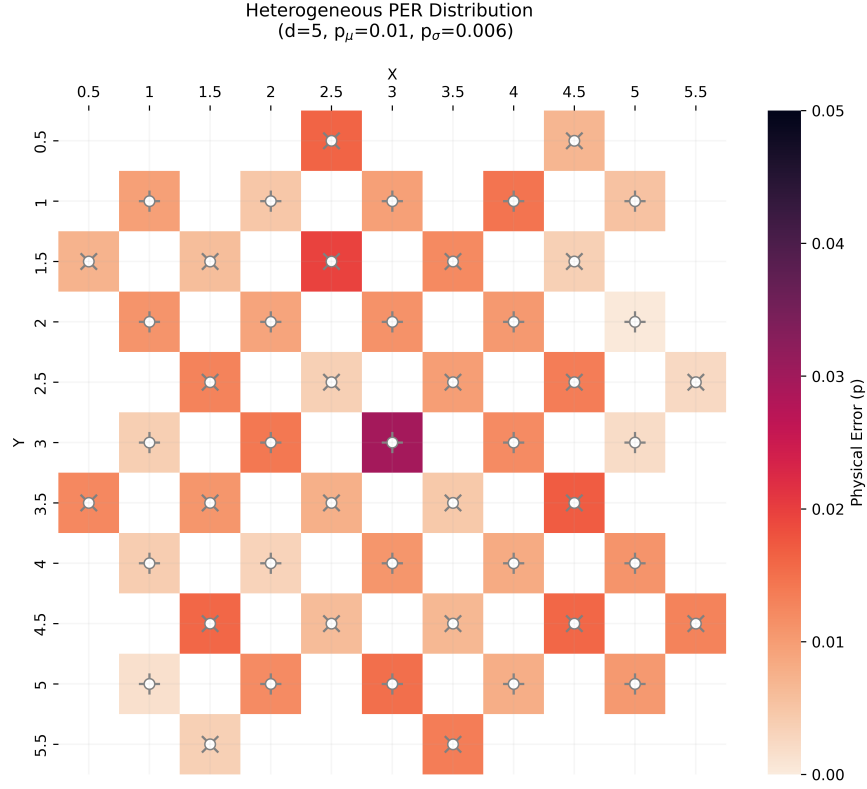


Figure 5.7: (*Case Study 3*) Heatmap for a heterogeneous noise model on a $d=5$ code. Data qubits are denoted by $+$ and measurement qubits are denoted with $\mathbf{X}$.

model of heterogeneous noise on logical performance, we simulated $\alpha \in \{0.01, 0.3, 0.6, 0.9\}$, corresponding to normal distributions with a deviation of 1% to 90% of the p_μ . For each α value examined, we simulated their performance with 20 different p_μ values evenly distributed between 10^{-3} and 10^{-1} on a logarithmic scale. To prevent negative probabilities from being sampled, a truncated distribution is used. When a distribution is truncated, the input mean used to create the distribution, p_μ , can differ from the *actual* mean, $p_{\mu-out}$, of the values sampled from the distribution. Thus, to accurately reflect any shifts between input and actual mean, the $p_{\mu-out}$ is used as the x-axis in our performance plots. For clarity, if p_{q_i} is the randomly sampled physical noise for qubit q_i and $n = 2d^2 - 1$, then the average mean of the sampled distribution is as follows:

