

Data Structures of Nature: Fermionic Encodings

by

Emiliia Dyrenkova

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science (Quantum Information)

Waterloo, Ontario, Canada, 2025

© Emiliia Dyrenkova 2025

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis is based on the preprint [35], where Emiliia Dyrenkova is the first author. Her contributions included the development of ideas, implementation of simulation code, data analysis, and writing.

In addition, the following sections were written specifically for this thesis by Emiliia Dyrenkova, with feedback from Richard Cleve and Michael Vasmer: Section 2.1, Section 2.2, Section 2.3, Section 2.4, Section 3.1, and Section 3.3. Other sections also include original writing not present in [35].

Abstract

A compelling application of quantum computers with thousands of qubits is quantum simulation. Simulating fermionic systems is both a problem with clear real-world applications and a computationally challenging task.

In order to simulate a system of fermions on a quantum computer, one has to first map the fermionic Hamiltonian to a qubit Hamiltonian. The most popular such mapping is the Jordan-Wigner encoding, which suffers from inefficiencies caused by the high weight of some encoded operators. As a result, alternative *local* encodings have been proposed that solve this problem at the expense of a constant factor increase in the number of qubits required. Some such encodings possess local stabilizers, i.e., Pauli operators that act as the logical identity on the encoded fermionic modes. A natural error mitigation approach in these cases is to measure the stabilizers and discard any run where a measurement returns a -1 outcome. Using a high-performance stabilizer simulator, we classically simulate the performance of a local encoding known as the Derby-Klassen encoding and compare its performance with the Jordan-Wigner encoding and the ternary tree encoding. Our simulations use more complex error models and significantly larger system sizes (up to 18×18) than in previous work. We find that the high sampling requirements of postselection methods with the Derby-Klassen encoding pose a limitation to its applicability in near-term devices and call for more encoding-specific circuit optimizations.

Acknowledgements

First and foremost, I would like to thank Michael Vasmer, whose patient mentorship shaped my growth and made much of this work possible. I am also deeply grateful to my late advisor, Raymond Laflamme. His belief in me opened doors I could not have imagined, and his support gave me the freedom to let curiosity guide my Master’s research.

I am grateful to my co-advisor, Richard Cleve, whose rigorous feedback challenged me to deepen my understanding and consistently meet a high standard of explanation throughout this thesis. I would also like to thank my thesis readers, Luke Schaeffer and Sepehr Assadi, for generously offering their time and helpful insights.

While working on the paper that this thesis is based on, I have had many helpful interactions with other researchers. I thank Toby Cubitt, Joel Klassen, Nathan Wiebe, the Whitfield Group (Andrew, Brent, Jason, Weishi, James) and many others for helpful discussions.

I am grateful to have met wonderful friends and colleagues at IQC. Thank you to Zach, Alex, Ningping, Shayan, Sascha, Rodolfo, Cristina, Matt Duschene, Matt Graydon, Katya, Arsalan, Anirban, Alev, Junan and many others for being supportive groupmates and friends. I’m also thankful for many friends from IQC and Perimeter Institute, whose curiosity and conversations across disciplines shaped how I think and communicate about research.

Thank you to my family for their unconditional love and support. To my mom, whose courage and faith in my future brought me to the United States eight years ago—and who continues to ensure I have everything I need to thrive. To my dad, who always listened with interest and patience to my broken explanations of my research in Russian. To my brother, whose humor has brightened many of my days. And to the rest of my family in Ukraine, whose bravery and resilience continue to inspire me every day. Last but not least, thank you to Darren—whose unwavering love, unshakable belief in my abilities, and fierce passion for life have given me the audacity to pursue the boldest dreams and the brightest version of the life I want to live.

I acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). This research was enabled in part by support provided by Compute Ontario (<https://computeontario.ca>) and the Digital Research Alliance of Canada (<https://alliancecan.ca>). I also thank Mike and Ophelia Lazaridis for funding.

Dedication

This thesis is dedicated to my late advisor, Raymond Laflamme, whose boundless generosity and radical curiosity made it possible for me to keep pursuing the questions that drive me.

Table of Contents

Author's Declaration	ii
Statement of Contributions	iii
Abstract	iv
Acknowledgements	v
Dedication	vi
List of Figures	x
List of Tables	xii
1 Introduction	1
2 Background	4
2.1 Qubits	4
2.2 Quantum Time Dynamics	7
2.3 Time Dynamics Native to Qubits	8
2.4 Fermions	11
2.5 Fermionic Encodings	15
2.5.1 Jordan-Wigner Encoding	16

2.5.2	Ternary Tree encodings	17
2.5.3	Derby-Klassen Encoding	18
3	Constructing Circuits with Fermionic Encodings	21
3.1	Stabilizer Codes Essentials	21
3.1.1	Stabilizer Codes	21
3.1.2	Error Syndrome and Stabilizer Measurement	23
3.1.3	Binary Symplectic Representation of Paulis	25
3.2	Preparing Fermionic Basis States	26
3.3	Trotterization Circuits	29
3.3.1	FSWAP Networks for Improved Jordan-Wigner Encoding	30
3.4	Observable Readout and Error Mitigation	31
3.4.1	Observables	31
3.4.2	Global Parity Postselection	33
3.4.3	Error Detection with Stabilizer Measurement	33
3.4.4	Stabilizer Reconstruction	34
4	Stabilizer Simulation for Testing Fermionic Encodings' Performance	36
4.1	Stabilizer Simulation	36
4.2	Error models	38
4.3	Simulated Circuits	40
4.4	Sampling	41
4.5	Results	42
5	Conclusion	46
	References	48
	Appendices	60

A	Depth tables	61
B	Additional Simulation Data	63

List of Figures

2.1	Example quantum circuit preparing a Bell state.	6
2.2	Example decomposition of the unitary $e^{-i\theta(X_1 \otimes X_2 \otimes Y_3)}$ into 1- and 2-qubit gates.	10
2.3	Summary of fermions in second quantization.	14
2.4	Ternary Tree Encoding	17
2.5	DK encoding lattice of size 4×4 with periodic boundaries.	18
3.1	Circuit for measuring a stabilizer generator of the DK encoding.	23
3.2	Simple demonstration of stabilizer measurement.	24
3.3	Measuring the eigenstates of two data qubits non-destructively with one ancilla.	25
3.4	Stabilizer measurement of an arbitrary Pauli P	25
3.5	FSWAP network on a 3×4 lattice for implementing hopping terms between all vertical neighbors.	31
3.6	Efficient DK measurement schedule.	33
3.7	Circuit for measuring a stabilizer generator of the DK encoding with flags.	34
4.1	The DK hopping operator and its Clifford version we use in our stabilizer simulation.	37
4.2	Comparison of the process fidelity of Clifford and non-Clifford logical rotations	38
4.3	Simulated Circuit Structure.	40
4.4	Performance of fermionic encodings in the standard depolarizing error model.	43
4.5	Performance of various postselection methods in the DK encoding (SI error model).	44

4.6	Error detection rates at different physical error rates (standard depolarizing error model).	45
B.1	Performance of fermionic encodings in the superconducting inspired error model.	65
B.2	Performance of various postselection methods in the DK encoding (SD error model).	66
B.3	Error detection rates at different physical error rates (superconducting inspired error model).	67
B.4	Comparison of hopping term measurement performance at different error rates.	68
B.5	Performance of the DK encoding in the aspirational error regime ($p = 0.01\%$) for the standard depolarizing error model.	69
B.6	Performance of the DK encoding in the near future error regime ($p = 0.1\%$) for the standard depolarizing error model.	70
B.7	Performance of the DK encoding in the aspirational error regime ($p = 0.01\%$) for the superconducting inspired error model.	71
B.8	Performance of the DK encoding in the near future error regime ($p = 0.1\%$) for the superconducting inspired error model.	72
B.9	Effect of global parity postselection (GP) in the JW encoding for the standard depolarizing error model.	73

List of Tables

4.1	Error models	39
A.1	DK Encoding, occupation number measurement, stabilizer reconstruction error detection, random logical circuits.	61
A.2	DK Encoding, hopping term measurement, stabilizer measurement error detection, random logical circuits.	62
A.3	DK Encoding, occupation number measurement, stabilizer reconstruction and full Trotter circuits.	62
A.4	JW Encoding, occupation number measurement, global parity postselection, full Trotter circuits.	62
A.5	TT Encoding, occupation number measurement, full Trotter circuits.	62

Chapter 1

Introduction

It is widely believed that chemical and materials simulations will be a high-impact application of future quantum computers [19, 6, 73, 77, 28]. A commonly studied computational task in this context is simulating the time evolution of many-electron systems. Mathematically, this involves solving the Schrödinger equation—a differential equation whose unknown is a time-dependent complex vector representing the quantum state of the system (e.g., the joint state of electrons in a molecule). The constraints and interactions of the system are described by a Hamiltonian—a complex Hermitian matrix that encodes the energy landscape of the system and dictates how its quantum state evolves over time.

There exist quantum algorithms that, in some sense, discretize this evolution to allow qubits to approximate the continuous dynamics of quantum systems. However, applying these algorithms to many-electron systems requires encoding electrons and their interactions into qubit states and operations. These encodings are not unique—and this freedom of representation creates opportunities to reduce the quantum resources required, much like how choosing an appropriate data structure can improve the performance of a classical algorithm.

The research studying these encodings adopts the language of fermions, which was originally developed in particle physics and is now widely used in condensed matter physics and quantum chemistry. A *fermion* is a type of fundamental particle that makes up matter—electrons fall under this category. It was within condensed matter physics that the first fermion-to-qubit encoding was developed: the Jordan-Wigner transform [63]. Ironically, its original purpose was to map qubit-like spin systems that were difficult to solve into fermionic systems that were analytically tractable.

The *Jordan-Wigner (JW) encoding* is an elegant solution using n qubits to represent n

fermionic modes (possible states any fermion in the system can take on). However, in the JW encoding, *local* fermionic operators (operators that act on only a few fermionic modes) are often represented by *non-local* qubit operators (qubit operators that, in the worst case, act on all qubits), which can require many gates to implement on hardware.

A variety of other fermionic encodings have been proposed in more recent times [14, 100, 90, 52, 91, 60, 31], some of which map local fermionic operators to local qubit operators at the cost of requiring more than one qubit per encoded fermion. Some of these encodings have the property that the encoded states lie in a subspace of the full Hilbert space, which allows us to detect errors by checking whether a state resides in the correct subspace. In certain encodings, such as *the Derby-Klassen (DK) or Compact encoding* [30], local Pauli operators stabilize the subspace similar to those of topological codes [65, 12], making it possible to detect errors by measuring these local operators. The DK encoding achieves these advantages by being specifically designed for the *Fermi-Hubbard model*—a widely studied fermionic system in which particles are arranged on a lattice and interact only with their nearest neighbors. Although this makes the DK encoding narrowly applicable—and less relevant to broader quantum chemistry and materials science applications—studying its performance serves as a testbed for the broader idea that model-specific encodings can offer meaningful advantages for targeted quantum simulations.

Small-scale demonstrations of quantum chemistry have been carried out on quantum computers with promising results [83, 54, 42, 93]. But achieving quantum advantage in these tasks likely requires a larger number of qubits and deeper circuits than are possible with today’s devices [26]. We could, therefore, assume that useful results will only be achieved with large fault-tolerant quantum computers capable of running arbitrarily long computations with arbitrarily low error rates. If such a machine was available, fermionic encodings would likely provide only minor improvements in runtime.

Unfortunately, we are still far from having such a machine. On the other hand, better and better pre-fault-tolerant devices are becoming available, with qubit counts approaching several hundreds and error rates low enough for non-trivial demonstrations of quantum error correction [3, 11]. It is possible that, with enough modest improvements—such as optimized fermionic encodings and encoding-enabled error mitigation—quantum simulation could yield useful results without the substantial overheads required for full fault-tolerant error correction.

How could we test the performance of such improvements? The most obvious answer would be to run experiments. The DK encoding was recently compared to the JW encoding on an ion-trap device in [82]. Although this work demonstrated several interesting encoding-based optimizations, current devices still fall short of reaching regimes where

classical simulation becomes intractable.

Using classical simulation of circuits may seem counterintuitive at first, since our ultimate goal is to use quantum computers for calculations inaccessible to classical computers. However, certain sub-theories of quantum mechanics—such as stabilizer quantum mechanics [45]—are classically simulable, while still exhibiting quantum phenomena such as entanglement and superposition. Indeed, stabilizer simulations are widely used to benchmark the performance of quantum error-correcting (QEC) codes under Pauli error models, and are believed to provide reliable insight into performance.

Here, we use an efficient stabilizer circuit simulator [40] to benchmark the performance of fermionic encodings at sizes beyond all previous simulations to the best of our knowledge. We use the circuit error model, where every circuit element (state preparation, gate, measurement) is followed by a stochastic Pauli error. This error model is commonly used in QEC research, as it captures a broad range of noise present in today’s devices. Furthermore, arbitrary Markovian noise (including coherent errors) can be transformed into stochastic Pauli noise using randomized compiling [101, 51], often with little or no experimental overhead [38]. This technique has also proven useful for error mitigation techniques such as probabilistic error cancellation [98, 36].

We leverage the error-detecting properties of the DK encoding to design robust procedures for state preparation and measurement in the Fermi-Hubbard Hamiltonian. We compare the performance of the DK encoding with the JW encoding and the *ternary tree (TT) encoding* [60] for circuits implementing first-order Trotterized time evolution of the 2D Fermi-Hubbard Hamiltonian. We also study the performance of multiple error mitigation methods for the DK encoding using random logical operator circuits inspired by [22].

We find that, although promising in terms of potential accuracy improvements, DK-enabled error mitigation suffers from high sampling costs, which limit its applicability to very low physical error rates or small lattice sizes. Our results confirm the need for more specialized circuit optimizations akin to those implemented in [82].

The remainder of this thesis is organized as follows. In Chapter 2, we provide the necessary background, including a review of basic facts about qubits, an introduction to quantum time evolution and its implementation on qubits, and a discussion about fermions and fermionic encodings. In Chapter 3, we present techniques for constructing efficient and error-resistant circuits by leveraging the structure of fermionic encodings. In Chapter 4, we report the results of our numerical study comparing the performance of the DK, JW, and TT encodings. Finally, in Chapter 5, we conclude with a discussion of possible extensions to this work, including implementations on quantum hardware.

Chapter 2

Background

The title of this thesis draws a parallel between fermionic encodings and classical data structures. In classical computing, the choice of data structure can significantly impact the efficiency of an algorithm by changing how information is organized and accessed. Similarly, in quantum computing, fermionic encodings determine how fermionic systems are mapped onto qubits, and different choices can lead to major differences in how various subtasks of simulation are implemented and in their resource requirements.

We begin this chapter by introducing the medium of quantum computation—qubits and their operations—in Section 2.1. In Section 2.2, we define the computational task of simulating quantum time evolution, and in Section 2.3, we explain how this task can be implemented on qubits using the *Trotterization* algorithm. Finally, in Section 2.4, we introduce fermions and the Fermi-Hubbard model, and in Section 2.5, we discuss how these systems can be encoded onto qubits through three fermionic encodings that we will later compare numerically.

2.1 Qubits

The fundamental building blocks of quantum computing are qubits. They can be thought of as quantum analogs of bits.

The state of one qubit is described by a unit vector in a two-dimensional complex *Hilbert space* (a vector space equipped with an inner product).

The canonical orthonormal basis for a single-qubit Hilbert space is:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (2.1)$$

The angle bracket notation is called Dirac bra-ket notation and has its origins in quantum physics¹.

An arbitrary pure state of a single qubit is a linear combination (called superposition) of these basis states:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \text{where } \alpha, \beta \in \mathbb{C}, \quad \text{and } |\alpha|^2 + |\beta|^2 = 1. \quad (2.2)$$

States of n qubits live in the 2^n -dimensional Hilbert space:

$$\mathcal{H}^{\otimes n} = \underbrace{\mathcal{H} \otimes \cdots \otimes \mathcal{H}}_{n \text{ times}}. \quad (2.3)$$

This means that single-qubit states can be combined into multi-qubit states using the tensor product. For example, given single-qubit states $|\psi\rangle$ and $|\phi\rangle$, their joint two-qubit state is given by:

$$|\psi\rangle \otimes |\phi\rangle. \quad (2.4)$$

The basis states of this space are $|x_1 x_2 \dots x_n\rangle$, where each $x_i \in \{0, 1\}$. Any pure n -qubit state is a normalized linear combination of these 2^n basis vectors.

Quantum operations on qubits can be represented as matrices acting on these vectors. For example, Pauli gates are commonly used:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (2.5)$$

Other common gates include the Hadamard (H), S and T gates:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}. \quad (2.6)$$

The two-qubit gates used in this work are the controlled-X (CX), controlled-Z (CZ), and controlled-Y (CY) gates, which are defined as follows:

¹The notation $\langle\psi|$ is called a *bra*, representing a row vector, and $|\phi\rangle$ is called a *ket*, representing a column vector. The inner product is written as $\langle\psi|\phi\rangle$.

$$\text{CX} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \text{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad \text{CY} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{bmatrix}. \quad (2.7)$$

In this work, quantum circuit notation will be used. Fig. 2.1 demonstrates a simple circuit. Circuits are read left to right.

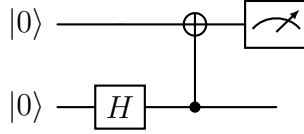


Figure 2.1: Example quantum circuit preparing a Bell state. $|0\rangle$ on the left represents the initial state, then a Hadamard gate is applied on the second qubit, then a CNOT is applied with the second qubit as control and the first qubit as target. The final box on the first qubit is the measurement.

This circuit demonstrates preparation of a special state called a *Bell state*:

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) \quad (2.8)$$

This is an entangled state, i.e., it cannot be represented as a tensor product of any two valid single-qubit states. When the first qubit is measured, we can fully determine the state of the second qubit without measuring it. If the first qubit is measured as $|0\rangle$ then we know that the second qubit is in the $|0\rangle$ state as well, same goes for $|1\rangle$.

Finally, a special subset of quantum gates known as *Clifford gates* allows for efficient classical simulation of quantum circuits. This is due to the *Gottesman-Knill theorem*[45], which states that any quantum circuit composed exclusively of Clifford gates can be efficiently simulated in polynomial time using classical computers. The key property of Clifford gates is that they map Pauli operators to Pauli operators under conjugation. The set of all such operations forms the *Clifford group* $\mathcal{C}_n$ on n qubits, which is defined as:

$$\mathcal{C}_n = \{U \in \mathcal{U}(2^n) \mid UPU^\dagger \in \mathcal{P}_n \text{ for all } P \in \mathcal{P}_n\}, \quad (2.9)$$

where $\mathcal{P}_n$ is the n -qubit *Pauli group* including phases,

$$\mathcal{P}_n = \{\alpha P_1 \otimes \cdots \otimes P_n \mid P_j \in \{I, X, Y, Z\}, \alpha \in \{\pm 1, \pm i\}\}, \quad (2.10)$$

and $\mathcal{U}(2^n)$ is the unitary group on n qubits.

The Clifford group is generated by the following three operations:

$$\mathcal{C}_n = \langle H, S, \text{CNOT} \rangle \quad (2.11)$$

2.2 Quantum Time Dynamics

Quantum dynamics, i.e., the time evolution of quantum systems, is considered to be particularly well-suited for quantum computers. Time evolution of generic quantum systems is governed by the Schrödinger equation:

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H |\psi(t)\rangle, \quad (2.12)$$

where $|\psi(t)\rangle$ is a *normalized complex vector* representing the state of a quantum system within a Hilbert space, and H is a *Hermitian matrix* acting on this Hilbert space. In this text, we assume that *both $|\psi(t)\rangle$ and H are finite dimensional*. The Hamiltonian H encodes both the energy and the interactions present in the system. When the Hamiltonian is time-independent, the solution describing the evolution of the quantum state takes the form:

$$|\psi(t)\rangle = e^{-iHt/\hbar} |\psi(0)\rangle. \quad (2.13)$$

Therefore, the central task in simulating quantum dynamics is to implement the operator $e^{-iHt/\hbar}$. This task, commonly referred to as *Hamiltonian simulation*, has also become a fundamental subroutine in numerous quantum algorithms, both within and beyond the scope of quantum simulation.

In quantum simulation, the goal is often not to reconstruct the full state vector, but rather to extract specific physical quantities. This can typically be described by computing the expectation value of an observable $\mathcal{O}$:

$$\langle \mathcal{O} \rangle = \langle \psi(t) | \mathcal{O} | \psi(t) \rangle, \quad (2.14)$$

where $\mathcal{O}$ is a *Hermitian matrix* corresponding to a measurable physical property.

In practice, this involves preparing qubits in the state $|\psi(t)\rangle$, performing measurements corresponding to the observable $\mathcal{O}$, and averaging the measurement results obtained from repeating this process multiple times.

To summarize, what are the computational subroutines of quantum time dynamics simulation? We can break it down into the following three subproblems:

- Preparation of $|\psi(0)\rangle$
- Implementation of $e^{-iHt/\hbar}$
- Measurement of observable $\mathcal{O}$

In this chapter, we will primarily focus on the second subproblem of implementing the time evolution, however, we will discuss preparation of certain states and measurement of certain observables in the next chapter.

2.3 Time Dynamics Native to Qubits

To understand how to implement time evolution on a quantum computer, it is useful to consider Hamiltonians that are, in a sense, native to qubits. These are known as *spin Hamiltonians*.

Interestingly, long before the idea of a quantum computer was formalized, condensed matter physicists were already working with qubits—except they called them *spins*. *Spin models* have long been used to describe magnetic materials in quantum mechanics.