$$p_{\mu-out} = \frac{\sum_{i=0}^{n-1} p_{q_i}}{n} \quad (5.2)$$

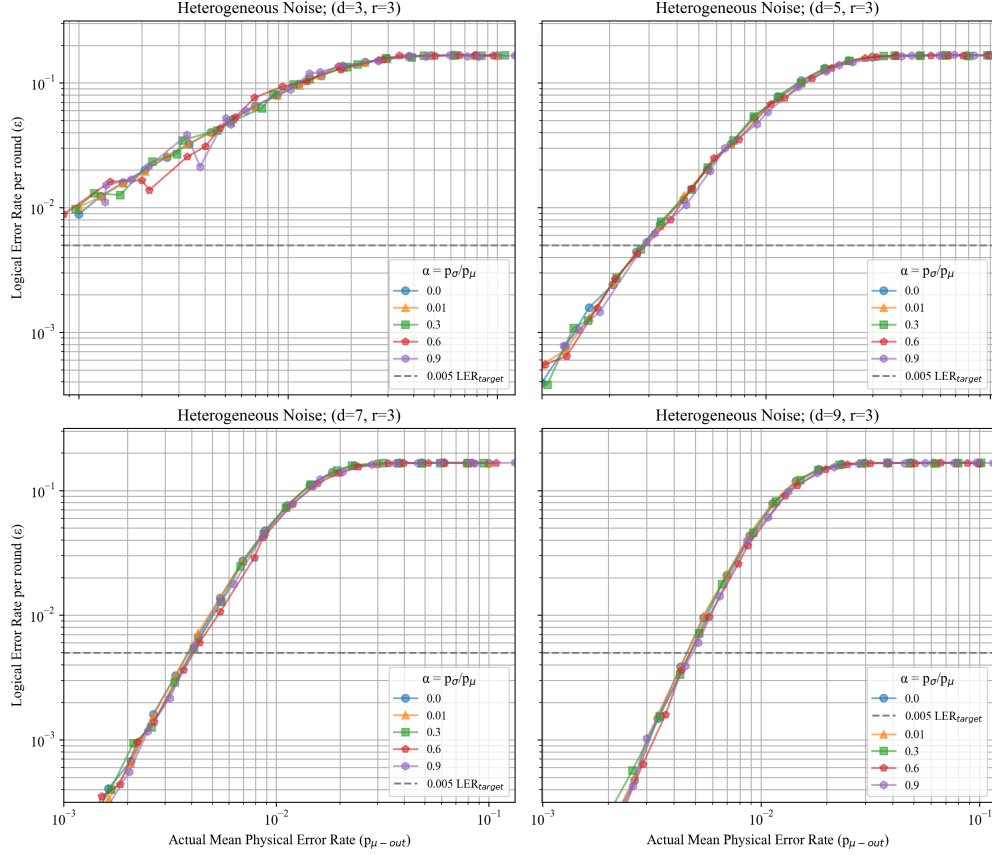


Figure 5.8: (*Case Study 3*) Performance of heterogeneous noise model for distances 3 through 9. The blue line ($\alpha = 0$) represents a unilateral homogeneous noise model. Each point in the plots represents a single simulated processor instance: for each fixed distance and α , we generate one circuit per $p_\mu \in \{10^{-3+\frac{2*k}{19}} | k \in \{0, 1, 2, \dots, 19\}\}$ (e.g., 20 points evenly spread in log-space ranging from 10^{-3} and 10^{-1}) and estimate the logical error rate at that point from 50,000 shots of that circuit.

Similar to Case 1 (Fig. 5.2) and 2 (Fig. 5.4 and Fig. 5.5), the performance of the heterogeneous noise is benchmarked against a uniform homogeneous distribution, which is equivalent to when the $\alpha = p_\sigma = 0.0$. The non-monotonic behavior of the plotted data in Fig. 5.8 and Fig. 5.9 is a result of this randomly sampled distribution, in which some circuits have a higher distribution of physical noise placed on more detrimental locations, such as higher noise on measurement qubits or edge qubits of which have a greater impact on LER [8]. However, the noise and coherence properties of individual superconducting qubits can change frequently [15]. Thus, purely monotonic performance curves when utilizing superconducting qubits is not a guarantee.

This is most notable with $d=3$ in Fig. 5.8 in which the ε_{round} appears to fluctuate more dras-

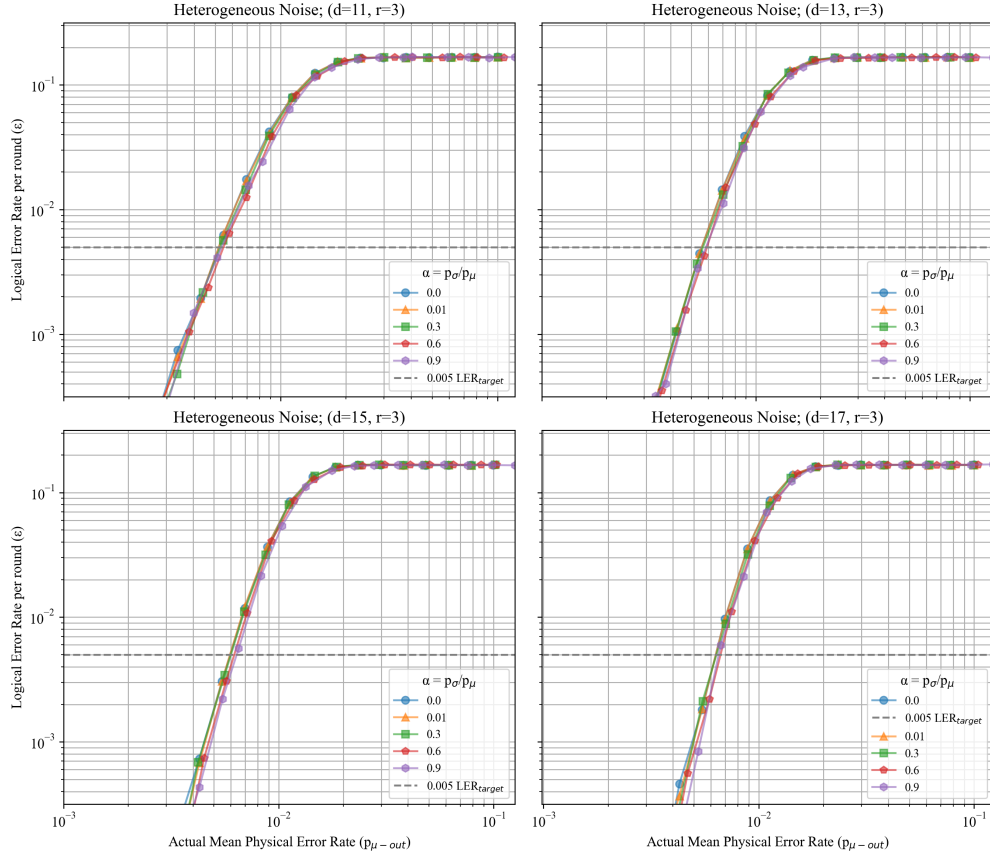


Figure 5.9: (*Case Study 3*) Performance of heterogeneous noise model for distances 11 through 17. The blue line ($\alpha = 0.0$) represents a unilateral homogeneous noise model covered in Case 1.

tically as α increases. In a $d=3$, the proportion of 2-weight stabilizers on the edge of a lattice to 4-weight stabilizers in the bulk of the lattice is higher. With a higher proportion of low-weight stabilizers in $d=3$, the increased deviation leads to greater fluctuations, as shown for $\alpha = 0.9$ and $\alpha = 0.6$ in Fig. 5.8.

5.4 Case Study 4: Heterogeneous (Gaussian) Noise with Outlier Defect

In the fourth case, we examine performance changes under a heterogeneous noise model in the presence of a center data-defect qubit. A heatmap demonstrating a potential configuration for a $d=5$ heterogeneous noise model with a center defect can be seen in Fig. 5.10. The performance results for Case 4 are shown in Fig. 5.11 for $d=[3, 5, 7, 9]$ and Fig. 5.12 for $d=[11, 13, 15, 17]$. The

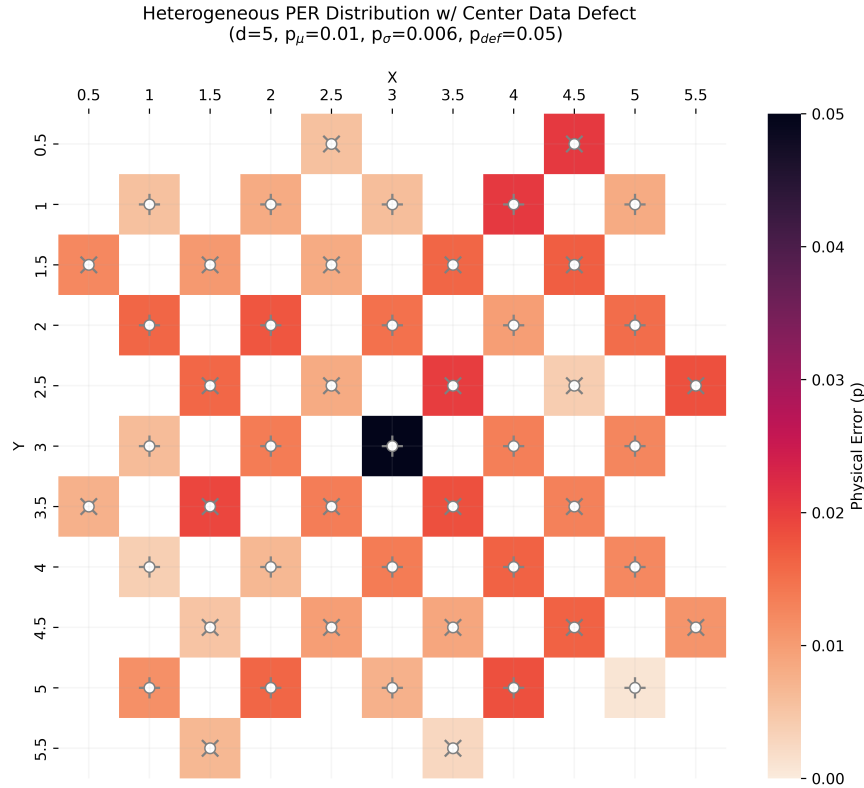


Figure 5.10: (*Case Study 4*) Heatmap for a heterogeneous noise model on a $d=5$ code with an outlier center data qubit. In this heatmap, data qubits are denoted by $+$ and measurement qubits are denoted with X .

location of the defect (center data qubit) and $p_{def} \in \{0.05, 0.25, 0.5, 0.75\}$ is identical to the setup in Case 2, with the exception that the noise of all other qubits is normally distributed rather than constant. As with the setup of the experiments in Case 3, we use the actual mean of the physical noise that was randomly sampled, $p_{\mu-out}$, as the horizontal axis. Additionally, like in Case 2, we want to assess how increasing the noise of all other qubits in the presence of a defect impacts the LER. Thus, our $p_{\mu-out}$ does not include the p_{def} value as it is not sampled from the distribution