A widely studied example in the context of quantum simulation is the transverse-field Ising model (TFIM), which is both physically relevant² and computationally tractable. It describes spins on a lattice with nearest-neighbor interactions in the Z direction and a transverse magnetic field in the X direction:

$$H = -J \sum_{i=1}^{N-1} Z_i Z_{i+1} - h \sum_{i=1}^N X_i, \quad (2.15)$$

where Z_i and X_i are Pauli operators acting on qubit i , J is the interaction strength, and h is the transverse field strength.

²It was initially developed as a mathematical model for ferromagnets.

This Hamiltonian is expressed as a sum of *local* terms—Pauli operators acting on only one or two qubits. More generally, we consider Hamiltonians of a similar form:

$$H = \sum_k H_k, \quad (2.16)$$

where each term H_k acts on a subset of qubits. In discussions of quantum simulation algorithms, it is commonly assumed that each H_k acts on a constant number of qubits, although this assumption is not strictly required for these algorithms to work.

A natural strategy for implementing the evolution operator $e^{-i\tau H}$ is to break it down into exponentials of local terms $e^{-i\tau H_k}$. However, since the H_k generally do not commute, this factorization is not exact:

$$e^{-i\tau \sum_k H_k} \neq \prod_k e^{-i\tau H_k}. \quad (2.17)$$

To address this, we use the *Trotter decomposition*, also known as the product formula. The first-order Trotter approximation expresses the full evolution as a repeated product of exponentials of individual terms:

$$e^{-iH\tau} \approx \left(\prod_k e^{-i\frac{\tau}{T} H_k} \right)^T, \quad (2.18)$$

where T is the number of *Trotter steps*. The approximation becomes exact in the limit $T \rightarrow \infty$, and higher-order formulas can improve accuracy at the cost of additional gate depth.

For the first-order Trotter formula, the approximation error in operator norm is [70]:

$$\epsilon = \left\| e^{-i\tau H} - \left(\prod_k e^{-i\frac{\tau}{T} H_k} \right)^T \right\| = \mathcal{O}\left(\frac{\tau^2}{T}\right), \quad (2.19)$$

which implies that to achieve precision ϵ , it suffices to take

$$T = \mathcal{O}(\tau^2 \epsilon^{-1}). \quad (2.20)$$

This scaling can be made more precise in specific cases. For example, if the Hamiltonian has only two non-commuting terms $H = H_1 + H_2$, it is shown in [70] that:

$$\left\| e^{-i\frac{\tau}{T} H} - e^{-i\frac{\tau}{T} H_1} e^{-i\frac{\tau}{T} H_2} \right\| \leq \frac{(\tau/T)^2}{2} \| [H_1, H_2] \|. \quad (2.21)$$

This shows that the number of Trotter steps needed depends not only on τ and ϵ , but also on the commutation relations among the Hamiltonian terms³. If all terms commute, then the product formula is exact and a single step is enough.

Employing Trotter decomposition as an algorithmic tool is generally referred to as **Trotterization**. Trotterization gives us a systematic way to approximate time evolution by decomposing it into a sequence of local operations (as local as the terms of the Hamiltonian, of course). These local operations can be further compiled into circuits optimized for specific hardware architectures. This is the approach commonly taken in near-term experimental demonstrations of digital quantum simulation, e.g. in [64].

Here, we will describe a standard hardware-agnostic decomposition of local operations $e^{-i\theta P_i}$, where P_i is a tensor product of Pauli matrices—for example, $P_i = X_1 \otimes X_2 \otimes Y_3$ —and θ is a real parameter. Such unitaries can be compiled into short sequences of 1- and 2-qubit gates. The decomposition typically involves a single *arbitrary-angle gate*—a gate whose angle depends on the parameter θ —together with a sequence of Clifford gates. An example is shown in Fig. 2.2.

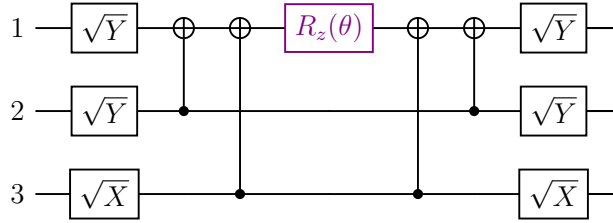


Figure 2.2: Example decomposition of the unitary $e^{-i\theta(X_1 \otimes X_2 \otimes Y_3)}$ into 1- and 2-qubit gates. This decomposition was automatically generated with CIRQ [32].

The $R_z(\theta)$ gate is the only arbitrary-angle component; all other gates in the circuit are Clifford operations. The weight of the Pauli operator, i.e., the number of qubits it acts on non-trivially, determines the number of Clifford gates required. *Thus, minimizing the weight of Pauli terms in the Hamiltonian can reduce circuit depth.*

This is especially important for near-term hardware, where gate fidelities and/or qubit connectivity are limited. As a rule of thumb, two-qubit gates are more noisy and more difficult to implement than single-qubit gates in various different architectures.

³The *commutator* of two operators A and B is defined as $[A, B] = AB - BA$, while the *anticommutator* is $\{A, B\} = AB + BA$.

2.4 Fermions

In the previous section, we outlined how qubits could be used to study quantum dynamics governed by Hamiltonians that are “native” to qubit systems. However, these spin Hamiltonians have limited applicability to many physical systems of real-world interest. For this reason, we now turn our attention to fermionic systems.

Fermions are ubiquitous in nature. They include all fundamental matter particles — electrons, protons, and neutrons are all fermions. Interestingly, even certain atoms (e.g., Lithium-6 [85]) behave as fermions. This means that these particles and atoms obey the same fundamental set of rules and can be modeled by Hamiltonians and states with similar mathematical structures. Although we can abstractly discuss simulating generic fermionic particles representing any of the aforementioned matter particles, it is useful to remember that fermionic models are most commonly intended to represent electrons.

Whereas qubits are quantum analogs of bits, fermions are better thought of as quantum analogs of balls and bins. The “balls” are indistinguishable fermionic particles, and the “bins” are the single-particle quantum states—referred to as *modes*—that these particles can occupy. For electrons in atoms, for example, the modes correspond to spin-orbitals: the atomic energy levels and sublevels (such as the 1s, 2p orbitals one might recall from high school chemistry). Electrons occupy these states in an atom, and the outermost partially filled states determine how the atom interacts and forms bonds with others. Consequently, simulating electron interactions during a chemical reaction could provide valuable insights into the potential outcomes of such reactions.

Because fermions are *indistinguishable*, we cannot track individual particles, only the occupancy of each mode—that is, how many “balls” are in each “bin”. Let us attempt to represent fermions with a Hilbert space and contrast them to qubits. Let the binary values (0 or 1) encode whether the mode is unoccupied or occupied with a fermion. For example, the state $|010\rangle$ corresponds to a configuration where only the second mode is occupied by a fermion. We are limited to binary values due to the *Pauli exclusion principle*: from theory, experiments, and observations we know that at most one fermion will occupy each mode.

Now, let us consider a simple fermionic model: the **spinless Fermi-Hubbard model**. This model describes fermions that can hop (or tunnel) between spatially adjacent modes and experience mutual repulsion when occupying neighboring modes. Thus, it incorporates two fundamental interactions: *hopping*, which allows fermions to move between adjacent modes, and the *Coulomb interaction*, which imposes an energy penalty whenever two fermions simultaneously occupy adjacent modes—discouraging (though not forbidding) such configurations. We note that this Hamiltonian does not change the number of

fermions in the system which means it represents a closed system.

To illustrate these interactions, we start with a system of two fermionic modes. The Hilbert space in this case is four-dimensional, with computational basis states

$$|00\rangle, \quad |01\rangle, \quad |10\rangle, \quad |11\rangle, \quad (2.22)$$

where, for example, $|10\rangle$ corresponds to one fermion in mode 1 and none in mode 2.

The hopping interaction between modes 1 and 2, which we denote by a Hamiltonian H_{12} , swaps a fermion between these two modes. Its action on basis states is:

$$\begin{aligned} H_{12} |00\rangle &= 0, \\ H_{12} |01\rangle &= |10\rangle, \\ H_{12} |10\rangle &= |01\rangle, \\ H_{12} |11\rangle &= 0. \end{aligned} \quad (2.23)$$

In matrix form, written in the ordered basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, this operator is:

$$H_{12} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (2.24)$$

The Coulomb interaction between modes 1 and 2, which we denote by a Hamiltonian C_{12} , contributes energy only when both modes are occupied. In some sense, it “selects” states with two fermions in neighboring modes. Its action is:

$$\begin{aligned} C_{12} |00\rangle &= 0, \\ C_{12} |01\rangle &= 0, \\ C_{12} |10\rangle &= 0, \\ C_{12} |11\rangle &= |11\rangle, \end{aligned} \quad (2.25)$$

with the corresponding matrix:

$$C_{12} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.26)$$

A spinless Fermi-Hubbard Hamiltonian on an arbitrary graph can be written as:

$$H = -t \sum_{\langle u,v \rangle} H_{uv} + U \sum_{\langle u,v \rangle} C_{uv}, \quad (2.27)$$

where t is the hopping amplitude and U is the interaction strength. As before, H_{uv} exchanges a fermion between modes u and v , and C_{uv} assigns energy to configurations in which both modes are occupied. The angled brackets $\langle u, v \rangle$ indicate that u and v are neighbors on the graph. Hence, H_{uv} and C_{uv} are local: like local qubit operators, they act non-trivially only on the involved modes. That is,

$$H_{uv} = I \otimes \cdots \otimes H \otimes \cdots \otimes I. \quad (2.28)$$

In order to demonstrate the effects of these interactions on the total energy of the system, let us consider a simple example: a system with three fermionic modes and two fermions. The allowed basis states are $|110\rangle$, $|101\rangle$, and $|011\rangle$. Suppose we are interested in a 1D Fermi-Hubbard Hamiltonian on this three-mode system:

$$H = C_{12} + C_{23} - H_{12} - H_{23}, \quad (2.29)$$

We now compute the expectation values of each term in H on each of the basis states:

$$\begin{aligned} \langle 110 | C_{12} | 110 \rangle &= 1, & \langle 110 | C_{23} | 110 \rangle &= 0, \\ \langle 110 | H_{12} | 110 \rangle &= 0, & \langle 110 | H_{23} | 110 \rangle &= 0, \\ \\ \langle 101 | C_{12} | 101 \rangle &= 0, & \langle 101 | C_{23} | 101 \rangle &= 0, \\ \langle 101 | H_{12} | 101 \rangle &= 0, & \langle 101 | H_{23} | 101 \rangle &= 0, \\ \\ \langle 011 | C_{12} | 011 \rangle &= 0, & \langle 011 | C_{23} | 011 \rangle &= 1, \\ \langle 011 | H_{12} | 011 \rangle &= 0, & \langle 011 | H_{23} | 011 \rangle &= 0. \end{aligned} \quad (2.30)$$

We observe that the Coulomb terms contribute +1 energy to $|110\rangle$ and $|011\rangle$, while all hopping terms give zero expectation value since they map each basis state to an orthogonal one. Therefore, the basis state with the lowest energy is $|101\rangle$, which *avoids both Coulomb “penalties”*.

However, if one were to search for the Fermi-Hubbard Hamiltonian online, they would not find it expressed in terms of operators like H_{uv} and C_{uv} . Just as qubit (spin) Hamiltonians are built from Pauli matrices, fermionic Hamiltonians are built from a different set of operator primitives. In this work, we adopt the *second quantization* formalism, where the fundamental building blocks are the *creation* and *annihilation* operators, denoted $a^\dagger$ and a .

The simplest fermionic state is the *vacuum state*, in which no mode is occupied:

$$|vac\rangle = \underbrace{|00\dots 0\rangle}_{K \text{ modes}}. \quad (2.31)$$

In this work we will assume that there is some finite number of modes K .

We define the creation operator $a_i^\dagger$, which adds a fermion to mode i (for $i \leq K$). Acting on the vacuum state, it produces a state with a single fermion in mode i :

$$a_i^\dagger |vac\rangle = |0\dots 1\dots 0\rangle. \quad (2.32)$$

The annihilation operator a_i is the Hermitian conjugate of $a_i^\dagger$, and removes a fermion from mode i :

$$a_i |\dots 1\dots\rangle = |\dots 0\dots\rangle. \quad (2.33)$$

Due to the Pauli exclusion principle, fermionic creation and annihilation operators satisfy:

$$a_i^\dagger a_i^\dagger = 0, \quad a_i a_i = 0. \quad (2.34)$$

Beyond the Pauli exclusion principle, fermionic systems are believed to possess another fundamental property – *antisymmetry*⁴:

$$a_i^\dagger a_j^\dagger |vac\rangle = -a_j^\dagger a_i^\dagger |vac\rangle. \quad (2.35)$$

⁴The antisymmetry of fermionic states is widely accepted across various areas of physics and is often treated as synonymous with the Pauli exclusion principle. Strictly speaking, however, antisymmetry imposes additional constraints on the fermionic state space beyond those captured by the exclusion principle alone—constraints which are yet to be verified experimentally [67, 47].

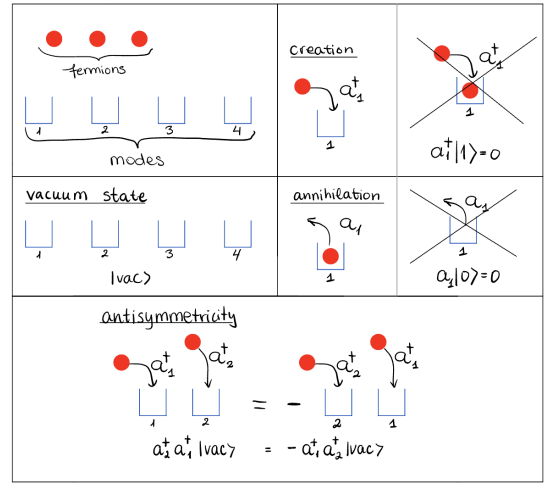


Figure 2.3: Summary of fermions in second quantization.

The second quantization operators can be summarized using their anticommutation relations:

$$\{a_i, a_j^\dagger\} = a_i a_j^\dagger + a_j^\dagger a_i = \delta_{ij} \mathbb{I}, \quad \{a_i, a_j\} = \{a_i^\dagger, a_j^\dagger\} = 0 \quad (2.36)$$

Using these operators, the local hopping and interaction terms can now be written in their standard second-quantized form. The hopping term between modes u and v is:

$$H_{uv} = a_u^\dagger a_v + a_v^\dagger a_u, \quad (2.37)$$

and the interaction term is:

$$C_{uv} = a_u^\dagger a_u a_v^\dagger a_v. \quad (2.38)$$

Now, the Fermi-Hubbard Hamiltonian from Eq. (2.27) can be written as:

$$H_{FH} = -t \sum_{\langle u,v \rangle} (a_u^\dagger a_v + a_v^\dagger a_u) + U \sum_{\langle u,v \rangle} a_u^\dagger a_u a_v^\dagger a_v, \quad (2.39)$$

How could we use the techniques described in Section 2.3 to implement time evolution under this Hamiltonian? Turns out we can map this Hamiltonian (or any fermionic Hamiltonian) to a qubit Hamiltonian, and this mapping is not unique. The freedom to choose a mapping allows for optimizing various implementation details, and, as we will see in Chapter 3, even creates an opportunity for mapping-specific error mitigation protocols. These mappings are commonly referred to as *fermionic encodings* and they are the topic of the next section.

2.5 Fermionic Encodings

In this section, we will discuss three fermionic encodings that will be compared numerically in Chapter 4. Jordan-Wigner (JW) encoding is the first, most simple and, therefore, most popular encoding. Ternary Tree (TT) encoding is the one-to-one (one fermionic mode per qubit) encoding that maps fermionic operators to qubit operators of *optimal* Pauli weight. Finally, Derby-Klassen (DK) encoding, also referred to as Compact encoding, is the encoding tailor-made for the 2D Fermi-Hubbard Hamiltonian. This encoding has smaller Pauli weight of the encoded Fermi-Hubbard operators than the optimal Ternary Tree encoding at the cost of constant factor of additional qubits. Additionally, its special structure allows for additional error mitigation opportunities unavailable for other encodings.

2.5.1 Jordan-Wigner Encoding

Let us begin by directly representing second-quantized fermionic basis states as computational basis states of qubits. The simplest approach is to represent each fermionic mode occupation (0 or 1) with a single qubit, thus directly encoding the occupation number basis into qubit basis states.

In this case, what would be the Pauli operators representing the creation and annihilation operators? Consider the following operators: $\frac{1}{2}(X_j + iY_j)$ and $\frac{1}{2}(X_j - iY_j)$. If we only consider single qubits, they behave exactly as $a_j, a_j^\dagger$:

$$\frac{1}{2}(X_j + iY_j)|0\rangle_j = 0; \quad \frac{1}{2}(X_j + iY_j)|1\rangle_j = |0\rangle_j; \quad \frac{1}{2}(X_j - iY_j)|0\rangle_j = |1\rangle_j; \quad \frac{1}{2}(X_j - iY_j)|1\rangle_j = 0; \quad (2.40)$$

However, if we consider the action of these operators on multi-qubit states, we realize that this is not sufficient. The creation and annihilation operators anticommute even when acting on different modes, while qubit operations with non-overlapping support always commute.

Therefore, the Jordan-Wigner encoding [63] maps the creation and annihilation operators to the following Pauli matrices:

$$a_j^\dagger = \left(\prod_{k=1}^{j-1} Z_k \right) \frac{X_j - iY_j}{2} \quad (2.41)$$

$$a_j = \left(\prod_{k=1}^{j-1} Z_k \right) \frac{X_j + iY_j}{2} \quad (2.42)$$

where X_j, Y_j , and Z_j are the Pauli operators acting on the j -th qubit.

If Jordan-Wigner encoding is applied to the 2D spinless Fermi-Hubbard, the operators are mapped as following (without loss of generality, assume that $i < j$):

$$a_i^\dagger a_j + a_j^\dagger a_i \rightarrow \frac{1}{2} \left(\prod_{k=i+1}^{j-1} Z_k \right) (X_i X_j + Y_i Y_j), \quad (2.43)$$

$$a_i^\dagger a_i a_j^\dagger a_j \rightarrow \frac{1}{4} (I + Z_i Z_j - Z_i - Z_j). \quad (2.44)$$

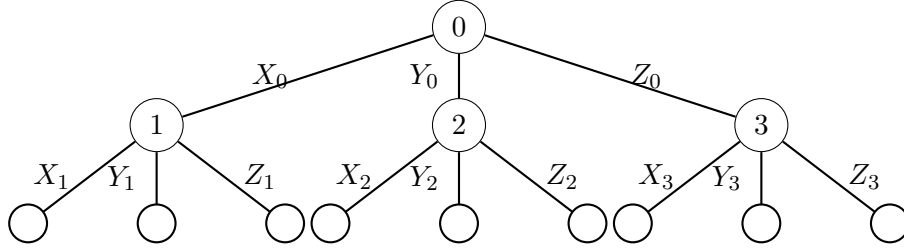


Figure 2.4: Ternary Tree Encoding operators are defined by assigning qubit indices to tree nodes that are not leaves and Pauli operators to the edges. This example tree creates 9 anticommuting Pauli operators on 4 qubits, with any subset of 8 of them giving a set of 8 encoded Majorana operators for 4 fermionic modes.

2.5.2 Ternary Tree encodings

If we focus on implementing operators following correct anticommutation relations without worrying about how fermionic states are represented, we can achieve the optimal weight of operators when using K qubits.

It is helpful to use the Majorana operators, which for a given fermionic mode j are defined as:

$$\gamma_{2j} = a_j + a_j^\dagger, \quad \gamma_{2j+1} = -i(a_j - a_j^\dagger). \quad (2.45)$$

Majorana operators are an equivalent representation of fermionic systems that satisfy simpler anticommutation relations:

$$\{\gamma_u, \gamma_v\} = 2\delta_{uv}, \quad \text{for } u, v \in [2n], \quad (2.46)$$

where we use the shorthand $[2n]$ to denote $\{1, 2, \dots, 2n\}$.

Any set of $2n$ Pauli operators that satisfy Eq. (2.46) defines a valid encoding. Ternary trees have been used to find optimal encodings in terms of the weight of encoded Majorana operators. To demonstrate this, consider a small example in Fig. 2.4. Each numbered node corresponds to a qubit with $n_{\text{modes}} = n_{\text{qubits}} = 4$. Now, all paths from root to leaf correspond to mutually anticommuting Pauli strings by construction. Since we need 8 encoded Majorana operators for $n_{\text{modes}} = 4$, we can select any 8 Pauli operators from the tree. In [60] it was proved that the average weight w of a one-to-one encoding must satisfy $w \geq \log_3 2n$, with ternary tree encodings saturating the bound.

The encoding from [60] can create entangled qubit states implementing simple fermionic basis states. However, in the numerical portion of our work, we will use pruned Sierpinski

tree encoding from [50], which is a modified ternary tree encoding where the fermionic basis states are encoded with the qubit computational basis states.

There is no simple way to summarize the exact definition of the encoded Fermi-Hubbard operators as they depend on the lattice size non-trivially. In our numerical results we used code associated with [50] to generate the encoded operators programmatically.

2.5.3 Derby-Klassen Encoding

The Derby-Klassen (DK) or Compact Encoding [31] is an encoding optimized for the 2D square lattice spinless Fermi-Hubbard model.

Let $\mathcal{L}$ be a square lattice of length $L \times L$ with periodic boundary conditions. We use $\mathcal{L}_v$, $\mathcal{L}_e$, and $\mathcal{L}_f$ to refer to the vertices, edges, and faces of $\mathcal{L}$, respectively. Recall that the faces of the square lattice are two-colourable, i.e., each face can be assigned a colour such that faces sharing an edge have different colours. We call these two sets of faces red and blue, and we denote them by $\mathcal{L}_f^r$ and $\mathcal{L}_f^b$, respectively. In the DK encoding, we place a qubit on each vertex and each blue face of $\mathcal{L}$. Each edge is also assigned an orientation as shown in Fig. 2.5.