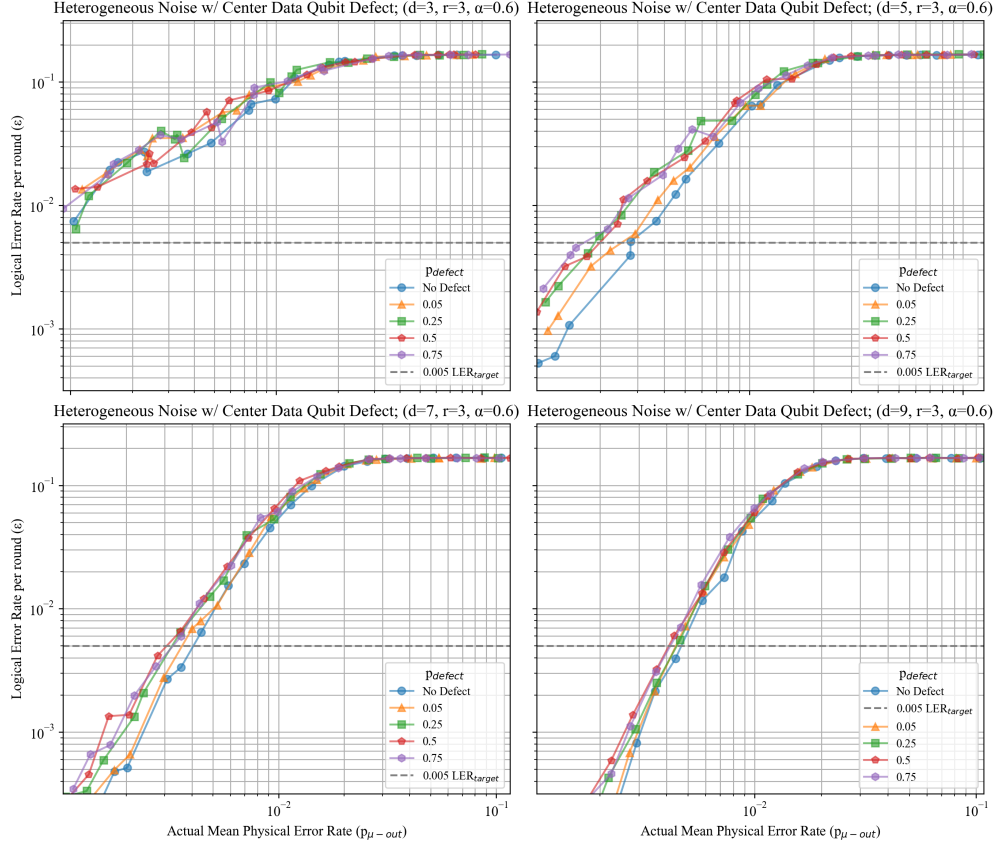


Figure 5.11: (Case Study 4) Performance of heterogeneous noise model with center data defect for distances 3 through 9. In Case 4, we simulated the performance of heterogeneous noise resembling a Gaussian distribution with a single defect at the center data qubit with a p_{def} in the range of 0.05 to 0.75. In all simulations in Case 4, the α parameter, which determines the deviation as a percentage of the input mean, is held constant at $\alpha = 0.6$. The blue line represents a baseline heterogeneous noise model with no defects present, which is the same as the $\alpha = 0.6$ simulations in Case 3. Each point in the plots represents a single simulated processor instance: for each fixed distance and p_{defect} , we generate one circuit per $p_\mu \in \{10^{-3+\frac{2*k}{19}} | k \in \{0, 1, 2, \dots, 19\}\}$ (e.g., 20 points evenly spread in log-space ranging from 10^{-3} and 10^{-1}) and estimate the logical error rate at that point from 50,000 shots of that circuit.

itself. For clarity, in Case 4, the $p_{\mu-out}$ is as follows:

$$p_{\mu-out} = \frac{\sum_{i=0}^{n-1} p_{q_i} - p_{def}}{n - 1} \quad (5.3)$$

For this experiment, we selected a single α used in Case 3, $\alpha = 0.6$, and used the same p_{def} values that were selected in Case 2. Interestingly, the performance trends shown in Case 4 (Fig. 5.11 and Fig. 5.12) follow a pattern similar to that observed in Case 2 (Fig. 5.4 and Fig. 5.5)

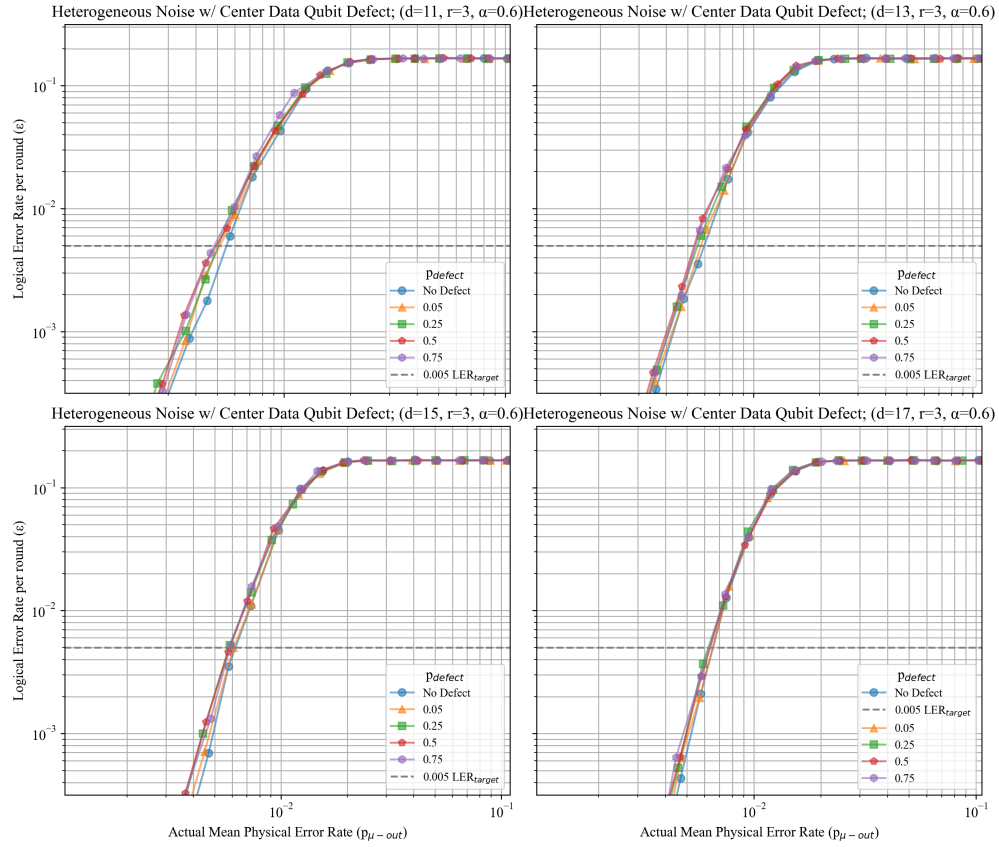


Figure 5.12: (*Case Study 4*) Performance of heterogeneous noise model with center data defect for distances 11 through 17. The blue line represents a baseline heterogeneous noise model with no defects present.

in which higher distances show a strong resilience to increasing magnitudes of p_{def} .

CHAPTER 6

Discussion

Our experiments aimed to highlight how heterogeneity affects performance on the SC, and the results indicate that the presumed model of noise greatly impacts the purported performance of the SC. We proposed in this work that, in order to accurately model SC on superconducting qubits, accepting some level of heterogeneity is necessary due to the inherent non-uniform noise exhibited by superconducting qubits. The aim of our work is thus to provide a preliminary accounting of how the performance of heterogeneous distributions directly affects the LER. As shown in our experiments, the presence of a single defective qubit with an error rate as high as 75% significantly affects the LER only at small code distances and becomes increasingly negligible as the code distance increases.

With regard to the impact of individual defects, the performance data for both a single defect in the presence of heterogeneous and homogeneous noise models show that the defective qubit's impact is only non-negligible at distances at or closely above the minimum viable code distance (the distance at which the target LER is reachable).

In the performance experiments for Case 2 (Fig. 5.4 and Fig. 5.5), the minimum viable code distance occurs at $d=5$. At $d=3$, the target LER is not reachable even when there are no defects present. Although the performance at $d=3$ is insufficient, the performance across the increasing p_{def} noise is nearly identical, indicating that this defect has little impact on the LER. At $d=5$, the trend lines suddenly deviate, and the impact of higher p_{def} noise on performance is noticeable. At $d=5$ through $d=9$, this performance deviation between p_{def} noise begins to converge. At $d=11$ and

above, the impact of an increasingly defective qubit is nearly negligible again. What we can extrapolate from this is that there exists a sort of “Goldilocks zone” in which the detrimental impact of increasing noise on the defective qubit is evident. Within this “Goldilocks zone,” choosing to drop out or mitigate the defective qubit with large p_{def} values could result in noticeable performance gains. At $d=11$ and above, however, the impact of this defective qubit, even at $p=0.75$, is negligible and thus implementing a mitigation technique incurs additional cost, such as an increase in physical overhead [32, 27] or circuit depth [5], but results in no performance gains. Ideally, when making design decisions, techniques that incur additional costs should only be used when there is a noticeable performance gain. In this way, our research suggests that treating defect mitigation should be viewed as an optimization problem in which we aim to only drop out qubits in which the resulting performance gain is non-negligible in light of the cost incurred by the mitigation method.