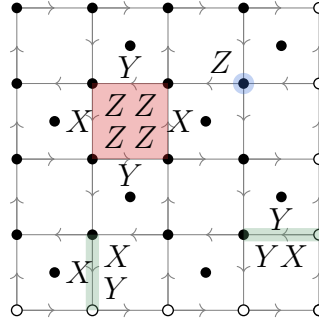


Figure 2.5: DK encoding lattice of size 4×4 with periodic boundaries. Each hollow vertex is identified with the corresponding filled vertex on the opposite boundary. We highlight a stabilizer in red, a vertex operator in blue, and two edge operators in green.

The DK encoding has an encoded fermionic mode for every vertex of $\mathcal{L}_v$ and is customarily defined in terms of a convenient Majorana operator basis:

$$\begin{aligned} V_j &= -i\gamma_{2j}\gamma_{2j+1} \quad j \in \mathcal{L}_v, \\ E_{ij} &= -i\gamma_{2i}\gamma_{2j} \quad (i, j) \in \mathcal{L}_e, \end{aligned} \tag{2.47}$$

where γ_{2j} and γ_{2j+1} are Majorana operators at vertex j , and we refer to V_j and E_{ij} as vertex and edge operators, respectively. The DK encoding maps these operators to the following Pauli operators:

$$\tilde{V}_j = Z_j, \quad (2.48)$$

$$\tilde{E}_{ij} = \begin{cases} X_i Y_j X_{f(i,j)} & (i,j) \text{ oriented downwards,} \\ -X_i Y_j X_{f(i,j)} & (i,j) \text{ oriented upwards,} \\ X_i Y_j Y_{f(i,j)} & (i,j) \text{ horizontal,} \end{cases} \quad (2.49)$$

$$\tilde{E}_{ji} = -\tilde{E}_{ij}$$

where indices i, j correspond to vertex qubits, and $P_{f(i,j)}$ denotes a Pauli operator P acting on the qubit located at the face containing edge (i, j) . We will use the V_j, E_{ij} for the fermionic versions of these operators and $\tilde{V}_j, \tilde{E}_{ij}$ for the encoded qubit operators. In this work, we only consider encodings defined on lattices with even L , as this ensures that there are no undetectable Majorana errors [30].

This encoding maps the hopping terms of the Fermi-Hubbard Hamiltonian as follows:

$$a_i^\dagger a_j + a_j^\dagger a_i \rightarrow \begin{cases} \frac{1}{2} \left(X_i X_j Y_{f(i,j)} + Y_i Y_j Y_{f(i,j)} \right), & (i,j) \text{ horizontal,} \\ \frac{1}{2} (-1)^{g(i,j)} \left(X_i X_j X_{f(i,j)} + Y_i Y_j X_{f(i,j)} \right), & (i,j) \text{ vertical,} \end{cases} \quad (2.50)$$

where $g(i, j)$ allows for a factor of (-1) when appropriate due to the definition of vertical edge operators in Eq. (2.49). The encoding for the Coulomb terms is the same as Eq. (2.44).

The compact encoded operators are not the only benefit provided by the DK encoding. The extra qubits introduced in this encoding create redundancy, causing all valid encoded states to reside in a subspace of the full qubit Hilbert space. This subspace is stabilized by a set of operators known as *stabilizers*, which correspond to closed loops of Majorana operators [31, 61]. Consequently, encoded states under the DK encoding have the structure of a *stabilizer code*, the most widely studied type of *quantum error correcting (QEC) code*.

However, because the encoded operators (known as *logical operators* in QEC) are low-weight, the code can only detect errors of even lower weight. For instance, because the vertex operator is encoded as a single-qubit Z , single-qubit Z errors are undetectable. As

a result, this stabilizer code has a code distance of one, making it unsuitable for fault tolerance⁵. Thus, the desirable feature of low-weight encoded operators is directly at odds with robust error-correction performance. Nevertheless, certain errors can still be detected. It is therefore natural to ask whether this stabilizer structure can be leveraged for error mitigation—either for near-term applications or in conjunction with “proper” quantum error-correcting codes.

In the next chapter, we will explore how this stabilizer structure can be utilized for error mitigation in the context of quantum simulation. We will borrow concepts from quantum error correction and fault tolerance, adapting them to accommodate the constraints of the encoding and the limitations of near-term hardware.

⁵Fault tolerance, in the most general sense, refers to the capability of performing computations of arbitrary length through repeated rounds of error correction.

Chapter 3

Constructing Circuits with Fermionic Encodings

Now that we have established the computational problem (time evolution), the algorithm (Trotterization), and our choice between three “data structures” (the JW, TT, and DK encodings), we can discuss how this choice affects the practical implementation of Trotterization. As suggested at the end of the previous chapter, viewing the DK encoding as a stabilizer code provides useful insight. Therefore, we begin this chapter by reviewing some key concepts and techniques from quantum error correction. We will then proceed to discuss circuit constructions for state preparation, time evolution, and measurement.

3.1 Stabilizer Codes Essentials

In this section, we will outline properties and definitions of the stabilizer error correcting codes relevant to this work, drawn from Chapter 3 in [43]. We will illustrate some of the properties with the structures from the DK encoding.

3.1.1 Stabilizer Codes

A *stabilizer code* is a subclass of quantum error correction codes in which the set of encoded states (*codespace*) is specified by a set of *stabilizers* which form an abelian group¹ S called

¹A *group* is defined as a set of elements equipped with a group operation. The group operation for the stabilizer group is multiplication. When a group operation is applied to two elements, the result is also a

the *stabilizer group*. This means that the codespace (the set of encoded operators) is the mutual +1 eigenspace of the stabilizers:

$$\mathcal{C}_f = \{|\psi\rangle : S|\psi\rangle = |\psi\rangle, \forall S \in \mathbf{S}\} \quad (3.1)$$

Note that $-I \notin \mathbf{S}$ since $-I$ does not have any +1 eigenstates.

A group will have a set of *generators*: a minimal set of elements from which any element in the group can be generated with the group operation. Let us denote this set of stabilizer generators as $\{S_k\}$ where k indexes the stabilizer generators.

As mentioned before, in the DK encoding, the set of stabilizers corresponds to closed loops of Majorana operators (encoded operators corresponding to a sequence of Majorana operators on a closed loop multiplied together). Remember that we defined the two-colourable lattice $\mathcal{L}$, where faces with qubits colored as blue and faces without qubits colored as red. We consider a generating set of the stabilizers comprising operators associated with the red faces of $\mathcal{L}$. Let $k \in [L^2/2]$ index the red faces. For each k we have a stabilizer generator S_k formed by taking the product of all edge operators surrounding face k . Each of these operators is weight eight and acts on the vertex qubits surrounding the face as Z , on horizontally neighbouring face qubits as X , and on vertically neighbouring face qubits as Y .

The explicit form of such a stabilizer generator with the vertex qubits surrounding face k indexed as i, j, l, m in clockwise direction starting from upper left corner:

$$S_k = Z_i Z_j Z_l Z_m Y_{f(i,j)} Y_{f(l,m)} X_{f(j,l)} X_{f(m,i)} \quad (3.2)$$

See Fig. 2.5 for an example.

It is not generally required in quantum error correction codes that the stabilizer group and the encoded operations are Pauli operators. However, it is convenient that is the case for the DK encoding, because Pauli operators will only have two eigenvalues (+1 and -1) and they will either commute or anticommute with each other.

From these definitions, it is also clear that all *encoded operators* will commute with all stabilizers, since the encoded operators are designed to map encoded states to encoded states²:

member of the group. In *abelian groups*, elements of the group commute with each other.

²In quantum error correction literature, these are called logical operations and logical states, but we will use the term encoded operators and encoded states as the primary goal of our stabilizer code is to encode a fermionic system onto qubits.

$$[E_{ij}, S_k] = 0, [V_j, S_k] = 0, \forall i, j, k \quad (3.3)$$

On the other hand, operators that anticommute with the stabilizers are the *detectable errors*. For example, it is easy to see that a single X_i or Y_i on a vertex qubit i will anticommute with a stabilizer generators that include this vertex qubit.

3.1.2 Error Syndrome and Stabilizer Measurement

Let us now define the concept of *error syndrome*. Let $|\psi\rangle$ be an eigenstate of all the stabilizers, not necessarily with the eigenvalue $+1$. Let g be the total number of stabilizer generators. Then the error syndrome $\sigma \in \{+1, -1\}^g$ of the state $|\psi\rangle$ has $+1$ at position k if $|\psi\rangle$ is the $+1$ eigenstate of S_k and -1 otherwise. Let us denote the value corresponding to a stabilizer generator S_k as $\sigma(k)$.

Conveniently, we can “read out” the error syndrome without measuring and, therefore, destroying the state $|\psi\rangle$. For this, we will employ the standard Hadamard test circuit in Fig. 3.1.

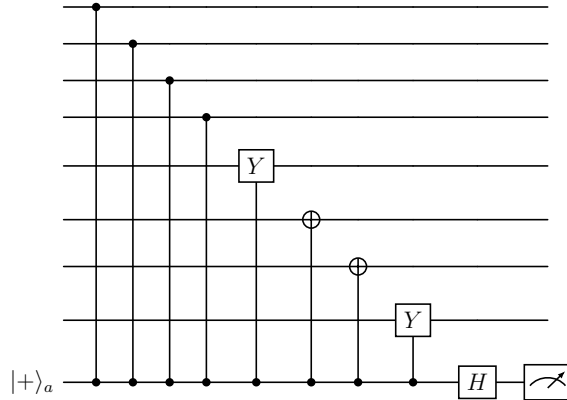


Figure 3.1: Circuit for measuring a stabilizer generator of the DK encoding. The top eight qubits are data qubits. The ancilla qubit $|+\rangle_a$ is measured to extract the eigenvalue of the stabilizer.

In order to demonstrate why this circuit works, let us analyze a simpler circuit (Fig. 3.2).

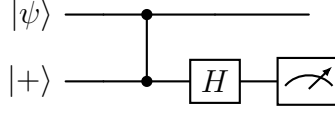


Figure 3.2: Simple demonstration of stabilizer measurement. $|\psi\rangle$ is a single-qubit initial state. This qubit is referred to as the *data qubit*. The second qubit, initialized in $|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$, is called the *ancilla qubit*.

Right before measurement, the joint state of the two qubits in this circuit is:

$$\frac{1}{2} (|\psi\rangle \otimes (|0\rangle + |1\rangle) + Z|\psi\rangle \otimes (|0\rangle - |1\rangle)) = \frac{1}{2} ((|\psi\rangle + Z|\psi\rangle) \otimes |0\rangle + (|\psi\rangle - Z|\psi\rangle) \otimes |1\rangle) \quad (3.4)$$

In general, we can write $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Then, with some algebra, the joint state of the data and ancilla qubits before the measurement becomes:

$$\alpha|0\rangle \otimes |0\rangle + \beta|1\rangle \otimes |1\rangle \quad (3.5)$$

This tells us that when we measure $|0\rangle$ on the ancilla qubit, we *know* that the data qubit is in the state $|0\rangle$, and if we measure $|1\rangle$ on the ancilla qubit, we *know* the data qubit is in $|1\rangle$. The probability of measuring $|0\rangle$ on the ancilla qubit is $|\alpha|^2$, and the probability of measuring $|1\rangle$ is $|\beta|^2$.

If we replace the CZ gate with a CX gate and represent $|\psi\rangle = \alpha'|+\rangle + \beta'|-\rangle$, we can measure the ancilla to determine if the qubit is in the state $|+\rangle$ or $|-\rangle$. Importantly, the states $|0\rangle, |1\rangle$ are the eigenstates of Z , while the states $|+\rangle, |-\rangle$ are the eigenstates of X . So, we notice that this type of a circuit will project the data qubit into the eigenspace of the controlled gate, and the measurement of the ancilla determines which eigenstate the data qubit is in.

We can also use the same ancilla qubit to project onto and “read out” the eigenstates of two data qubits at once, as shown in Fig. 3.3.

In this case, the measurement output will be $m = e_1 + e_2 \bmod 2$, where $e_1 = 0$ if $|\psi_1\rangle$ is in the $+1$ eigenstate of Z , and $e_1 = 1$ if it is in the -1 eigenstate. The same holds for e_2 , but with respect to X and $|\psi_2\rangle$. At the same time, we could think of this circuit as putting both data qubits into an eigenstate of the $Z \otimes X$ operator and the ancilla telling us which eigenvalue it corresponds to.

This can be generalized to any multi-qubit Pauli operator as shown in (Fig. 3.4).

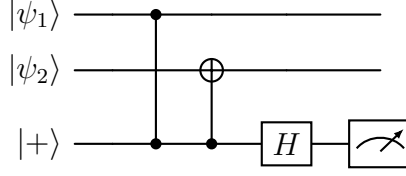


Figure 3.3: Measuring the eigenstates of two data qubits non-destructively with one ancilla.

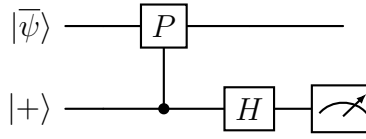


Figure 3.4: Stabilizer measurement of an arbitrary Pauli P . $|\bar{\psi}\rangle$ is a multi-qubit initial state of data qubits. The ancilla qubit is still initialized in $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$

3.1.3 Binary Symplectic Representation of Paulis

The last concept introduced in this section will be *binary symplectic representation* of Pauli operators. This is the foundational idea of stabilizer simulation techniques we use in our numerics but it will also be useful for the state preparation procedure we will discuss in the next section.

The main idea is to represent a Pauli operator as a binary vector in $\mathbb{Z}_2^n \times \mathbb{Z}_2^n$, where n is the total number of qubits in support of this operator. We will use a stabilizer generator $Z_1 Z_2 Z_3 Z_4 Y_5 Y_6 X_7 X_8$ of the DK encoding as an example with specific indexes assigned to each qubit without worrying about the bigger lattice it may be a part of for convenience (the general definition of the stabilizers can be found in Eq. (3.2)). A Pauli operator P will be represented with a vector of length $2n$ with the structure $(x_P | z_P)$. The i th component of x_P will be 1 if P acts on qubit i with X or Y , and 0 otherwise. Similarly, the i th component of z_P will be 1 if P acts on qubit i with Z or Y , and 0 otherwise. Our example stabilizer generator will map to the following:

$$Z_1 Z_2 Z_3 Z_4 Y_5 Y_6 X_7 X_8 \longleftrightarrow (00001111 \mid 11111100) \quad (3.6)$$

Now, the Pauli operators will anticommute when different single-qubit Paulis (e.g.

$\{Z_1Z_2, X_1Z_2\} = 0$) act on an odd number of qubits, and they will commute otherwise (e.g. $[Z_1Z_2, X_1X_2] = 0$). This fact makes it intuitive to see that the so-called *binary symplectic product* is equivalent to checking whether two Paulis commute or anticommute:

$$(\mathbf{x} \mid \mathbf{z}) \odot (\mathbf{x}' \mid \mathbf{z}') := \mathbf{x} \cdot \mathbf{z}' + \mathbf{z} \cdot \mathbf{x}' \pmod{2}, \quad (3.7)$$

where the dot products are over $\mathbb{F}_2$, and the result is 0 if the corresponding Paulis commute and 1 if they anticommute.

For example, consider $P = Z_1Z_2$ and $Q = X_1Z_2$, which correspond to binary symplectic forms

$$(\mathbf{x}_P \mid \mathbf{z}_P) = (00 \mid 11), \quad (\mathbf{x}_Q \mid \mathbf{z}_Q) = (10 \mid 01).$$

Then their symplectic product is

$$(00 \mid 11) \odot (10 \mid 01) = 00 \cdot 01 + 11 \cdot 10 = 0 + 1 = 1 \pmod{2},$$

so P and Q anticommute.

We are now fully equipped for discussing how to prepare fermionic basis states in the DK encoding via non-destructive stabilizer measurement.

3.2 Preparing Fermionic Basis States

Recall that fermionic basis states take the form $|x_1x_2x_3\dots\rangle$, where the binary value x_i represents the occupation of mode i —that is, $x_i = 1$ if there is a fermion in mode i , and $x_i = 0$ otherwise.

In the Jordan–Wigner (JW) encoding, these map directly to the corresponding qubit basis states: a fermionic state $|f\rangle$ is represented by the identical qubit state $|f\rangle$, where f is a binary vector indicating the occupation of modes. The variant of the ternary-tree (TT) encoding used in this work similarly maps fermionic basis states to qubit basis states, but does so via a binary linear transformation G , such that the fermionic state $|f\rangle$ is mapped to the qubit state $|Gf\rangle$.

In the Derby–Klassen (DK) encoding, the fermionic basis states are also mapped directly to the vertex qubits: the fermionic state $|f\rangle$ is mapped to the vertex qubit state $|f\rangle$. However, a key distinction is the presence of additional face qubits. These enlarge the qubit

Hilbert space beyond the dimension of the target fermionic state space, requiring us to impose constraints on the physical qubit states to ensure a valid fermionic representation. For now, we assume perfect operations with no noise.

The following constraints are given in [55]:

$$V_j|\psi_f\rangle = (-1)^{f_j}|\psi_f\rangle, \quad j \in [L^2], \quad (3.8)$$

$$S_k|\psi_f\rangle = |\psi_f\rangle, \quad k \in [L^2/2], \quad (3.9)$$

where $f \in \mathbb{F}_2^{|\mathcal{L}_v|}$, and S_k and V_j are the stabilizer generators and logical operators of the DK encoding, respectively.

The second constraint ensures the state lies in the +1 eigenspace of the stabilizer group as required by definition of a stabilizer code. The first enforces the correct fermionic occupations on the vertex qubits. More explicitly, from the definition of V_j in Eq. (2.47), we have $a_j^\dagger a_j = \frac{I - V_j}{2}$. Thus, while $a_j^\dagger a_j |\dots 1 \dots\rangle = |\dots 1 \dots\rangle$ the operator V_j acts as:

$$V_j |\dots 1 \dots\rangle = -|\dots 1 \dots\rangle. \quad (3.10)$$

As noted in [55], DK-encoded fermionic basis states factorize as

$$|\psi_f\rangle = |v\rangle \otimes |\phi\rangle, \quad (3.11)$$

with $|v\rangle \in (\mathbb{C}^2)^{\otimes |\mathcal{L}_v|}$ and $|\phi\rangle \in (\mathbb{C}^2)^{\otimes |\mathcal{L}_f^r|}$.

Let R_k and T_k denote the restrictions of S_k to the vertex and face qubits, respectively. Then,

$$\begin{aligned} R_k|v\rangle &= r_k|v\rangle, \\ T_k|\phi\rangle &= t_k|\phi\rangle, \\ r_k &= t_k, \quad k \in [L^2/2]. \end{aligned} \quad (3.12)$$

Consider first the case $f = \mathbf{0}$ (the vacuum). Preparing the vertex qubits in the $|0\rangle$ state satisfies the V_j constraint and the R_k part of the stabilizer constraint. If the face qubits are also initialized in $|0\rangle$, then measuring T_k projects the state onto the +1 eigenspace with 50% probability. A naïve strategy is to postselect on all +1 outcomes, but this becomes exponentially unlikely as the system size grows.

Instead, the total Hilbert space is partitioned into equivalent subspaces—called *cospaces*—indexed by the syndrome $\sigma \in \{\pm 1\}^{|\mathcal{L}_f^r|}$. Each syndrome defines a cospace with stabilizers

$S'_k = \sigma(k)S_k$, and logical operators may need to be modified accordingly (e.g., by flipping signs of E_{ij} operators).

Given σ , we can construct a *recovery operator* D_σ that maps the state back to the canonical codespace $\mathcal{C}_f$, or alternatively, is used to conjugate all logical operators:

$$E_{ij} \mapsto D_\sigma E_{ij} D_\sigma. \quad (3.13)$$

Since all operators involved are Pauli operators, we have $D_\sigma = D_\sigma^\dagger$.

We seek a recovery operator D_σ such that it **anticommutes** with each S_k for which $\sigma(k) = -1$, **commutes** with the remaining S_k , and **commutes** with all logical operators V_j . This ensures fermion number is preserved. The binary symplectic formalism (see Section 3.1.3) allows us to translate these into linear constraints over $\mathbb{F}_2$.

Let n_q be the number of total qubits, n_v the number of vertex qubits, and $n_s = |\mathcal{L}_f|$ the number of stabilizer-generator faces. Let σ_b be the binary representation of σ , where $\sigma(k)_b = 0$ if $\sigma(k) = +1$ and $\sigma(k)_b = 1$ if $\sigma(k) = -1$. Define the constraint vector e as:

$$e = (\sigma(1)_b, \sigma(2)_b, \dots, \sigma(n_s)_b \mid \underbrace{0, 0, \dots, 0}_{n_v}) \quad (3.14)$$

Let the recovery operator D be represented by the binary symplectic vector

$$p_D = (x_1, x_2, \dots, x_{n_q} \mid z_1, z_2, \dots, z_{n_q}) \quad (3.15)$$

Let S be the $2n_q \times n_q$ matrix whose columns are the symplectic binary representations of the S_k and V_j :

$$S = \left(\begin{array}{c|c|c|c|c} \begin{array}{c} | \\ S_1 \\ | \end{array} & \begin{array}{c} | \\ S_2 \\ | \end{array} & \cdots & \begin{array}{c} | \\ S_{n_s} \\ | \end{array} & \begin{array}{c} | \\ V_1 \\ | \end{array} & \begin{array}{c} | \\ V_2 \\ | \end{array} & \cdots & \begin{array}{c} | \\ V_{n_v} \\ | \end{array} \end{array} \right) \quad (3.16)$$

Let Λ be the $2n_q \times 2n_q$ matrix defining the symplectic product:

$$\Lambda = \begin{pmatrix} 0 & I_{n_q} \\ I_{n_q} & 0 \end{pmatrix} \quad (3.17)$$

where I_{n_q} is the $n_q \times n_q$ identity matrix.