Regarding the impact of heterogeneous noise that resembles a Gaussian distribution, our results show that the LER is not greatly affected by increasing deviation. In comparing the performance curves in Cases 1 with static p and Cases 3 with normally distributed noise, we see nearly identical trends, indicating the increased deviation had a minimal impact on the LER. In Case 4, the same defect configuration (center data qubit) is tested under the heterogeneous model with a fixed $\alpha = 0.6$. In the performance experiments conducted (Fig. 5.11 and Fig. 5.12), the defect noise reflects a significant outlier from the remainder of the comparatively small normally distributed noise. These results closely resemble the trends discussed in Case 2. When comparing the insights of Case 1 vs. Case 3 and Case 2 vs. Case 4, our results indicate that while evenly distributed distributions may not impact LER, outlier defects much higher than the average physical noise do. The simulations used 3 rounds across all distances, which was done to enable rapid experimenting over many configurations and distances while we were exploring these preliminary models of noise. We performed the same experiments for each case study with rounds=distance. While higher p_{def} values became slightly more noticeable impact on LER at higher distances, the trends discussed for all cases involving both defects and impact of Gaussian noise were the same.

It appears that if noise is normally distributed, the slightly higher fidelity qubits essentially

cancel out their slightly noisier counterparts. However, this balancing requires that for every bad qubit in the lattice, there is an equally good qubit, which is an unrealistic assumption. Further, the presence of a significant defect cannot be balanced and thus has a noticeable impact on the LER. After all, it is much easier to fabricate a noisy qubit than it is to fabricate a perfect qubit. Ultimately, a heterogeneous noise model based on a Gaussian distribution does not appear to capture the complexity or impact of real heterogeneous noise, nor does it accurately reflect what we observe on real devices, further arguing that real quantum noise on superconducting devices cannot be fully represented as Gaussian.

CHAPTER 7

Extensions and Future Directions

Overall, the results of our case studies concur with prior work examining the impact of individual defects and normally distributed heterogeneous noise [3], and expand these insights across a wide range of distances, outlier defects, and distribution deviations. Going forward, we believe that examining the impacts of heterogeneity on SC is vital as both physical devices and QEC techniques advance toward fault-tolerant quantum computing. Building upon the tool and results presented here, we believe there are many areas in which this work could be expanded. A particular area of further development would be to expand the simplified noise model that we use to create our circuits, such as introducing additional scaling factors to more accurately represent the different operational error rates [20]. Second, the experiments focused primarily on testing individual defects on a single data qubit, but expanding these experiments to encompass how additional configurations and patterns of defect characteristics affect performance is an area for future work. Identifying how these defects accumulate would provide further insight into when deformation techniques are absolutely necessary. The experimental results presented in our research demonstrate preliminary evidence showing the existence of BADs as well as the disparity between viewing the SC under a homogeneous model vs. a heterogeneous model. If a primary goal of simulation is to optimize performance for computation on real devices, restricting examinations of the SC to only homogeneous noise models can lead to vastly different design outcomes, such as picking optimal code distances to meet a target logical error rate, that fail to fully utilize the already limited physical resources available in present and near-term quantum hardware.

Secondly, our current analysis in the distributed heterogeneous case studies (Sec. 5.3 and 5.4) focuses on single-processor samples for each parameter setting (e.g., p_{def} or α), where each plotted point corresponds to one simulated processor instance with many shots, rather than an ensemble average over independently resampled processors. As a result, we do not report statistical error bars across heterogeneous instances in this work, due in part to computational resource constraints for this initial study. Another natural direction for future work is to extend these experiments to an ensemble regime in which, for each set of parameters, many independent processors are generated and averaged. Such ensemble-based studies would enable us to quantify variability more adeptly across different heterogeneous noise landscapes.

Finally, these results suggest that modeling heterogeneity with a Gaussian distribution is a poor proxy for the kinds of imbalance observed on superconducting devices. When noise on a lattice is approximately normally distributed, the slightly better-than-average qubits effectively counterbalance the slightly worse ones within a logical patch, and the resulting LER closely resembles that of an idealized homogeneous lattice with a physical error rate equal to the mean. In this sense, a Gaussian model collapses back to the homogeneous assumption, with the p_μ playing the same role as a uniform p in the standard surface-code scaling relation in 2.4.1. However, this same observation also points to a useful design goal: when selecting logical patches and code distance on a fixed, heterogeneous device, one might aim to partition the physical qubits so that each logical patch’s noise distribution is as close to balanced (i.e., approximately Gaussian) as possible, thereby allowing high-fidelity qubits to offset noisier ones and achieve near-homogeneous performance. Since empirical studies indicate that real-device noise is neither perfectly Gaussian nor perfectly balanced [29], this further motivates future work on noise models and mapping strategies that explicitly capture skewed, multi-modal, or otherwise imbalanced error landscapes rather than relying on symmetric Gaussian assumptions.