The recovery condition is then expressed as:

$$p_D \Lambda S = e \pmod{2} \quad (3.18)$$

Solving this system over $\mathbb{F}_2$ gives the recovery operator D . We can do this efficiently with Gaussian elimination. Alternatively, we could pre-compute recovery operators for each one stabilizer generator separately and multiply those based on the syndrome.

The case $f \neq \mathbf{0}$ is only slightly more complicated. First, compute the eigenvalues r_k such that $R_k|f\rangle = r_k|f\rangle$. After measuring T_k and obtaining outcomes t_k , the syndrome is recovered as $\sigma_k = r_k t_k$.

An additional simplification can be made by preparing face qubits differently. Suppose face qubits on even rows are initialized in $|+\rangle$ and those on odd rows in $|+i\rangle$. This prepares the state in known ± 1 eigenstates of stabilizers centered on even rows, so only the T_k on odd rows need to be measured.

More encoded states can be prepared similarly. Consider a set of edge operators $\{E_{\ell m}\}$ with disjoint support. For each $E_{\ell m}$, prepare its support qubits in a $+1$ eigenstate of $E_{\ell m}$ —e.g., if $E_{\ell m} = X_\ell Y_m X_{f(\ell, m)}$, then prepare $|+\rangle|+i\rangle|+\rangle$. Measure the stabilizers, record the syndrome, and update the edge operators as before. Note that in this case, the full set $\{S_k\}$ (not just $\{T_k\}$) must be measured. This method also generalizes to preparing -1 eigenstates of edge operators.

3.3 Trotterization Circuits

How do different encodings affect the Trotterized evolution implementation?

As hinted in Section 2.3, the Pauli weights produced by different encodings directly affect the depth and gate counts of quantum circuits. For each term $e^{-i\tau P_i}$ in the Trotterized evolution, where P_i is a Pauli operator of weight w_{P_i} , the standard decomposition uses:

- 1 arbitrary single-qubit rotation gate,
- $2w_{P_i}$ single-qubit Clifford gates, and
- $2(w_{P_i} - 1)$ two-qubit Clifford gates.

Let N_H be the number of terms in the Hamiltonian and let $\bar{w}$ denote the average Pauli weight across all terms. In the Fermi-Hubbard model on an $L \times L$ square lattice, we have $N_H \sim L^2$. Then, the total gate count for the Trotterized evolution with T steps scales as

$$\text{Gate count} = \mathcal{O}(TL^2\bar{w}). \quad (3.19)$$

Lower-weight encodings yield smaller values of $\bar{w}$, resulting in shallower and more efficient circuits.

How many Trotter steps are needed for a useful simulation? There is no simple answer. As discussed in Section 2.3, a rough error bound tells us that to simulate Fermi-Hubbard evolution for time $t = 1$ with error $\epsilon = 0.01$, one would need approximately $T = 100$ steps. Alternatively, the same number of steps would give a precision of $\epsilon = 10^{-4}$ for a shorter time $t = 0.1$. We note that tighter and more model-specific error bounds for the Fermi-Hubbard model can be derived based on the structure and magnitudes of the Hamiltonian coefficients [88].

Naively, implementing Trotterization for Fermi-Hubbard with JW results in the deepest circuits since the encoded weights scale linearly with the size of the lattice. TT encoding’s weights scale logarithmically, while the DK encoding’s weights are constant. However, there is a known technique that dramatically improves the depth of JW circuits called fermionic SWAP (FSWAP) network.

3.3.1 FSWAP Networks for Improved Jordan-Wigner Encoding

The key idea is to dynamically “move” qubits so that operators are applied only between neighboring qubits, where the JW-encoded terms have minimal weight. The Fermi-Hubbard model on a 2D square lattice is particularly amenable to this optimization due to its regular structure. In fact, the optimal FSWAP protocol for an $M \times N$ lattice is known to require only $2M - 1$ swap layers [48].

The FSWAP operation is simply the regular SWAP but with fermionic asymmetry taken into account:

$$\text{FSWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (3.20)$$

The FSWAP can be decomposed into the Clifford CZ and SWAP:

$$\text{FSWAP} = \text{CZ} \cdot \text{SWAP} \quad (3.21)$$

In Fig. 3.5, we show non-optimal but simple swap ordering we used in our numerical results. This allows us to apply hopping terms between vertical neighbors without long chains of Z that we would have to include otherwise.

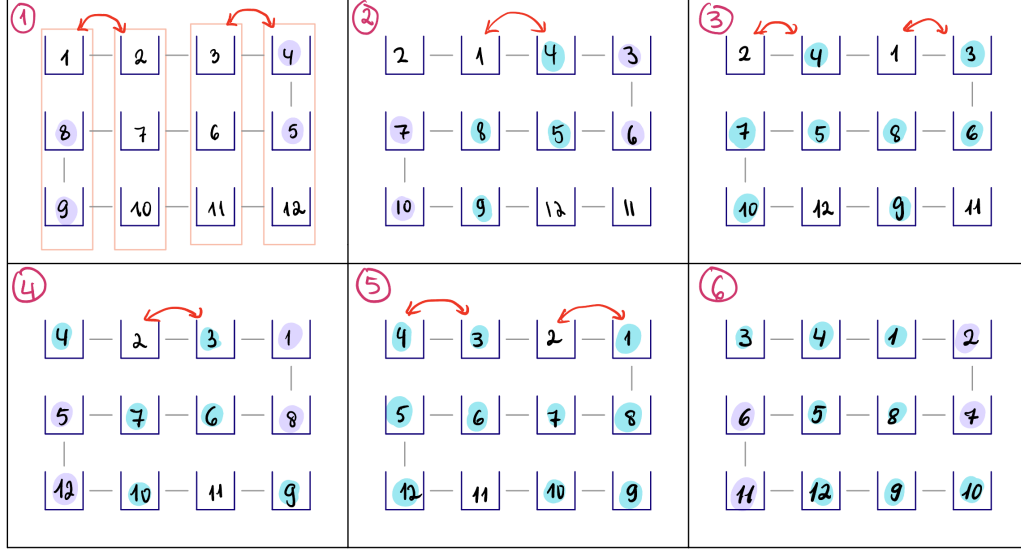


Figure 3.5: FSWAP network on a 3×4 lattice for implementing hopping terms between all vertical neighbors. In panel 1, we start with qubits numbered 1 through 12 and a snake-like JW ordering identified with gray lines. In this initial configuration, we can apply low-weight hopping terms between pairs (4,5) and (8,9), identified with purple highlighting. Finally, the two orange arrows show which columns will be swapped (via FSWAP application) to achieve the configuration in panel 2. Then in panel 2, we are able to apply hopping terms between (3,6) and (7,10). We identify pairs of qubits that have already been “coupled” previously with blue highlighting. Last configuration is different from the original ordering which could be mitigated by applying three more layers of swaps or postprocessing results based on the permuted ordering.

3.4 Observable Readout and Error Mitigation

3.4.1 Observables

Recall from Section 2.2 that we often would like to calculate a particular physical quantity from the evolved state $\psi(t)$. In order to do so, we would compute expectation values of a certain observable $\mathcal{O}$ by repeating the evolution and measurement circuit many times and averaging the results.

In the context of the Fermi-Hubbard model, measuring the occupation of modes, i.e. the operators $a_j^\dagger a_j$ for all indices of modes j is a common objective. In all three encod-

ings, this amounts to measuring the qubits in the computational (Z) basis (notably, in the DK encoding, we would only require measurement results from the vertex qubits). However, it can also be useful to measure the total energy of the state which implies using the total Hamiltonian as the observable of interest. Luckily, we can estimate such a complicated observable by measuring the terms separately. This means we would repeat the same evolution circuit with different measurements at the end, and add the results at postprocessing.

This means we would like to find a measurement procedure that allows us to estimate the measurement results of as many of the operators $a_i^\dagger a_i a_j^\dagger a_j$ (Coulomb) and $a_i^\dagger a_j + a_j^\dagger a_i$ (hopping) as possible with running as few different circuits as possible. In this work, we will focus on how a low-weight encoding such as the DK encoding, can allow for simultaneous measurement of many of the qubit observables due to their *qubit-wise commutativity*³.

Recall that JW encoding maps the hopping term to $\frac{1}{2} \left(\prod_{k=i+1}^{j-1} Z_k \right) (X_i X_j + Y_i Y_j)$ and the Coulomb term to $\frac{1}{4} (I + Z_i Z_j - Z_i - Z_j)$. The Coulomb term can be estimated by the occupation measurement, i.e. measuring all qubits in the computational basis. The hopping term measurement is a bit more complicated. Measuring high-weight encoded operators naively quickly becomes impractical. One way to overcome this would be to apply the same FSWAP network technique described in Section 3.3.1. However, more efficient methods can be created. In a square-lattice Fermi-Hubbard Hamiltonian encoded with the JW encoding, it is known how to extract the energy with 5 measurements using the procedure outlined by [16]. This procedure employs a simple, but non-Clifford, diagonalization circuit for the two parts of the hopping term. To the best of our knowledge, no efficient measurement procedure is known for ternary tree encoding.

The low weight of the Hamiltonian terms in the DK encoding simplifies the hopping term measurement. In order to measure all the operators in the Fermi-Hubbard Hamiltonian, we need to measure in nine different measurement bases. First, we can simultaneously extract the local occupation number operators $a_i^\dagger a_i$ and the Coulomb operators $a_i^\dagger a_i a_j^\dagger a_j$ by measuring the vertex qubits in the Z basis. After that, we can simultaneously measure the non-overlapping parts of the hopping operators using eight sets of measurements as shown in Fig. 3.6.

³Qubit-wise commutativity implies Pauli operators that commute with each other for each qubit in their support (e.g. $X_1 X_2$ and $X_2 X_3$). Pauli's could commute without qubit-wise commutation, e.g. $Z_1 Z_2$ and $X_1 X_2$. In this case, one could create a unitary that would put qubits into a simultaneous eigenbasis of these two Pauli operators. If such a unitary is applied before measurement, these two operators would be measured simultaneously as well. However, finding such unitaries is not a trivial task in general.

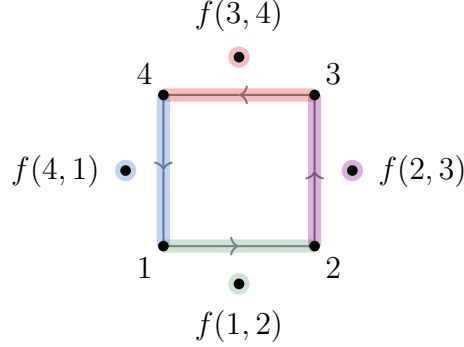


Figure 3.6: Efficient DK measurement schedule. For each stabilizer in the lattice, we can measure the hopping term operators corresponding to the same color simultaneously. Each highlighted edge corresponds to two operators in the hopping term: $a_i^\dagger a_j$ and $a_j^\dagger a_i$; see Eq. (2.50).

3.4.2 Global Parity Postselection

As discussed in Section 2.4, evolution under the Fermi–Hubbard Hamiltonian conserves the total number of fermions. Therefore, for all encoding schemes considered in this work, we can perform a simple form of error mitigation that we refer to as *global parity postselection*.⁴

When measuring the qubits in the computational (Z) basis, we are effectively measuring the occupation numbers of the fermionic modes. Hence, we can postselect on the measurement outcomes by discarding any runs in which the total fermion number differs from that of the initial state.

3.4.3 Error Detection with Stabilizer Measurement

The most general error detection method available for the DK encoding is to measure the stabilizer generators non-destructively at the end of the circuit as described in Section 3.1.2. We can then compare the error syndrome to the one measured at the state preparation, and discard any measurement results where the two syndromes differ. This strict postselection will ultimately result in discarding all results with increasing error rates/lattice

⁴More generally, fermionic systems are parity-conserving: no physical fermionic Hamiltonian can change the total fermion number from even to odd or vice versa. Even in exotic models where fermions may be created or annihilated, this can only occur in pairs. Thus, this form of error mitigation remains valid beyond the Fermi–Hubbard model. This is why we refer to it as *parity* postselection rather than *number* postselection.

sizes. Assessing if any useful results can be obtained before reaching that point is one of the questions we address in our numerical study. In [82], a method was proposed for recovering information from measurement results with non-trivial syndrome outcomes—that is, runs in which errors were detected. However, it remains unclear how well this approach would scale to circuits of greater depth. Although we did not test this method in our setup, it is in principle possible to apply it to the output of stabilizer simulations.

We can also use a flag qubit [20] to detect when high-weight errors propagate to the data qubit; see Fig. 3.7.

The flag qubit detects all weight ≥ 2 data qubit errors except for the stabilizer itself and errors that are stabilizer equivalent to weight 1 errors. However, a single fault on the auxiliary measurement qubit after all of the control gates can give an incorrect measurement result. Assessing effectiveness of flag qubits was another objective of our numerical study.

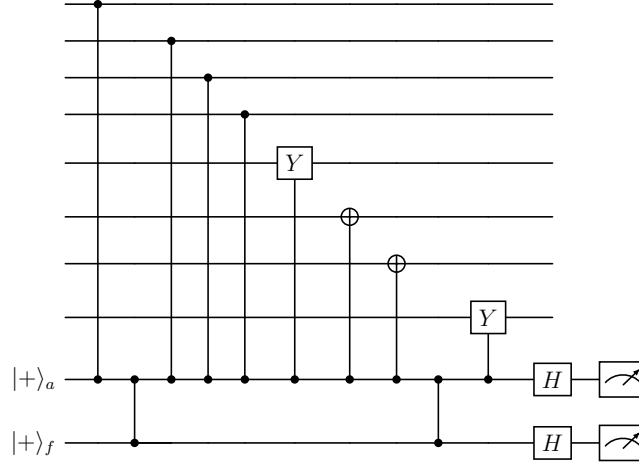


Figure 3.7: Circuit for measuring a stabilizer generator of the DK encoding with flags. The top eight qubits are the data qubits. The auxiliary qubit $|+\rangle_a$ is read out to give the measurement result and the auxiliary qubit $|+\rangle_f$ is a flag qubit that is used to detect high-weight errors.

3.4.4 Stabilizer Reconstruction

Last but not least, we describe a cost-effective technique for detecting errors in the readout of occupation number and Coulomb interaction terms using the DK encoding.

Consider a subset of vertex and edge operators with non-overlapping support. Any subset with this property can be measured simply by measuring the relevant qubits in the appropriate Pauli bases. For example, to measure all of the vertex operators we simply measure all of the vertex qubits in the Z basis. For this measurement in particular, we can also extract a subset of the stabilizer eigenvalues. Suppose we additionally measure the face qubits in even (odd) rows of the lattice in the X (Y) basis. Then by combining the relevant measurement outcomes, we can reconstruct the eigenvalues of the stabilizer generators associated with faces in even rows of the lattice. We can then discard any runs with stabilizer measurement outcomes that do not match the outcomes observed in state preparation. We call this readout technique *stabilizer reconstruction* (SR).

When using SR, we can reduce the number of stabilizer generators measured at the end, since we are essentially reading out half of the syndrome from a necessary destructive measurement. Then in order to obtain the second half of the syndrome, we can measure the operators T_k (from Section 3.2) that are not included in the additional destructive measurement of face qubits in the end.

Chapter 4

Stabilizer Simulation for Testing Fermionic Encodings' Performance

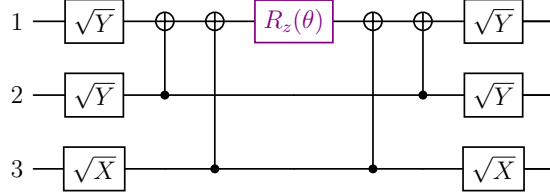
In this chapter, we will present numerical results comparing JW, TT and DK encodings' performance in noisy implementation of Trotterization algorithm. In order to study the performance of these methods in regimes potentially inaccessible to classical computation, we utilize stabilizer simulation—a common and extremely powerful numerical tool in QEC research.

4.1 Stabilizer Simulation

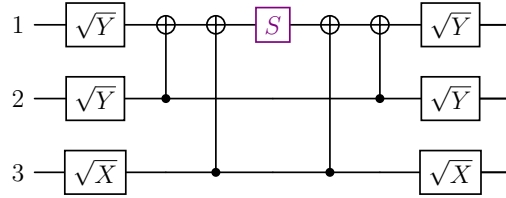
Stabilizer simulations are routinely used to benchmark the performance of quantum error-correcting codes with thousands of qubits, see e.g. [37, 46, 99, 15]. Stabilizer simulations can simulate logical operations as long as they are Clifford, and capture limited form of noise, namely stochastic Pauli channels.

When using stabilizer simulations to benchmark the performance of fermionic encodings we also have the above limitations. However, we argue that simulations in this restricted model can nevertheless serve as a reasonable proxy for performance. First, note that the state preparation and readout circuits we discussed fall within the stabilizer formalism. Requiring the logical operator circuits to be Clifford restricts the possible rotation angles we can choose in Fig. 4.1 but we emphasize that the only part of these circuits that changes for different choices of rotation angle is one single-qubit gate, and therefore we would expect the Clifford circuits to accurately capture the error propagation in the generic circuits. We

also numerically confirm that the performance of such Clifford and non-Clifford circuits is identical in the stochastic Pauli error model via calculating process fidelities for varying error rates; see Fig. 4.2.



(a) Example decomposition of the DK encoding (half) hopping operator $e^{-i\theta X_i X_j Y_{f_{i,j}}}$ into 1- and 2-qubit gates.



(b) Clifford version of (a) corresponding to $e^{-i\frac{3\pi}{4} X_i X_j Y_{f_{i,j}}}$

Figure 4.1: The DK hopping operator and its Clifford version we use in our stabilizer simulation.

The restriction to stochastic Pauli noise is perhaps the biggest limitation of our simulation, as this does not capture important device noise such as leakage [103, 9], amplitude damping [80, 97, 24], or coherent errors. Coherent errors in particular are damaging in deep circuits as they accumulate adversarially with increasing circuit depth. However, we note that arbitrary Markovian errors (including coherent errors) can be tailored to stochastic Pauli errors using randomized compiling [101, 51], and we expect our simulations to be especially predictive for circuits compiled in this way.

Our numerical results are enabled by STIM [40], a commonly used and powerful tool for testing QEC codes by constructing noisy Clifford circuits and sampling their outcomes under varying circuit-level error models. The advantages of using STIM compared to prior approaches assessing noise resistance of fermionic encodings [22, 49] are: 1) the ability to simulate much larger sizes; 2) the flexibility to define hardware-inspired error models; and

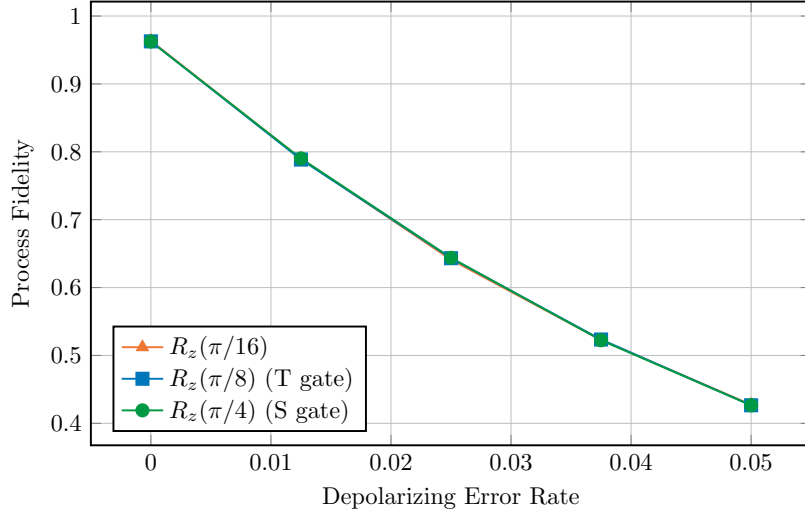


Figure 4.2: Comparison of the process fidelity of Clifford and non-Clifford logical rotations, calculated using QISKIT’s process tomography functionality [59]. Each data point is the average value of five runs with 1000 shots each (the error bars are too small to be visible).

3) the requirement to construct explicit noisy Clifford circuits resembling circuits that one would run on a device.

The code is available at <https://github.com/edyrenkova/FermiStabilizers>.

4.2 Error models

We consider two error models: Standard Depolarizing (SD) and Superconducting Inspired (SI), which are described in Table 4.1. Variants of the SD model are often used as a benchmark in QEC research, while SI is meant to represent the errors that are common in superconducting devices and was itself inspired by [41]. We do not consider idling errors in this work as adding idling errors would disadvantage encodings with high-weight logical operators, which are mostly applied sequentially in our simulations.

Noise type	Definition
p_2	A 2-qubit depolarizing channel with rate p_2 is added after each 2-qubit Clifford gate.
p_1	A single-qubit depolarizing channel with rate p_1 is added after each 1-qubit Clifford gate.
p_s	Initialization to $ 0\rangle$ is followed by a bitflip channel with rate p_s .
p_m	All measurements are done in the Z basis and their results are flipped at a rate p_m .

(a) Definitions of error types

Name	Standard Depolarizing	Superconducting Inspired
Noisy Gateset	$p_2 = p$ $p_1 = p$ $p_s = p$ $p_m = p$	$p_2 = p$ $p_1 = p/10$ $p_s = 2p$ $p_m = 5p$

(b) Comparison of noisy gatesets in SD and SI

Error Regime	Definition
Aspirational	$p = 0.0001$, or 0.01%
Intermediate	$p = 0.0005$, or 0.05%
Near future	$p = 0.001$, or 0.1%

(c) Error regimes that we simulated

Table 4.1: Error models

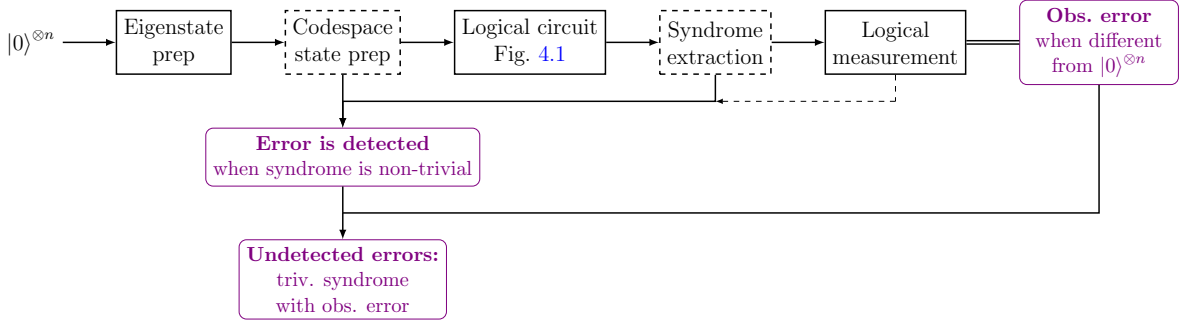


Figure 4.3: **Simulated Circuit Structure.** Dashed elements are only present in some circuits. Codespace state preparation is only present in DK circuits. Syndrome extraction is only present in the DK circuits where non-destructive syndrome measurement is performed. Logical measurements contain information used for postselection in JW global parity postselection and stabilizer eigenvalue readout in DK described in Section 3.4.4. Purple boxes indicate postselection steps from classical data.

4.3 Simulated Circuits

The overview of the structure of our circuits can be found Fig. 4.3. The main idea is to create a noisy version of the circuit that implements

$$UU^\dagger|\psi_{\text{init}}\rangle = |\psi_{\text{init}}\rangle. \quad (4.1)$$

Applying the mirror circuit $UU^\dagger$ instead of just U is the simplest way to make sure that the final logical measurement is deterministic without noise, which is a requirement for STIM simulations. Therefore, $|\psi_{\text{init}}\rangle$ is chosen to be an eigenstate of the final measurement basis. For simplicity, in our simulations $|\psi_{\text{init}}\rangle$ is always the +1 eigenstate of the measured basis, but that is not a requirement and does not affect the outcomes.

Eigenstate preparation in Fig. 4.3 refers to single qubit rotations that put physical qubits into the +1 eigenstate of the final logical measurement. *Codespace state preparation* refers to the methods described in Section 3.2. *Syndrome extraction* is achieved through non-destructive stabilizer measurement circuit Fig. 3.7. Depending on the error mitigation method, the syndrome could be a combination of measurement results from codespace state preparation, syndrome extraction and logical measurement. The *logical circuit* block in Fig. 4.3 refers to $UU^\dagger$ in Eq. (4.1), where U represents a logical computation performed with an encoding ($U^\dagger$ is automatically generated with STIM and is always counted towards the total depth of the circuit). In our simulations, we consider two example logical computations, which we refer to as random logical circuits and full Trotter circuits. For random

logical circuits we apply some number of rotations of Hamiltonian terms (generated automatically using CIRQ [32]) in random order by drawing terms uniformly at random from the Hamiltonian without replacement [22]. For full Trotter circuits, we apply all Hamiltonian terms some number of times. In this case, there is an optimized circuit for the JW encoding utilizing fermionic swap networks [66, 93, 48], which we can simulate using STIM as the fermionic swap operation is a Clifford operator.

For completeness, we provide tables of one- and two-qubit gate counts for all cases in Appendix A.

4.4 Sampling

For each quantum circuit, a sample is represented by two binary vectors: the syndrome vector $\mathbf{s}$ and the observables vector $\mathbf{o}$. The syndrome vector $\mathbf{s} = (s_1, s_2, \dots, s_n)$, where $s_i \in \{0, 1\}$, encodes the detection of errors in the system. The error detection rate is defined as

$$R_{\text{det}} = \frac{n_{\text{discard}}}{n_{\text{samp}}}, \quad (4.2)$$

where n_{samp} is the total number of samples in the simulation and n_{discard} is the number of samples where $\mathbf{s}$ has at least one nonzero entry.

After postselection, the total number of samples used for calculating observable error rates is denoted as $n_{\text{samp,post}}$:

$$n_{\text{samp,post}} = n_{\text{samp}} - n_{\text{discard}}. \quad (4.3)$$

The observables vector $\mathbf{o} = (o_1, o_2, \dots, o_m)$, where $o_j \in \{0, 1\}$, tracks the outcomes of measured observables. A sample is counted toward the $R_{\text{obs,any}}$ if $\mathbf{o}$ contains at least one non-zero entry, and this rate is calculated as

$$R_{\text{obs,any}} = \frac{\text{Number of samples with any } o_j = 1}{n_{\text{samp,post}}}. \quad (4.4)$$

Additionally, the worst local observable rate $R_{\text{obs,worst}}$ is determined as the maximum local observable rate, where the local observable rate for observable j is defined as

$$R_{\text{obs,local}}^{[j]} = \frac{\text{Number of samples with } o_j = 1}{n_{\text{samp,post}}} \quad (4.5)$$

and

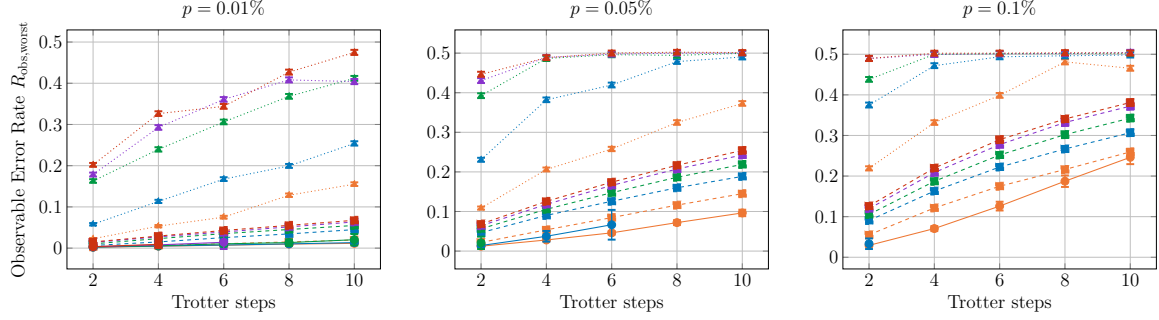
$$R_{\text{obs,worst}} = \max_j (R_{\text{obs,local}}^{[j]}). \quad (4.6)$$

In all simulations, $n_{\text{samp}} = 100,000$. Error bars are computed using bootstrap resampling with 1000 resamples [34]. The error bars represent variance of error rates for the same circuit. However, since each circuit is also generated by randomly selecting the order Hamiltonian terms applied and an automatic optimization by Cirq is applied, there is variation in the effective depth (i.e. gate counts) of circuits that is not captured by the error bars calculation. The random order of Hamiltonian terms is consistent between encodings to ensure a fair comparison.

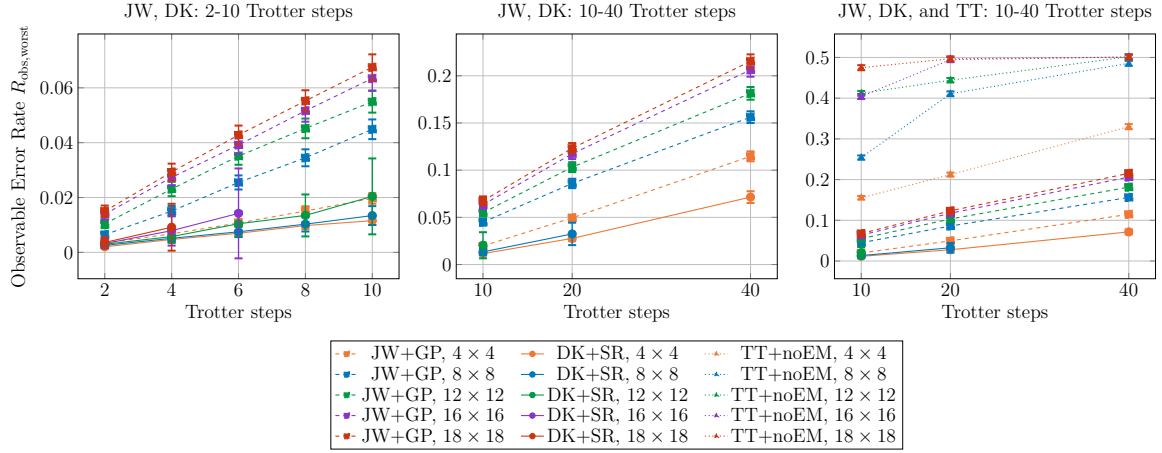
4.5 Results

First, we consider the performance of the JW, DK, and TT encodings with full Trotter circuits and occupation number measurements. This means that the observable vector $\mathbf{o} = (o_1, o_2, \dots, o_m)$ (see Section 4.4) records the outcomes of computational basis (Z -basis) measurements on each qubit (for DK, this includes only vertex qubits). For JW, we implement fermionic swap network circuits (Section 3.3.1) and global parity postselection (Section 3.4.2). For DK, we implement stabilizer reconstruction as described in Section 3.4.4. We compare these to unoptimized, unmitigated TT encoding, which is an asymptotically optimal one-to-one encoding. From Fig. 4.4, it is evident that the optimized JW and DK circuits produce less noisy results than TT. The winner between JW and DK is less clear. While DK with stabilizer reconstruction (SR) shows a clear accuracy improvement, this method quickly exhausts the sample budget $n_{\text{samp}} = 100,000$ with increasing circuit depth and physical error rate. This is reflected in the incompleteness of DK curves, as we include only data points calculated from at least 500 samples. The most promising results obtained with our naive DK circuit construction were 10 Trotter steps for a 12×12 lattice or 6 Trotter steps for a 16×16 lattice at the lowest physical error rate of $p = 0.01\%$. But in that regime, the accuracy returns compared to JW are diminishing.

Next, we analyze the performance of various postselection methods for the DK encoding in Fig. 4.5. We simulate random logical circuits and both occupation number and hopping term measurements. For the hopping term measurements, the observable vector $\mathbf{o}$ corresponds to measurement results of a multi-qubit observable $X_i X_j X_{f(i,j)}$, arbitrarily chosen from the four possible encoded hopping operators listed in Eq. (2.50). Since we estimate the hopping term expectation by measuring all four separately, stabilizers must be measured non-destructively to obtain the syndrome vector $\mathbf{s}$ (see Section 3.4.3). We

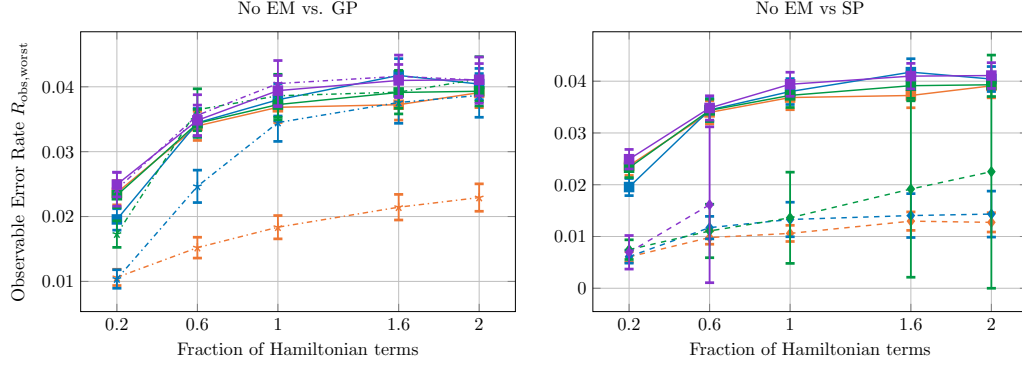


(a) Standard depolarizing error model at different error rates

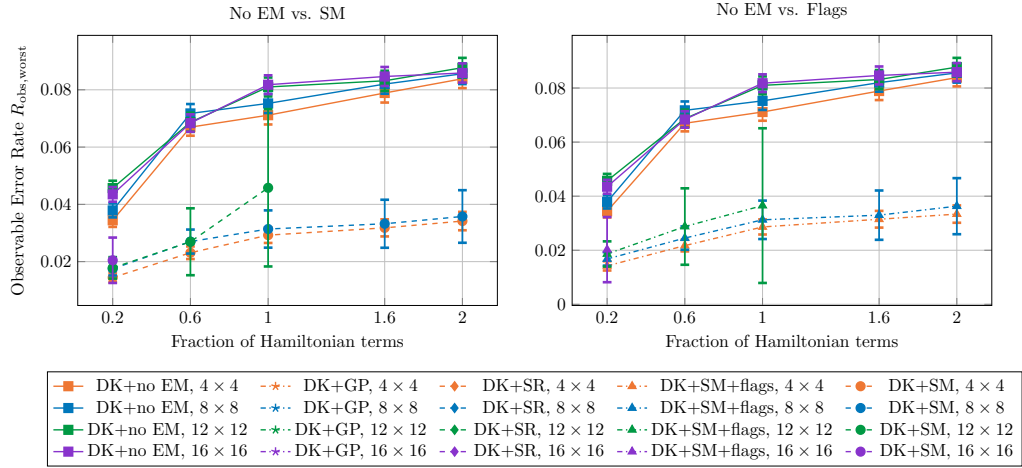


(b) Standard depolarizing error model at $p=0.01\%$

Figure 4.4: Performance of fermionic encodings in the standard depolarizing error model. We consider the DK, JW, and TT encodings, with no error mitigation (no EM), stabilizer reconstruction (SR) and global parity postselection (GP). One Trotter step corresponds to an application of all the Hamiltonian terms, and we measure the occupation number at the end of the circuit. $R_{\text{obs,worst}}$ is the maximum local observable error rate and data points with error detection rate $R_{\text{det}} \geq 0.995$ are excluded. Error bars are computed using bootstrap resampling with 1000 resamples [34]. (a) compares encoding performance across error rates. (b) zooms in on the lowest error rate, $p = 0.01\%$: the left plot magnifies DK and JW data from (a), the center compares deeper circuits for DK and JW, and the right includes all three encodings (note the y-axis scale change).



(a) Observable errors in occupation number basis at $p = 0.05\%$ (standard depolarizing error model)



(b) Observable errors in hopping operator basis at $p = 0.05\%$ (standard depolarizing error model)

Figure 4.5: **Performance of various postselection methods in the DK encoding (SI error model).** Each circuit applies a random sequence of logical operators, with operator count given as a fraction of Hamiltonian terms. Error bars are computed using bootstrap resampling with 1000 resamples. (a) and (b) showcase improvements in accuracy enabled by global parity postselection (GP) and stabilizer reconstruction (SR) for occupation number measurements, as well as stabilizer measurement (SM) and stabilizer measurement with flag qubits (SM+flags) for hopping term measurements, compared with no error mitigation (no EM). Data points with error detection rate $R_{\text{det}} \geq 0.995$ are excluded. The legend in (b) is shared by (a) and (b).

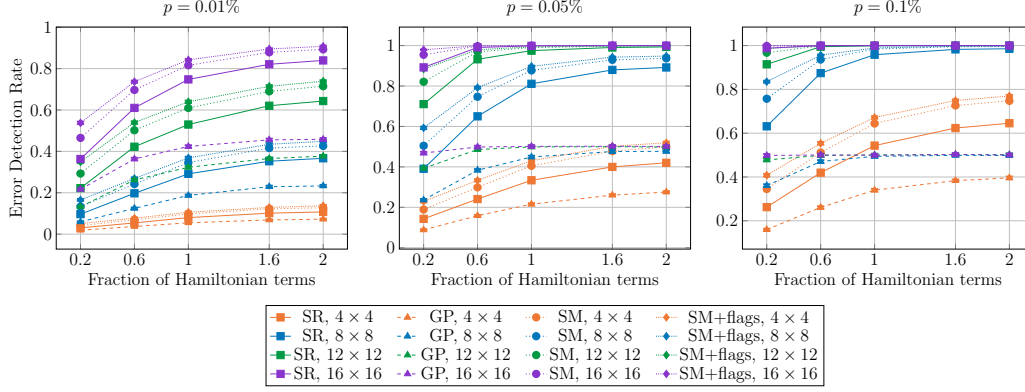


Figure 4.6: Error detection rates at different physical error rates (standard depolarizing error model).

choose an error rate of $p = 0.05\%$ in Figs. 4.5a and 4.5b to illustrate the tradeoff between accuracy and sampling requirements for different methods. The two plots in Fig. 4.5a highlight the advantage of stabilizer reconstruction over global parity postselection in the DK encoding. Accuracy is improved due to the finer-grained error detection enabled by SR. This is also evident in Section 4.5, where GP saturates at $R_{\text{det}} = 0.5$ because a fully random occupation number measurement yields even global parity half of the time and odd global parity the other half. The two plots in Fig. 4.5b compare stabilizer measurement postselection with and without flag qubits in the hopping term measurement basis. We observe little difference between flagged and non-flagged stabilizer measurements in both observable error rates and error detection rates (see Section 4.5).

In Appendix B, we provide additional results for the superconducting inspired error model, hopping term measurements, and GP with the JW encoding.

Chapter 5

Conclusion

In this work we used modern stabilizer simulation tools to estimate the performance of fermionic encodings on near-term devices. We were able to assess the performance of encodings in regimes that are potentially impractical for classical computation [4, 84], reaching lattice sizes of up to 18×18 and up to 40 Trotter steps. To achieve this our error model was necessarily restricted to stochastic Pauli noise, but realistic noise in devices can be tailored to this form at little or no experimental cost.

The limitations in our simulations arose from the performance of the encodings themselves, not from computational resource constraints. The maximum circuit depths and system sizes we report are determined by the error mitigation capacity of the encodings, rather than by any inability to carry out stabilizer simulations. The most computationally demanding aspect of our simulations was circuit generation, which could be further optimized. We utilized high-performance computing resources to test various combinations of encodings, circuits, and error models. However, we emphasize that results for a single set of simulation parameters can be efficiently obtained on a standard laptop. The code and setup instructions are publicly available at <https://github.com/edyrenkova/FermiStabilizers>.

In our simulations, we observed that the optimal average operator weight of the TT encoding cannot compete with depth reductions provided with specialized methods such as the JW encoding with fermionic swap networks and the DK encoding with error mitigation. We also observed that although the DK encoding with error detection provides an accuracy improvement, the returns are diminishing due to the high sampling requirements. The most promising regime for the DK encoding is the aspirational regime ($p = 0.01\%$). However, even in this case the DK encoding exceeds our sample budget of 100,000 shots at 10 Trotter

steps for the 12×12 lattice or 6 Trotter steps for the 16×16 lattice. Furthermore, in the aspirational regime, the JW encoding with fermionic swap networks and global parity postselection also provides relatively low-error results.

Overall, our results indicate that reductions in operator locality and error mitigation enabled by the DK encoding are not enough to enable significant improvements in near-term condensed matter simulation involving Trotterization. However, recent work [82] gives more optimized circuits and improved error detection techniques for the DK encoding, which we would expect to improve our results. Considering the superior performance of the DK encoding in the aspirational regime, we could consider concatenating the DK encoding with a quantum error-correcting code such as the surface code. The surface code layer could be used to reduce the logical error rate sufficiently so that the DK encoding postselection would be effective. This combination could have superior resource requirements to either encoding on its own, and is compatible with stabilizer simulation.

Another next step for our work could be to test the validity of our simulations by implementing our circuits on quantum hardware, where the correct choice of encoding for a particular device will depend both on the characteristics of the device and the nature of the application. For example, our circuits assume auxiliary qubits that can couple to eight data qubits, which is not possible on, e.g., IBM's devices with heavy hex connectivity. We may therefore need to consider compact encodings defined on other lattices [30] in this case. But our simulation tools are flexible and could be used to estimate the performance of other encodings with minor modifications. After selecting a device and application, it would be interesting to compare stabilizer simulations with device performance, both with and without randomized compiling. We expect randomized compiling to improve agreement between simulations and experiments, as well as enhance the performance of fermionic encodings with error detection. While achieving quantum advantage with fermionic encodings in the near term is uncertain, stabilizer simulations will remain crucial for assessing quantum circuit performance at the scale where quantum advantage is possible.