CHAPTER 8

Conclusion

In this work, we explore the impact of heterogeneous noise on the SC's performance. This work is among the first to study the physical error rate of device qubits as a distribution and the first to provide a framework and preliminary toolset for rapidly evaluating how configurations of non-identical noise affect the LER in rotated SCs. Using the BADs framework, we explore the impact of both outlier qubits and normally distributed physical errors on SC LER. As a result of our work, we discovered that one bad qubit does not necessarily spoil the bunch. Furthermore, we demonstrate that if heterogeneous noise follows a Gaussian distribution, the SC LER scales only with the mean of the SC lattice, not its deviation. This ultimately indicates that modeling heterogeneous noise as a Gaussian distribution is functionally equivalent to treating it as homogeneous, further emphasizing the need to develop more complex models that capture the effects of superconducting qubit deviation. To conclude, this work motivates the inclusion of BADs in QEC analysis based on the SC, and we hope that the software framework developed here will open the door to design space exploration, enabling shorter timelines to fault-tolerant quantum computing.

References

- [1] Rajeev Acharya et al. “Quantum error correction below the surface code threshold”. en. In: *Nature* 638.8052 (Feb. 2025), pp. 920–926. ISSN: 1476-4687. DOI: 10.1038/s41586-024-08449-y. URL: <https://www.nature.com/articles/s41586-024-08449-y>.
- [2] James M. Auger et al. “Fault-tolerance thresholds for the surface code with fabrication errors”. In: *Physical Review A* 96.4 (Oct. 2017), p. 042316. ISSN: 2469-9926, 2469-9934. DOI: 10.1103/PhysRevA.96.042316. URL: <http://arxiv.org/abs/1706.04912>.
- [3] Malcolm S. Carroll, James R. Wootton, and Andrew W. Cross. *Subsystem Surface and Compass Code Sensitivities to Non-Identical Infidelity Distributions on Heavy-Hex Lattice*. Version 1. Feb. 14, 2024. DOI: 10.48550/arXiv.2402.08203. arXiv: 2402.08203 [quant-ph]. URL: <http://arxiv.org/abs/2402.08203>. Pre-published.
- [4] Samudra Dasgupta and Travis S Humble. “Reliable devices yield stable quantum computations”. In: *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 2. IEEE. 2023, pp. 223–226.
- [5] Dripto M. Debroy et al. *LUCI in the Surface Code with Dropouts*. Oct. 2024. DOI: 10.48550/arXiv.2410.14891. URL: <http://arxiv.org/abs/2410.14891>.
- [6] Eric Dennis et al. “Topological quantum memory”. In: *Journal of Mathematical Physics* 43.9 (2002), pp. 4452–4505.
- [7] Cirq Developers. *Cirq*. Zenodo. Aug. 2025. DOI: 10.5281/ZENODO.16867504.

- [8] Austin G. Fowler et al. “Surface codes: Towards practical large-scale quantum computation”. In: *Physical Review A* 86.3 (Sept. 2012), p. 032324. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.86.032324. URL: <http://arxiv.org/abs/1208.0928>.
- [9] Craig Gidney. “Stability Experiments: The Overlooked Dual of Memory Experiments”. In: *Quantum* 6 (Aug. 24, 2022), p. 786. ISSN: 2521-327X. DOI: 10.22331/q-2022-08-24-786. arXiv: 2204.13834 [quant-ph]. URL: <http://arxiv.org/abs/2204.13834>.
- [10] Craig Gidney. *Stim*. Version v1.15.0. May 2025. DOI: 10.5281/zenodo.15354872. URL: <https://doi.org/10.5281/zenodo.15354872>.
- [11] Craig Gidney. “Stim: a fast stabilizer circuit simulator”. In: *Quantum* 5 (July 2021), p. 497. ISSN: 2521-327X. DOI: 10.22331/q-2021-07-06-497. URL: <http://arxiv.org/abs/2103.02202>.
- [12] Craig Gidney and Austin G. Fowler. “Efficient Magic State Factories with a Catalyzed $|CCZ\rangle$ to $2|T\rangle$ Transformation”. In: *Quantum* 3 (Apr. 2019), p. 135. DOI: 10.22331/q-2019-04-30-135.
- [13] Oscar Higgott and Craig Gidney. “Sparse Blossom: correcting a million errors per core second with minimum-weight matching”. In: *Quantum* 9 (Jan. 2025), p. 1600. ISSN: 2521-327X. DOI: 10.22331/q-2025-01-20-1600. URL: <https://doi.org/10.22331/q-2025-01-20-1600>.
- [14] Ali Javadi-Abhari et al. *Quantum Computing with Qiskit*. June 2024. DOI: 10.48550/arXiv.2405.08810. arXiv: 2405.08810 [quant-ph].
- [15] Youngseok Kim et al. “Error Mitigation with Stabilized Noise in Superconducting Quantum Processors”. In: *Nature Communications* 16.1 (Sept. 2025), p. 8439. ISSN: 2041-1723. DOI: 10.1038/s41467-025-62820-9.