References

- [1] The Hubbard model at half a century. *Nature Physics*, 9(9):523–523, 2013.
- [2] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70:052328, November 2004.
- [3] Rajeev Acharya, Dmitry A. Abanin, Laleh Aghababaie-Beni, Igor Aleiner, Trond I. Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham Asfaw, Nikita As-trakhantsev, Juan Atalaya, Ryan Babbush, Dave Bacon, Brian Ballard, Joseph C. Bardin, Johannes Bausch, Andreas Bengtsson, Alexander Bilmes, Sam Blackwell, Sergio Boixo, Gina Bortoli, Alexandre Bourassa, Jenna Bovaird, Leon Brill, Michael Broughton, David A. Browne, Brett Buchea, Bob B. Buckley, David A. Buell, Tim Burger, Brian Burkett, Nicholas Bushnell, Anthony Cabrera, Juan Campero, Hung-Shen Chang, Yu Chen, Zijun Chen, Ben Chiaro, Desmond Chik, Charina Chou, Jahan Claes, Agnetta Y. Cleland, Josh Cogan, Roberto Collins, Paul Con-ner, William Courtney, Alexander L. Crook, Ben Curtin, Sayan Das, Alex Davies, Laura De Lorenzo, Dripto M. Debroy, Sean Demura, Michel Devoret, Agustin Di Paolo, Paul Donohoe, Ilya Drozdov, Andrew Dunsworth, Clint Earle, Thomas Edlich, Alec Eickbusch, Aviv Moshe Elbag, Mahmoud Elzouka, Catherine Erick-son, Lara Faoro, Edward Farhi, Vinicius S. Ferreira, Leslie Flores Burgos, Ebrahim Forati, Austin G. Fowler, Brooks Foxen, Suhas Ganjam, Gonzalo Garcia, Robert Gasca, Élie Genois, William Giang, Craig Gidney, Dar Gilboa, Raja Gosula, Ale-jandro Grajales Dau, Dietrich Graumann, Alex Greene, Jonathan A. Gross, Steve Habegger, John Hall, Michael C. Hamilton, Monica Hansen, Matthew P. Harri-gan, Sean D. Harrington, Francisco J. H. Heras, Stephen Heslin, Paula Heu, Os-car Higgott, Gordon Hill, Jeremy Hilton, George Holland, Sabrina Hong, Hsin-Yuan Huang, Ashley Huff, William J. Huggins, Lev B. Ioffe, Sergei V. Isakov, Justin Ive-land, Evan Jeffrey, Zhang Jiang, Cody Jones, Stephen Jordan, Chaitali Joshi, Pavol Juhas, Dvir Kafri, Hui Kang, Amir H. Karamlou, Kostyantyn Kechedzhi, Julian

Kelly, Trupti Khaire, Tanuj Khattar, Mostafa Khezri, Seon Kim, Paul V. Klimov, Andrey R. Klots, Bryce Kobrin, Pushmeet Kohli, Alexander N. Korotkov, Fedor Kostritsa, Robin Kothari, Borislav Kozlovskii, John Mark Kreikebaum, Vladislav D. Kurilovich, Nathan Lacroix, David Landhuis, Tiano Lange-Dei, Brandon W. Langley, Pavel Laptev, Kim-Ming Lau, Loïck Le Guevel, Justin Ledford, Joonho Lee, Kenny Lee, Yuri D. Lensky, Shannon Leon, Brian J. Lester, Wing Yan Li, Yin Li, Alexander T. Lill, Wayne Liu, William P. Livingston, Aditya Locharla, Erik Lucero, Daniel Lundahl, Aaron Lunt, Sid Madhuk, Fionn D. Malone, Ashley Maloney, Salvatore Mandrà, James Manyika, Leigh S. Martin, Orion Martin, Steven Martin, Cameron Maxfield, Jarrod R. McClean, Matt McEwen, Seneca Meeks, Anthony Megrant, Xiao Mi, Kevin C. Miao, Amanda Mieszala, Reza Molavi, Sebastian Molina, Shirin Montazeri, Alexis Morvan, Ramis Movassagh, Wojciech Mruczkiewicz, Ofer Naaman, Matthew Neeley, Charles Neill, Ani Nersisyan, Hartmut Neven, Michael Newman, Jiun How Ng, Anthony Nguyen, Murray Nguyen, Chia-Hung Ni, Murphy Yuezhen Niu, Thomas E. O'Brien, William D. Oliver, Alex Opremcak, Kristoffer Ottosson, Andre Petukhov, Alex Pizzuto, John Platt, Rebecca Potter, Orion Pritchard, Leonid P. Pryadko, Chris Quintana, Ganesh Ramachandran, Matthew J. Reagor, John Redding, David M. Rhodes, Gabrielle Roberts, Elliott Rosenberg, Emma Rosenfeld, Pedram Roushan, Nicholas C. Rubin, Negar Saei, Daniel Sank, Kannan Sankaragomathi, Kevin J. Satzinger, Henry F. Schurkus, Christopher Schuster, Andrew W. Senior, Michael J. Shearn, Aaron Shorter, Noah Shutt, Vladimir Shvarts, Shraddha Singh, Volodymyr Sivak, Jindra Skruzny, Spencer Small, Vadim Smelyanskiy, W. Clarke Smith, Rolando D. Somma, Sofia Springer, George Sterling, Doug Strain, Jordan Suchard, Aaron Szasz, Alex Sztein, Douglas Thor, Alfredo Torres, M. Mert Torunbalci, Abeer Vaishnav, Justin Vargas, Sergey Vdovichev, Guifre Vidal, Benjamin Villalonga, Catherine Vollgraff Heidweiller, Steven Waltman, Shannon X. Wang, Brayden Ware, Kate Weber, Travis Weidel, Theodore White, Kristi Wong, Bryan W. K. Woo, Cheng Xing, Z. Jamie Yao, Ping Yeh, Bicheng Ying, Juhwan Yoo, Noureldin Yosri, Grayson Young, Adam Zalcman, Yaxing Zhang, Ningfeng Zhu, Nicholas Zobrist, and Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. *Nature*, 638(8052):920–926, February 2025.

- [4] Anjali A. Agrawal, Joshua Job, Tyler L. Wilson, S. N. Saadatmand, Mark J. Hodson, Josh Y. Mutus, Athena Caesura, Peter D. Johnson, Justin E. Elenewski, Kaitlyn J. Morrell, and Alexander F. Kemper. Quantifying fault tolerant simulation of strongly correlated systems using the Fermi-Hubbard model. 2024.

- [5] Manuel G. Algaba, P. V. Sriluckshmy, Martin Leib, and Fedor Šimkovic IV. Low-depth simulations of fermionic systems on square-grid quantum hardware. *Quantum*, 8:1327, April 2024.
- [6] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. Quantum algorithms for quantum chemistry and quantum materials science. *Chem. Rev.*, 120(22):12685–12717, November 2020.
- [7] Johannes Bausch, Toby Cubitt, Charles Derby, and Joel Klassen. Mitigating Errors in Local Fermionic Encodings, June 2020.
- [8] Andrea Beggi, Ilaria Siloi, Claudia Benedetti, Enrico Piccinini, Luca Razzoli, Paolo Bordone, and Matteo G A Paris. Back and forth from Fock space to Hilbert space: a guide for commuters. *European Journal of Physics*, 39(6):065401, August 2018.
- [9] A. Bermudez, X. Xu, M. Gutiérrez, S. C. Benjamin, and M. Müller. Fault-tolerant protection of near-term trapped-ion topological qubits under realistic noise sources. *Phys. Rev. A*, 100(6):062307, December 2019.
- [10] Dhruv Bhatnagar, Matthew Steinberg, David Elkouss, Carmen G. Almudever, and Sebastian Feld. Low-Depth Flag-Style Syndrome Extraction for Small Quantum Error-Correction Codes, May 2023.
- [11] Dolev Bluvstein, Simon J. Evered, Alexandra A. Geim, Sophie H. Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, J. Pablo Bonilla Ataides, Nishad Maskara, Iris Cong, Xun Gao, Pedro Sales Rodriguez, Thomas Karolyshyn, Giulia Semeghini, Michael J. Gullans, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. Logical quantum processor based on reconfigurable atom arrays. *Nature*, 626(7997):58–65, February 2024.
- [12] H. Bombín and M. A. Martin-Delgado. Topological Quantum Distillation. *Phys. Rev. Lett.*, 97(18):180501, October 2006.
- [13] Héctor Bombín, Chris Dawson, Ryan V. Mishmash, Naomi Nickerson, Fernando Pastawski, and Sam Roberts. Logical blocks for fault-tolerant topological quantum computation. *PRX Quantum*, 4:020303, April 2023.
- [14] Sergey B. Bravyi and Alexei Yu. Kitaev. Fermionic Quantum Computation. *Annals of Physics*, 298(1):210–226, May 2002.

- [15] Nikolas P. Breuckmann and Vivien Londe. Single-shot decoding of linear rate LDPC quantum codes with high performance. *IEEE Transactions on Information Theory*, 68(1):272–286, 2022.
- [16] Chris Cade, Lana Mineh, Ashley Montanaro, and Stasja Stanisic. Strategies for solving the Fermi-Hubbard model on near-term quantum computers. *Phys. Rev. B*, 102:235122, December 2020.
- [17] Zhenyu Cai. Quantum Error Mitigation using Symmetry Expansion. *Quantum*, 5:548, September 2021.
- [18] Madelyn Cain, Chen Zhao, Hengyun Zhou, Nadine Meister, J Ataide, Arthur Jaffe, Dolev Bluvstein, and Mikhail D Lukin. Correlated decoding of logical algorithms with transversal gates. 2024.
- [19] Yudong Cao, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P. D. Sawaya, Sukin Sim, Libor Veis, and Alán Aspuru-Guzik. Quantum chemistry in the age of quantum computing. *Chemical Reviews*, 119(19):10856–10915, October 2019.
- [20] Rui Chao and Ben W. Reichardt. Quantum Error Correction with Only Two Extra Qubits. *Phys. Rev. Lett.*, 121(5):050502, August 2018.
- [21] Yu-An Chen and Yijia Xu. Equivalence between fermion-to-qubit mappings in two spatial dimensions. *PRX Quantum*, 4(1):010326, March 2023.
- [22] Riley W. Chien, Kanav Setia, Xavier Bonet-Monroig, Mark Steudtner, and James D. Whitfield. Simulating quantum error mitigation in fermionic encodings, May 2023.
- [23] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of Trotter error with commutator scaling. *Phys. Rev. X*, 11:011020, February 2021.
- [24] Luca Chirolli and Guido Burkard. Decoherence in solid-state qubits. *Advances in Physics*, 57(3):225–285, May 2008.
- [25] Laura Clinton, Johannes Bausch, and Toby Cubitt. Hamiltonian simulation algorithms for near-term quantum hardware. *Nat Commun*, 12(1):4989, August 2021.

- [26] Laura Clinton, Toby Cubitt, Brian Flynn, Filippo Maria Gambetta, Joel Klassen, Ashley Montanaro, Stephen Piddock, Raul A. Santos, and Evan Sheridan. Towards near-term quantum simulation of materials. *Nat Commun*, 15(1):211, January 2024.
- [27] Lawrence Z. Cohen, Isaac H. Kim, Stephen D. Bartlett, and Benjamin J. Brown. Low-overhead fault-tolerant quantum computing using long-range connectivity. *Science Advances*, 8(20):eabn1717, 2022.
- [28] Alexander M. Dalzell, Sam McArdle, Mario Berta, Przemyslaw Bienias, Chi-Fang Chen, András Gilyén, Connor T. Hann, Michael J. Kastoryano, Emil T. Khabiboulline, Aleksander Kubica, Grant Salton, Samson Wang, and Fernando G. S. L. Brandão. Quantum algorithms: A survey of applications and end-to-end complexities, October 2023.
- [29] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, September 2002.
- [30] Charles Derby and Joel Klassen. A Compact Fermion to Qubit Mapping Part 2: Alternative Lattice Geometries, January 2021.
- [31] Charles Derby, Joel Klassen, Johannes Bausch, and Toby Cubitt. Compact fermion to qubit mappings. *Phys. Rev. B*, 104(3):035118, July 2021.
- [32] Cirq Developers. Cirq. <https://zenodo.org/doi/10.5281/zenodo.4062499>, July 2025.
- [33] Zhiyan Ding, Yulong Dong, Yu Tong, and Lin Lin. Robust ground-state energy estimation under depolarizing noise, March 2024.
- [34] Philip M. Dixon. *Bootstrap Resampling*. John Wiley & Sons, Ltd, 2006.
- [35] Emiliia Dyrenkova, Raymond Laflamme, and Michael Vasmer. Scalable simulation of fermionic encoding performance on noisy quantum computers, 2025.
- [36] Samuele Ferracin, Akel Hashim, Jean-Loup Ville, Ravi Naik, Arnaud Carignan-Dugas, Hammam Qassim, Alexis Morvan, David I. Santiago, Irfan Siddiqi, and Joel J. Wallman. Efficiently improving the performance of noisy quantum computers. *Quantum*, 8:1410, July 2024.
- [37] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, 86:032324, September 2012.

- [38] Neelay Fruitwala, Akel Hashim, Abhi D Rajagopala, Yilun Xu, Jordan Hines, Ravi K Naik, Irfan Siddiqi, Katherine Klymko, Gang Huang, and Kasra Nowrouzi. Hardware-efficient randomized compiling. 2024.
- [39] György P. Gehér, Campbell McLauchlan, Earl T. Campbell, Alexandra E. Moylett, and Ophelia Crawford. Error-corrected Hadamard gate simulated at the circuit level. *Quantum*, 8:1394, July 2024.
- [40] Craig Gidney. Stim: A fast stabilizer circuit simulator. *Quantum*, 5:497, July 2021.
- [41] Craig Gidney, Michael Newman, and Matt McEwen. Benchmarking the Planar Honeycomb Code. *Quantum*, 6:813, September 2022.
- [42] Google AI Quantum and Collaborators, Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Sergio Boixo, Michael Broughton, Bob B. Buckley, David A. Buell, Brian Burkett, Nicholas Bushnell, Yu Chen, Zijun Chen, Benjamin Chiaro, Roberto Collins, William Courtney, Sean Demura, Andrew Dunsworth, Edward Farhi, Austin Fowler, Brooks Foxen, Craig Gidney, Marissa Giustina, Rob Graff, Steve Habegger, Matthew P. Harrigan, Alan Ho, Sabrina Hong, Trent Huang, William J. Huggins, Lev Ioffe, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Cody Jones, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Seon Kim, Paul V. Klimov, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Pavel Laptev, Mike Lindmark, Erik Lucero, Orion Martin, John M. Martinis, Jarrod R. McClean, Matt McEwen, Anthony Megrant, Xiao Mi, Masoud Mohseni, Wojciech Mruczkiewicz, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Hartmut Neven, Murphy Yuezhen Niu, Thomas E. O’Brien, Eric Ostby, Andre Petukhov, Harald Putterman, Chris Quintana, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Doug Strain, Kevin J. Sung, Marco Szalay, Tyler Y. Takeshita, Amit Vainsencher, Theodore White, Nathan Wiebe, Z. Jamie Yao, Ping Yeh, and Adam Zalcman. Hartree-Fock on a superconducting qubit quantum computer. *Science*, 369(6507):1084–1089, August 2020.
- [43] Daniel Gottesman. *Surviving as a Quantum Computer in a Classical World*. 2024 Textbook Draft.
- [44] Daniel Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, 1997.
- [45] Daniel Gottesman. The Heisenberg Representation of Quantum Computers, July 1998.

- [46] Antoine Grospellier, Lucien Grouès, Anirudh Krishna, and Anthony Leverrier. Combining hard and soft decoders for hypergraph product codes. *Quantum*, 5:432, April 2021.
- [47] Lucas Hackl, Dayang Li, Nika Akopian, and Matthias Christandl. Experimental proposal to probe the extended pauli principle. *Phys. Rev. A*, 108:012208, July 2023.
- [48] Tobias Hagge. Optimal fermionic SWAP networks for Hubbard models, August 2022.
- [49] Tobias Hagge and Nathan Wiebe. Error mitigation via error detection using Generalized Superfast Encodings, September 2023.
- [50] Brent Harrison, Mitchell Chiew, Jason Necaise, Andrew Projansky, Sergii Strelchuk, and James D. Whitfield. A Sierpinski triangle fermion-to-qubit transform, 2024.
- [51] Akel Hashim, Ravi K. Naik, Alexis Morvan, Jean-Loup Ville, Bradley Mitchell, John Mark Kreikebaum, Marc Davis, Ethan Smith, Costin Iancu, Kevin P. O’Brien, Ian Hincks, Joel J. Wallman, Joseph Emerson, and Irfan Siddiqi. Randomized compiling for scalable quantum computing on a noisy superconducting quantum processor. *Phys. Rev. X*, 11:041039, Nov 2021.
- [52] Vojtěch Havlíček, Matthias Troyer, and James D. Whitfield. Operator locality in the quantum simulation of fermionic models. *Phys. Rev. A*, 95(3):032332, March 2017.
- [53] Kévin Hémerly, Khaldoun Ghanem, Eleanor Crane, Sara L. Campbell, Joan M. Dreiling, Caroline Figgatt, Cameron Foltz, John P. Gaebler, Jacob Johansen, Michael Mills, Steven A. Moses, Juan M. Pino, Anthony Ransford, Mary Rowe, Peter Siegfried, Russell P. Stutz, Henrik Dreyer, Alexander Schuckert, and Ramil Nigmatullin. Measuring the Loschmidt amplitude for finite-energy properties of the Fermi-Hubbard model on an ion-trap quantum computer, September 2023.
- [54] Cornelius Hempel, Christine Maier, Jonathan Romero, Jarrod McClean, Thomas Monz, Heng Shen, Petar Jurcevic, Ben P. Lanyon, Peter Love, Ryan Babbush, Alán Aspuru-Guzik, Rainer Blatt, and Christian F. Roos. Quantum Chemistry Calculations on a Trapped-Ion Quantum Simulator. *Phys. Rev. X*, 8(3):031022, July 2018.
- [55] Oscar Higgott, Matthew Wilson, James Hefford, James Dborin, Farhan Hanif, Simon Burton, and Dan E. Browne. Optimal local unitary encoding circuits for the surface code. *Quantum*, 5:517, August 2021.

- [56] Dominic Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter. Surface code quantum computing by lattice surgery. *New Journal of Physics*, 14(12):123011, dec 2012.
- [57] Eric Huang, Arthur Pesah, Christopher T. Chubb, Michael Vasmer, and Arpit Dua. Tailoring Three-Dimensional Topological Codes for Biased Noise. *PRX Quantum*, 4(3):030338, September 2023.
- [58] John Hubbard. Electron correlations in narrow energy bands. *Proc. R. Soc. Lond. A*, 276(1365):238–257, 1963.
- [59] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with Qiskit, 2024.
- [60] Zhang Jiang, Amir Kalev, Wojciech Mruczkiewicz, and Hartmut Neven. Optimal fermion-to-qubit mapping via ternary trees with applications to reduced quantum states learning. *Quantum*, 4:276, June 2020.
- [61] Zhang Jiang, Jarrod McClean, Ryan Babbush, and Hartmut Neven. Majorana Loop Stabilizer Codes for Error Mitigation in Fermionic Quantum Simulations. *Phys. Rev. Applied*, 12(6):064041, December 2019.
- [62] Zhang Jiang, Kevin J. Sung, Kostyantyn Kechedzhi, Vadim N. Smelyanskiy, and Sergio Boixo. Quantum algorithms to simulate many-body physics of correlated fermions. *Phys. Rev. Appl.*, 9:044036, April 2018.
- [63] P. Jordan and E. Wigner. Über das Paulische Äquivalenzverbot. *Z. Physik*, 47(9-10):631–651, September 1928.
- [64] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout van den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, and et al. Evidence for the utility of quantum computing before fault tolerance. *Nature News*, June 2023.
- [65] A. Yu Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, January 2003.
- [66] Ian D. Kivlichan, Jarrod McClean, Nathan Wiebe, Craig Gidney, Alán Aspuru-Guzik, Garnet Kin-Lic Chan, and Ryan Babbush. Quantum simulation of electronic

structure with linear depth and connectivity. *Phys. Rev. Lett.*, 120:110501, March 2018.

- [67] Alexander A. Klyachko. The Pauli exclusion principle and beyond, 2009.
- [68] Aleksander Kubica, Beni Yoshida, and Fernando Pastawski. Unfolding the color code. *New Journal of Physics*, 17(8):083026, August 2015.
- [69] J. P. F. LeBlanc, Andrey E. Antipov, Federico Becca, Ireneusz W. Bulik, Garnet Kin-Lic Chan, Chia-Min Chung, Youjin Deng, Michel Ferrero, Thomas M. Henderson, Carlos A. Jiménez-Hoyos, E. Kozik, Xuan-Wen Liu, Andrew J. Millis, N. V. Prokof'ev, Mingpu Qin, Gustavo E. Scuseria, Hao Shi, B. V. Svistunov, Luca F. Tocchio, I. S. Tupitsyn, Steven R. White, Shiwei Zhang, Bo-Xiao Zheng, Zhenyue Zhu, and Emanuel Gull. Solutions of the two-dimensional Hubbard model: Benchmarks and results from a wide range of numerical algorithms. *Phys. Rev. X*, 5, Dec 2015.
- [70] Lin Lin. Lecture notes on quantum algorithms for scientific computation, 2022.
- [71] Daniel Litinski. A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery. *Quantum*, 3:128, March 2019.
- [72] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.
- [73] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. Quantum computational chemistry. *Rev. Mod. Phys.*, 92:015003, March 2020.
- [74] Jarrod R. McClean, Zhang Jiang, Nicholas C. Rubin, Ryan Babbush, and Hartmut Neven. Decoding quantum errors with subspace expansions. *Nat Commun*, 11(1):636, January 2020.
- [75] Aaron Miller, Adam Glos, and Zoltán Zimborás. Treespilation: Architecture- and State-Optimised Fermion-to-Qubit Mappings, March 2024.
- [76] Aaron Miller, Zoltán Zimborás, Stefan Knecht, Sabrina Maniscalco, and Guillermo García-Pérez. Bonsai Algorithm: Grow Your Own Fermion-to-Qubit Mappings. *PRX Quantum*, 4(3):030314, August 2023.
- [77] Mario Motta and Julia E. Rice. Emerging quantum computing algorithms for quantum chemistry. *WIREs Computational Molecular Science*, 12(3):e1580, 2022.

- [78] F. Motzoi, J. M. Gambetta, P. Rebentrost, and F. K. Wilhelm. Simple Pulses for Elimination of Leakage in Weakly Nonlinear Qubits. *Phys. Rev. Lett.*, 103(11):110501, September 2009.
- [79] Maryam Mudassar, Riley W. Chien, and Daniel Gottesman. Encoding majorana codes. *Physical Review A*, 110(3), September 2024.
- [80] C. J. Myatt, B. E. King, Q. A. Turchette, C. A. Sackett, D. Kielpinski, W. M. Itano, C. Monroe, and D. J. Wineland. Decoherence of quantum superpositions through coupling to engineered reservoirs. *Nature*, 403(6767):269–273, January 2000.
- [81] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [82] Ramil Nigmatullin, Kevin Hemery, Khaldoon Ghanem, Steven Moses, Dan Gresh, Peter Siegfried, Michael Mills, Thomas Gatterman, Nathan Hewitt, Etienne Granet, and Henrik Dreyer. Experimental demonstration of break-even for the Compact fermionic encoding. 2024.
- [83] P. J. J. O’Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik, and J. M. Martinis. Scalable Quantum Simulation of Molecular Energies. *Phys. Rev. X*, 6(3):031007, July 2016.
- [84] Miha Papič, Manuel G. Algaba, Emiliano Godinez-Ramirez, Inés de Vega, Adrian Auer, Fedor Šimkovic IV, and Alessio Calzona. Near-term fermionic simulation with subspace noise tailored quantum error mitigation, 2025.
- [85] Maxwell F. Parsons, Florian Huber, Anton Mazurenko, Christie S. Chiu, Widagdo Setiawan, Katherine Wooley-Brown, Sebastian Blatt, and Markus Greiner. Site-resolved imaging of fermionic ${}^6\text{Li}$ in an optical lattice. *Phys. Rev. Lett.*, 114:213002, May 2015.
- [86] Kaavya Sahay, Yingjia Lin, Shilin Huang, Kenneth R Brown, and Shruti Puri. Error correction of transversal CNOT gates for scalable surface code computation. 2024.
- [87] Hasan Sayginel, Francois Jamet, Abhishek Agarwal, Dan E. Browne, and Ivan Rungger. A fault-tolerant variational quantum algorithm with limited T-depth, March 2023.

- [88] Ansgar Schubert and Christian B. Mendl. Trotter error with commutator scaling for the Fermi-Hubbard model. *Physical Review B*, 108(19), November 2023.
- [89] Thomas R. Scruby, Michael Vasmer, and Dan E. Browne. Non-Pauli errors in the three-dimensional surface code. *Phys. Rev. Res.*, 4:043052, October 2022.
- [90] Jacob T. Seeley, Martin J. Richard, and Peter J. Love. The Bravyi-Kitaev transformation for quantum computation of electronic structure. *The Journal of Chemical Physics*, 137(22):224109, December 2012.
- [91] Kanav Setia, Sergey Bravyi, Antonio Mezzacapo, and James D. Whitfield. Superfast encodings for fermionic quantum simulation. *Phys. Rev. Research*, 1(3):033033, October 2019.
- [92] Fedor Simkovic IV, Martin Leib, and Francisco Revson F. Pereira. Low-Weight High-Distance Error Correcting Fermionic Encodings, February 2024.
- [93] Stasja Stanisic, Jan Lukas Bosse, Filippo Maria Gambetta, Raul A. Santos, Wojciech Mruczkiewicz, Thomas E. O’Brien, Eric Ostby, and Ashley Montanaro. Observing ground-state properties of the Fermi-Hubbard model using a scalable algorithm on a quantum computer. *Nat Commun*, 13(1):5743, October 2022.
- [94] Armands Strikis, Dayue Qin, Yanzhu Chen, Simon C. Benjamin, and Ying Li. Learning-based quantum error mitigation. *PRX Quantum*, 2(4):040330, November 2021.
- [95] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, and Jonathan Tennyson. The Variational Quantum Eigensolver: A review of methods and best practices. *Physics Reports*, 986:1–128, November 2022.
- [96] Kento Tsubouchi, Yasunari Suzuki, Yuuki Tokunaga, Nobuyuki Yoshioka, and Suguru Endo. Virtual quantum error detection. *Phys. Rev. A*, 108(4):042426, October 2023.
- [97] Q. A. Turchette, C. J. Myatt, B. E. King, C. A. Sackett, D. Kielpinski, W. M. Itano, C. Monroe, and D. J. Wineland. Decoherence and decay of motional quantum states of a trapped atom coupled to engineered reservoirs. *Phys. Rev. A*, 62(5):053807, October 2000.

- [98] Ewout Van Den Berg, Zlatko K. Mineev, Abhinav Kandala, and Kristan Temme. Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors. *Nat. Phys.*, 19(8):1116–1121, August 2023.
- [99] Michael Vasmer, Dan E. Browne, and Aleksander Kubica. Cellular automaton decoders for topological quantum codes with noisy measurements and beyond. *Scientific Reports*, 11(1):2027, December 2021.
- [100] F Verstraete and J I Cirac. Mapping local Hamiltonians of fermions to local Hamiltonians of spins. *J. Stat. Mech.*, 2005(09):P09012–P09012, September 2005.
- [101] Joel J. Wallman and Joseph Emerson. Noise tailoring for scalable quantum computation via randomized compiling. *Phys. Rev. A*, 94(5):052325, November 2016.
- [102] Kwok Ho Wan, Mark Webber, Austin G Fowler, and Winfried K Hensinger. An iterative transversal cnot decoder. 2024.
- [103] Christopher J. Wood and Jay M. Gambetta. Quantification and characterization of leakage errors. *Phys. Rev. A*, 97(3):032306, March 2018.
- [104] Daniel Zhang and Toby Cubitt. Quantum Error Transmutation, October 2023.

Appendices

Appendix A

Depth tables

We provide one-qubit (1Q) and two-qubit (2Q) gate counts in circuits that were used in our plots in Tables A.1 to A.5. This provides a rough prediction for how many noisy gates a simulation with different encodings can tolerate according to our results. In some sense, this is an upper bound on the number of gates that can be performed in similar implementations; if the Clifford simulation shows failure to extract information after a certain number of gates, then it is unlikely that the same circuit would perform better in a real device.

Fraction of Hamiltonian terms	Lattice size 4×4			Lattice size 8×8			Lattice size 12×12			Lattice size 16×16		
	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total
0.2	96	192	288	296	622	918	664	1450	2114	1226	2698	3924
0.6	218	390	608	710	1420	2130	1792	3762	5554	3154	6542	9696
1.0	334	624	958	1216	2330	3546	2510	5304	7814	4562	9392	13954
1.6	430	782	1212	1596	3100	4696	3396	6724	10120	5962	12084	18046
2.0	466	830	1296	1604	3180	4784	3566	7096	10662	6310	12592	18902

Table A.1: DK Encoding, occupation number measurement, stabilizer reconstruction error detection, random logical circuits.

Fraction of Hamiltonian terms	Lattice size 4×4			Lattice size 8×8			Lattice size 12×12			Lattice size 16×16		
	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total
0.2	208	232	440	746	792	1538	1704	1972	3676	2868	3142	6010
0.6	330	446	776	1186	1658	2844	2752	4130	6882	4790	7002	11792
1.0	446	662	1108	1602	2528	4130	3634	5720	9354	6374	10154	16528
1.6	540	846	1386	2044	3292	5336	4414	7138	11552	7696	12674	20370
2.0	566	854	1420	2054	3358	5412	4652	7612	12264	8178	13406	21584

Table A.2: DK Encoding, hopping term measurement, stabilizer measurement error detection, random logical circuits.

Trotter steps	Lattice size 4×4			Lattice size 8×8			Lattice size 12×12			Lattice size 16×16			Lattice size 18×18		
	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total
2	242	385	627	778	1439	2217	1610	3089	4699	2730	5313	8043	3398	6641	10039
4	470	753	1223	1490	2787	4277	3064	5941	9005	5188	10181	15369	6450	12711	19161
6	698	1121	1819	2202	4135	6337	4518	8793	13311	7632	15029	22661	9498	18765	28263
8	926	1489	2415	2914	5483	8397	5972	11645	17617	10086	19887	29973	12546	24825	37371
10	1154	1857	3011	3626	6831	10457	7426	14497	21923	12530	24735	37265	15594	30879	46473
20	2294	3697	5991	7186	13571	20757	14696	28757	43453	24780	49005	73785	30834	61167	92001
40	4574	7377	11951	14306	27051	41357	29236	57277	86513	49270	97535	146805	61314	121737	183051

Table A.3: DK Encoding, occupation number measurement, stabilizer reconstruction and full Trotter circuits.

Trotter steps	Lattice size 4×4			Lattice size 8×8			Lattice size 12×12			Lattice size 16×16			Lattice size 18×18		
	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total
2	140	117	257	642	419	1061	1436	919	2355	2562	1611	4173	3272	2029	5301
4	282	241	523	1478	977	2455	3258	2275	5533	5780	4071	9851	7364	5155	12519
6	424	365	789	2318	1541	3859	5086	3639	8725	9006	6545	15551	11464	8299	19763
8	566	489	1055	3158	2105	5263	6914	5003	11917	12232	9019	21251	15564	11443	27007
10	708	613	1321	3998	2669	6667	8742	6367	15109	15458	11493	26951	19664	14587	34251
20	1418	1233	2651	8198	5489	13687	17882	13187	31069	31588	23863	55451	40164	30307	70471
40	2838	2473	5311	16598	11129	27727	36162	26827	62989	63848	48603	112451	81164	61747	142911

Table A.4: JW Encoding, occupation number measurement, global parity postselection, full Trotter circuits.

Trotter steps	Lattice size 4×4			Lattice size 8×8			Lattice size 12×12			Lattice size 16×16			Lattice size 18×18		
	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total	2Q	1Q	Total
2	872	664	1536	4658	3740	8398	12060	9152	21212	23210	17676	40886	31158	23656	54814
4	1704	1384	3088	8510	7450	15960	22396	18314	40710	41376	35300	76676	53798	46970	100768
6	2536	2072	4608	12294	11194	23488	32726	27468	60194	59670	53164	112834	76672	70856	147528
8	3396	2728	6124	16146	14854	31000	42846	36480	79326	77848	70890	148738	99166	93942	193108
10	4240	3410	7650	19914	18688	38602	53244	45726	98970	95898	87952	183850	121712	117366	239078
20	8484	6654	15138	38652	37450	76102	104464	91002	195466	186002	177196	363198	234404	234538	468942
40	16944	13314	30258	76432	74850	151282	207134	182432	389566	369880	354218	724098	460258	470880	931138

Table A.5: TT Encoding, occupation number measurement, full Trotter circuits.

Appendix B

Additional Simulation Data

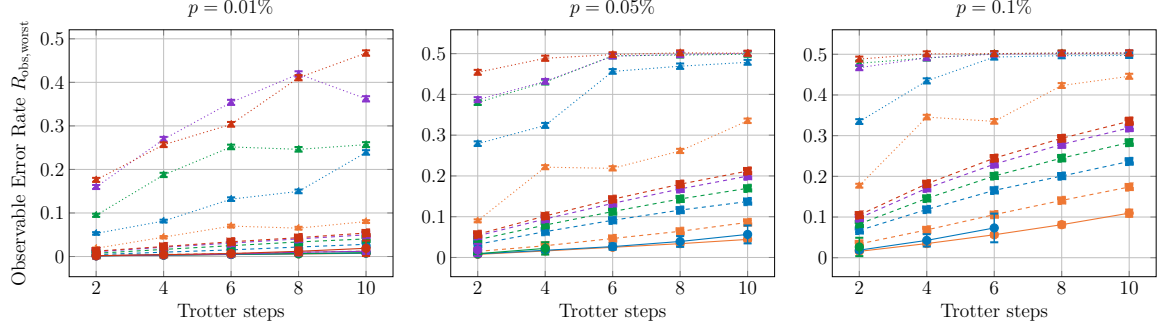
Superconducting inspired error model. Figs. B.1 and B.2 are the same as the figures as in main text but with the SI error model. This model, with its reduced single-qubit error rate, provides a slight advantage for DK encoding with 20 Trotter steps potentially accessible for a 12×12 lattice at the lowest physical error rate of $p = 0.01\%$ (central plot of Fig. B.1b). We see no significant differences in the performance of various error mitigation methods for the DK encoding in Fig. B.2. The performance of JW and TT encodings are almost identical to the SD error model case.

Hopping term measurements. Fig. B.4 provides a similar comparison of JW and DK to Figs. 4.4 and B.1 but with measurements extracting hopping term observables. For DK circuits we measure all hopping terms colored in red in Fig. 3.6 with stabilizer measurement for postselection (no flag qubits). For JW, we measure $X_i X_{i+1}$ observables, which is the most efficient (in terms of the number of Hamiltonian terms extracted) naive hopping term measurement. We do not include results for TT here because we do not have a comparable measurement strategy available for it. We observe that this kind of measurement significantly disadvantages the JW encoding (with fermionic swap networks) compared with the occupation number measurement. Hence, the difference in performance between DK and JW is much more pronounced, with the potential to extract relatively low-error observables using the DK encoding that are impossible using the JW encoding. However, the DK encoding suffers from high postselection costs in this setting as well, with most promising results being 10 Trotter steps for the 16×16 lattice and 8 Trotter steps for the 18×18 lattice.

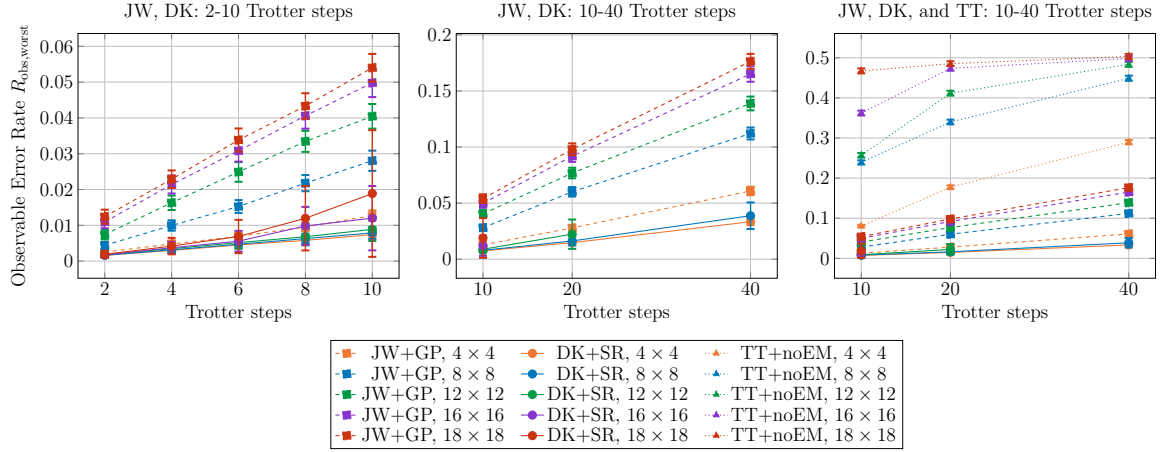
DK comparison for more error rates (both error models). Figs. 4.5 and B.2 compare the DK encoding in the intermediate error regime of $p = 0.05\%$ of SD and SI error

models respectively. As mentioned in the main text, we highlight this error regime as it best demonstrates the tradeoff between shot counts and accuracy in the DK encoding. Figs. B.5 to B.8 demonstrate the performance of the DK encoding in the aspirational ($p = 0.01\%$) and near future ($p = 0.1\%$) regimes for both the SD and SI error models. We again see no notable differences between SD and SI error models. Results in both regimes show little to no advantage from GP compared to no error mitigation, with some improvements present only in smaller lattices and low error rates. Meanwhile, when there are enough samples to discard in the aspirational regime, stabilizer reconstruction and stabilizer measurements provide an improvement in accuracy for all lattice sizes. However, we see that the sampling requirements are impractical for the near future error regime.

JW with GP vs. no GP. Figure B.9 demonstrates no significant improvement in accuracy resulting from GP in JW circuits except for the smallest lattice sizes and lowest error rates.

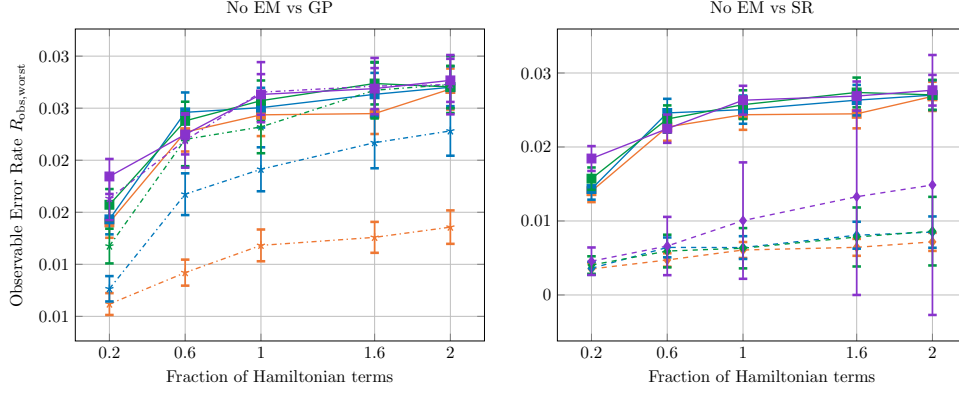


(a) Superconducting inspired error model at different error rates

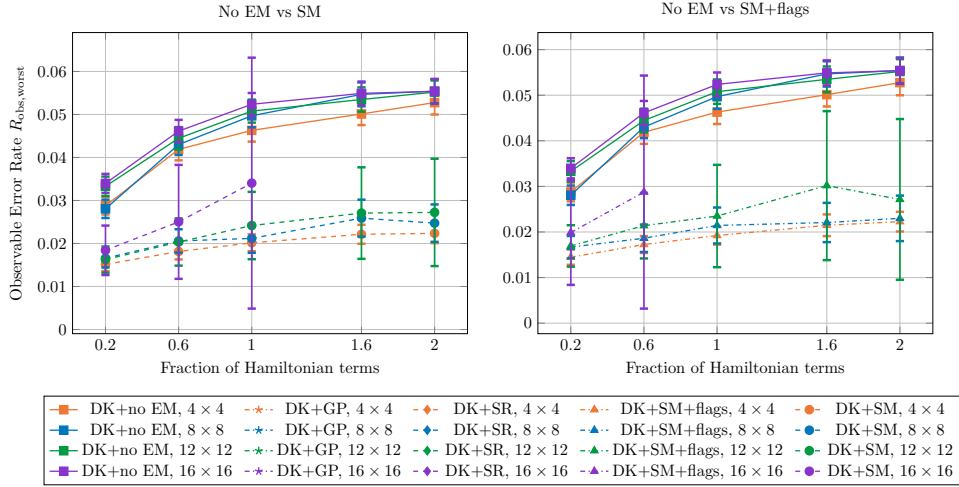


(b) Superconducting inspired error model at $p=0.01\%$

Figure B.1: Performance of fermionic encodings in the superconducting inspired error model. We consider the DK, JW, and TT encodings, with no error mitigation (no EM), stabilizer reconstruction (SR) and global parity postselection (GP). One Trotter step corresponds to an application of all the Hamiltonian terms, and we measure the occupation number at the end of the circuit. $R_{\text{obs,worst}}$ is the maximum local observable error rate and data points with error detection rate $R_{\text{det}} \geq 0.995$ are excluded. Error bars are computed using bootstrap resampling with 1000 resamples. (a) compares encoding performance across error rates. (b) zooms in on the lowest error rate, $p = 0.01\%$: the left plot magnifies DK and JW data from (a), the center compares deeper circuits for DK and JW, and the right includes all three encodings (note the y-axis scale change).



(a) Observable errors for occupation number measurements at $p = 0.05\%$ (superconducting inspired error model)



(b) Observable errors for hopping term measurements at $p = 0.05\%$ (superconducting inspired error model)

Figure B.2: Performance of various postselection methods in the DK encoding (SD error model). Each circuit applies a random sequence of logical operators, with operator count given as a fraction of Hamiltonian terms. Error bars are computed using bootstrap resampling with 1000 resamples. (a) and (b) showcase improvements in accuracy enabled by global parity postselection (GP) and stabilizer reconstruction (SR) for occupation number measurements, as well as stabilizer measurement (SM) and stabilizer measurement with flag qubits (SM+flags) for hopping term measurements, compared with no error mitigation (no EM). Data points with error detection rate $R_{\text{det}} \geq 0.995$ are excluded. The legend in (b) is shared by (a) and (b).