- [16] Catherine Leroux et al. *Snakes and Ladders: Adapting the surface code to defects*. Dec. 2024. DOI: 10.48550/arXiv.2412.11504. URL: <http://arxiv.org/abs/2412.11504>.
- [17] Sophia Fuhui Lin et al. “Codesign of quantum error-correcting codes and modular chiplets in the presence of defects”. In: *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. Apr. 2024, pp. 216–231. DOI: 10.1145/3620665.3640362. URL: <http://arxiv.org/abs/2305.00138>.
- [18] Daniel Litinski. “A game of surface codes: Large-scale quantum computing with lattice surgery”. In: *Quantum* 3 (2019), p. 128.
- [19] Daniel Litinski. “Magic state distillation: Not as costly as you think”. In: *Quantum* 3 (2019), p. 205.
- [20] Masoud Mohseni et al. *How to Build a Quantum Supercomputer: Scaling from Hundreds to Millions of Qubits*. Jan. 2025. DOI: 10.48550/arXiv.2411.10406. URL: <http://arxiv.org/abs/2411.10406>.
- [21] Prakash Murali et al. “Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers”. In: *Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems*. 2019, pp. 1015–1029.
- [22] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information 10th Anniversary Edition*. Cambridge University Press, 2010. ISBN: 9780511976667.
- [23] Jacob S. Palmer and Kaitlin N. Smith. *Boundaries of Acceptable Defectiveness: Redefining Surface Code Robustness under Heterogeneous Noise*. Nov. 2025. DOI: 10.48550/arXiv.2510.22001. arXiv: 2510.22001 [quant-ph]. (Visited on 12/13/2025).
- [24] Salonik Resch and Ulya R. Karpuzcu. “Benchmarking Quantum Computers and the Impact of Quantum Noise”. In: *ACM Comput. Surv.* 54.7 (July 2021), 142:1–142:35. ISSN: 0360-

0300. DOI: 10.1145/3464420. URL: <https://dl.acm.org/doi/10.1145/3464420>.
- [25] Joschka Roffe. “Quantum Error Correction: An Introductory Guide”. In: *Contemporary Physics* 60.3 (July 2019), pp. 226–245. ISSN: 0010-7514, 1366-5812. DOI: 10.1080/00107514.2019.1667078. arXiv: 1907.11157 [quant-ph].
- [26] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. DOI: <https://doi.org/10.1137/S0097539795293172>. URL: <https://arxiv.org/abs/quant-ph/9508027>.
- [27] Kaitlin N Smith et al. “Scaling superconducting quantum computers with chiplet architectures”. In: *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE. 2022, pp. 1092–1109.
- [28] Armands Strikis, Simon C Benjamin, and Benjamin J Brown. “Quantum computing is scalable on a planar array of qubits with fabrication defects”. In: *Physical Review Applied* 19.6 (2023), p. 064081.
- [29] Youngkyu Sung et al. “Non-Gaussian Noise Spectroscopy with a Superconducting Qubit Sensor”. In: *Nature Communications* 10.1 (Sept. 2019), p. 3715. ISSN: 2041-1723. DOI: 10.1038/s41467-019-11699-4.
- [30] Swamit S Tannu and Moinuddin K Qureshi. “Not all qubits are created equal: A case for variability-aware policies for NISQ-era quantum computers”. In: *Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems*. 2019, pp. 987–999.
- [31] Yu Tomita and Krysta M. Svore. “Low-Distance Surface Codes under Realistic Quantum Noise”. In: *Physical Review A* 90.6 (Dec. 2014), p. 062320. ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.90.062320. arXiv: 1404.3747 [quant-ph].

- [32] Keyi Yin et al. *Surf-Deformer: Mitigating Dynamic Defects on Surface Code via Adaptive Deformation*. Sept. 2024. DOI: 10.48550/arXiv.2405.06941. URL: <http://arxiv.org/abs/2405.06941>.

Vita

Jacob Palmer is a master's student in Computer Science at Northwestern University. He received a Bachelor of Science in Computer Science from Loyola University Chicago in 2022, where his research involved machine learning and open-source tooling. Since beginning his studies at Northwestern in 2024, his research has focused on quantum computing, particularly quantum error correction. After completing his master's degree in June 2026, he will continue at Northwestern to pursue a PhD in Computer Science under the supervision of Professor Kate Smith.