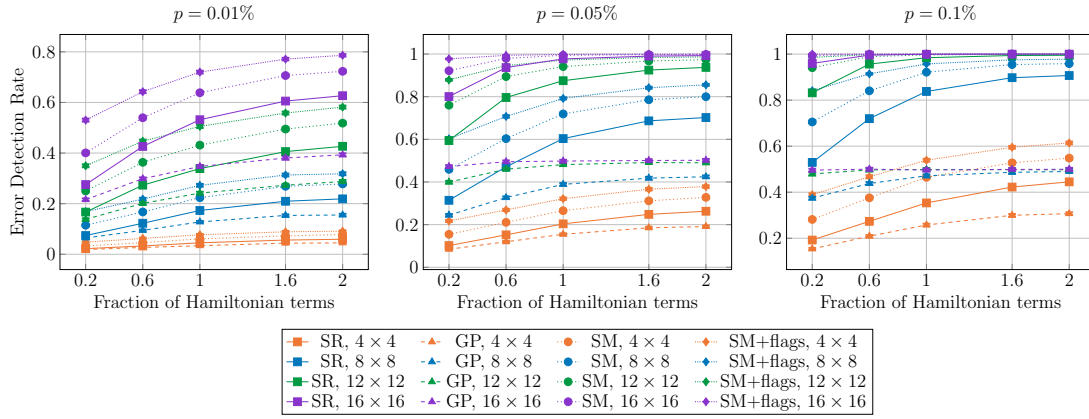
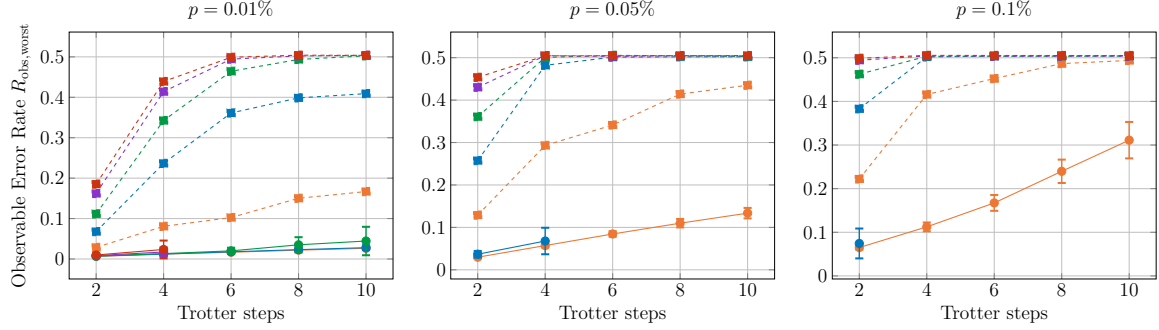
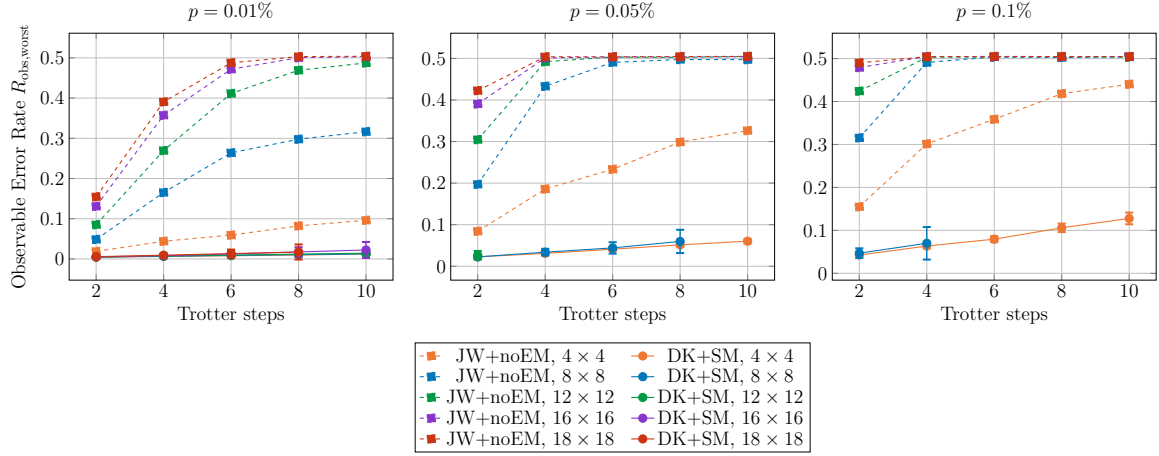


Figure B.3: Error detection rates at different physical error rates (superconducting inspired error model).

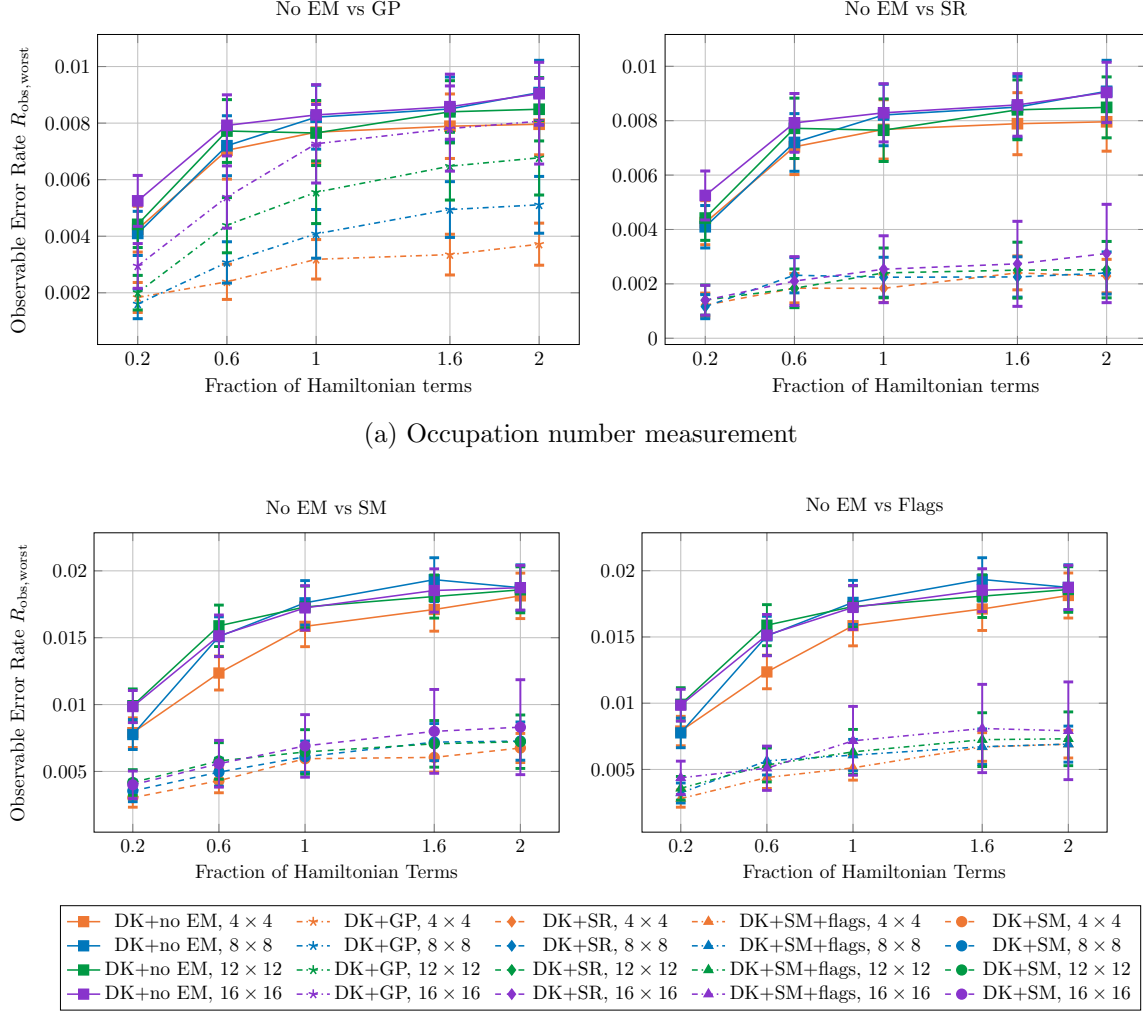


(a) Standard depolarizing error model at different error rates p



(b) Superconducting inspired error model at different error rates p

Figure B.4: **Comparison of hopping term measurement performance at different error rates.** (a) We consider the DK encoding with stabilizer measurement (DK+SM) and the JW encoding (with no error mitigation) on fermionic lattice sizes from 4×4 to 18×18 . One Trotter step corresponds to an application of all the terms in the Hamiltonian and the we measure in the hopping operator basis at the end of each circuit. $R_{\text{obs,worst}}$ is the maximum local observable error rate and data points with error detection rate $R_{\text{det}} \geq 0.995$ are excluded. Error bars are computed using bootstrap resampling with 1000 resamples. We simulate the standard depolarizing error model with strength $p = 0.01\%$ (left), $p = 0.05\%$ (middle), and $p = 0.1\%$ (right). (b) Identical to (a) except with the superconducting inspired error model.



(b) Hopping term measurement

Figure B.5: **Performance of the DK encoding in the aspirational error regime ($p = 0.01\%$) for the standard depolarizing error model.** We consider fermionic lattice sizes of 4×4 up to 16×16 . Each circuit applies a random sequence of logical operators, with operator count given as a fraction of Hamiltonian terms. $R_{\text{obs, worst}}$ is the maximum local observable error rate and data points with error detection rate $R_{\text{det}} \geq 0.995$ are excluded. Error bars are computed using bootstrap resampling with 1000 resamples. (a) Performance comparison of global parity postselection (GP) and stabilizer reconstruction (SR) with no error mitigation (no EM) for occupation number measurements. (b) Performance comparison of stabilizer measurement (SM) and flag qubits with no error mitigation for hopping term measurements.

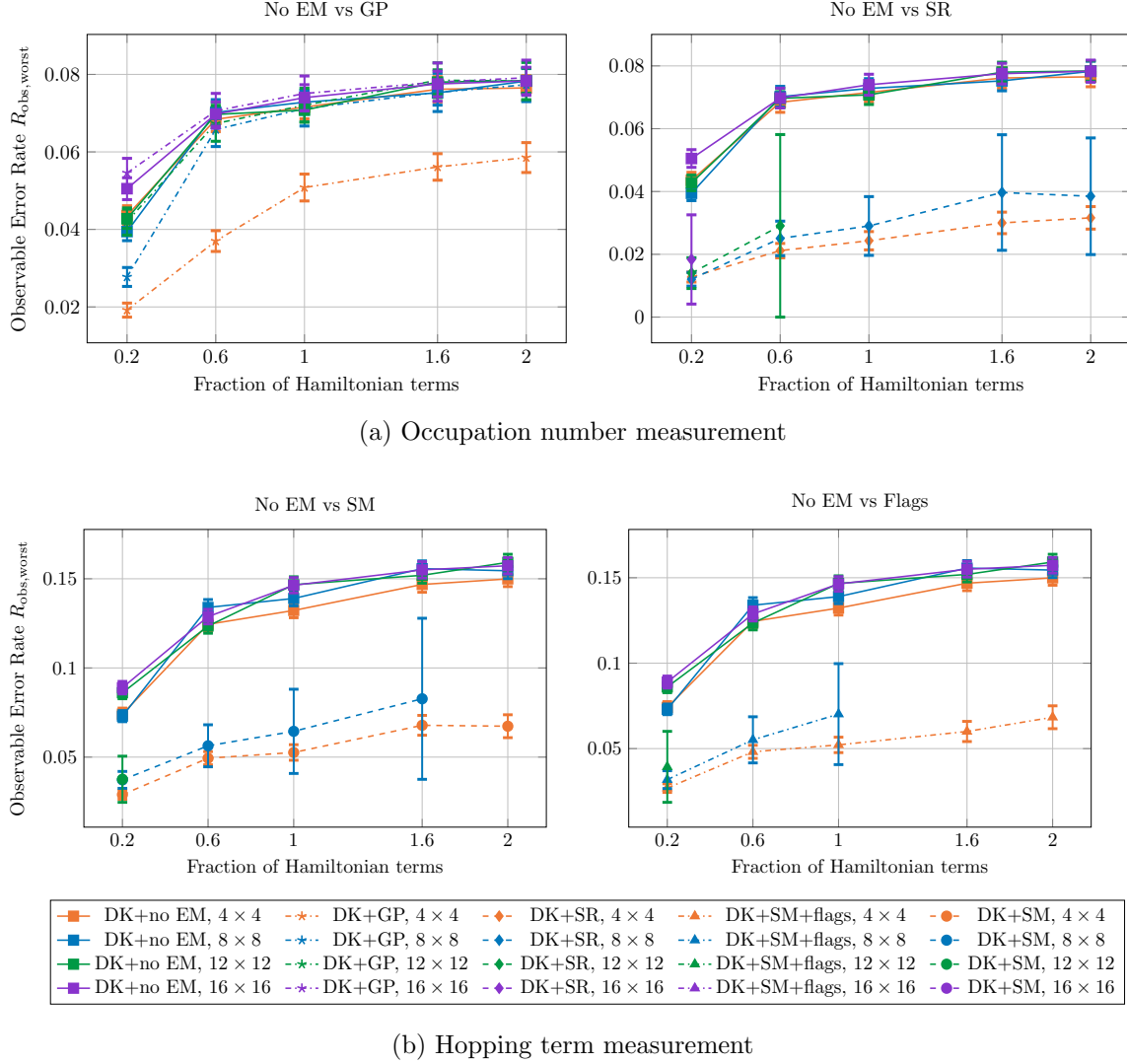


Figure B.6: **Performance of the DK encoding in the near future error regime ($p = 0.1\%$) for the standard depolarizing error model.** We consider fermionic lattice sizes of 4×4 up to 16×16 . Each circuit applies a random sequence of logical operators, with operator count given as a fraction of Hamiltonian terms. $R_{\text{obs,worst}}$ is the maximum local observable error rate and data points with error detection rate $R_{\text{det}} \geq 0.995$ are excluded. Error bars are computed using bootstrap resampling with 1000 resamples. (a) Performance comparison of global parity postselection (GP) and stabilizer reconstruction (SR) with no error mitigation (no EM) for occupation number measurements. (b) Performance comparison of stabilizer measurement (SM) and flag qubits with no error mitigation for hopping term measurements.

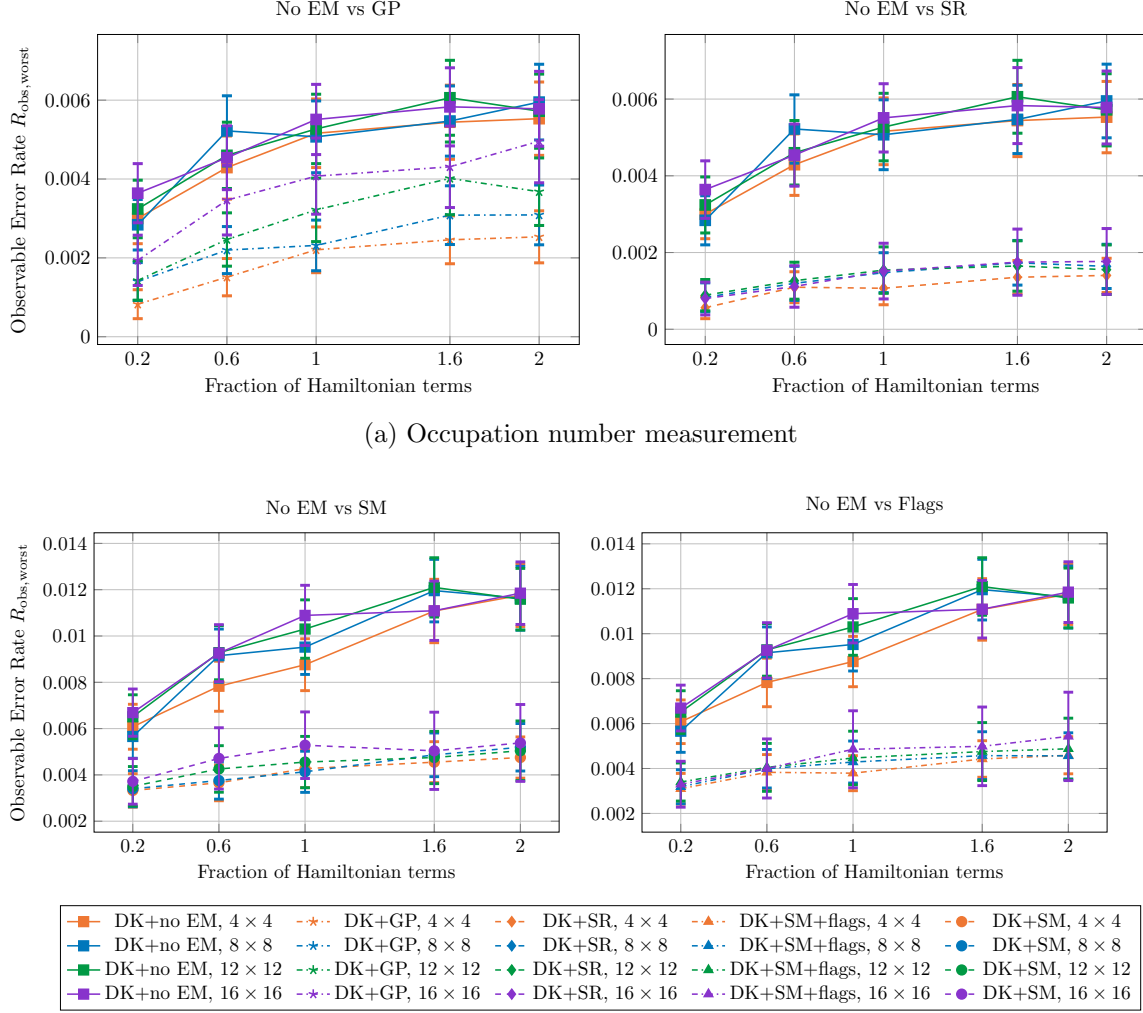


Figure B.7: **Performance of the DK encoding in the aspirational error regime ($p = 0.01\%$) for the superconducting inspired error model.** We consider fermionic lattice sizes of 4×4 up to 16×16 . Each circuit applies a random sequence of logical operators, with operator count given as a fraction of Hamiltonian terms. $R_{\text{obs, worst}}$ is the maximum local observable error rate and data points with error detection rate $R_{\text{det}} \geq 0.995$ are excluded. Error bars are computed using bootstrap resampling with 1000 resamples. (a) Performance comparison of global parity postselection (GP) and stabilizer reconstruction (SR) with no error mitigation (no EM) for occupation number measurements. (b) Performance comparison of stabilizer measurement (SM) and flag qubits with no error mitigation for hopping term measurements.

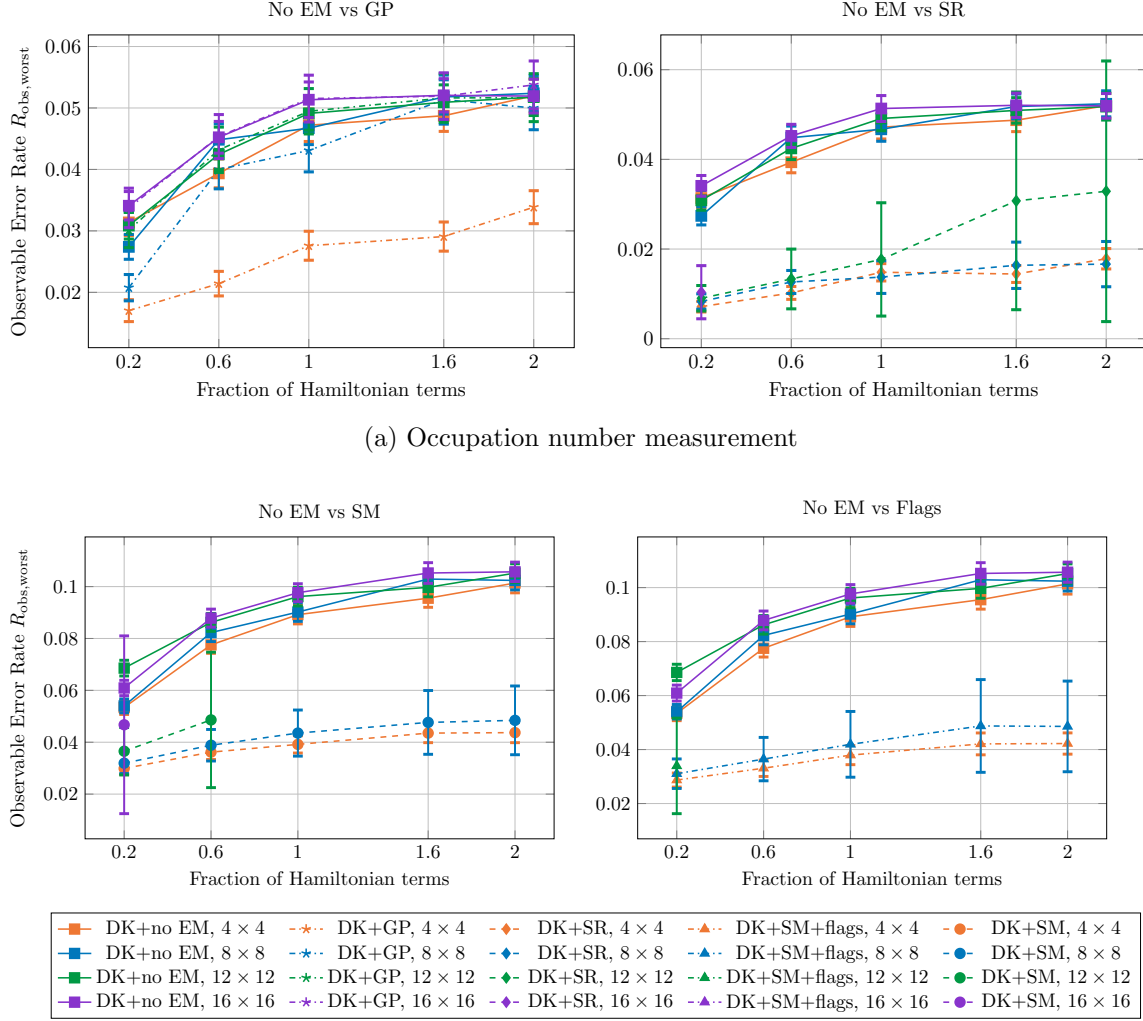


Figure B.8: **Performance of the DK encoding in the near future error regime ($p = 0.1\%$) for the superconducting inspired error model.** We consider fermionic lattice sizes of 4×4 up to 16×16 . Each circuit applies a random sequence of logical operators, with operator count given as a fraction of Hamiltonian terms. $R_{\text{obs, worst}}$ is the maximum local observable error rate and data points with error detection rate $R_{\text{det}} \geq 0.995$ are excluded. Error bars are computed using bootstrap resampling with 1000 resamples. (a) Performance comparison of global parity postselection (GP) and stabilizer reconstruction (SR) with no error mitigation (no EM) for occupation number measurements. (b) Performance comparison of stabilizer measurement (SM) and flag qubits with no error mitigation for hopping term measurements.

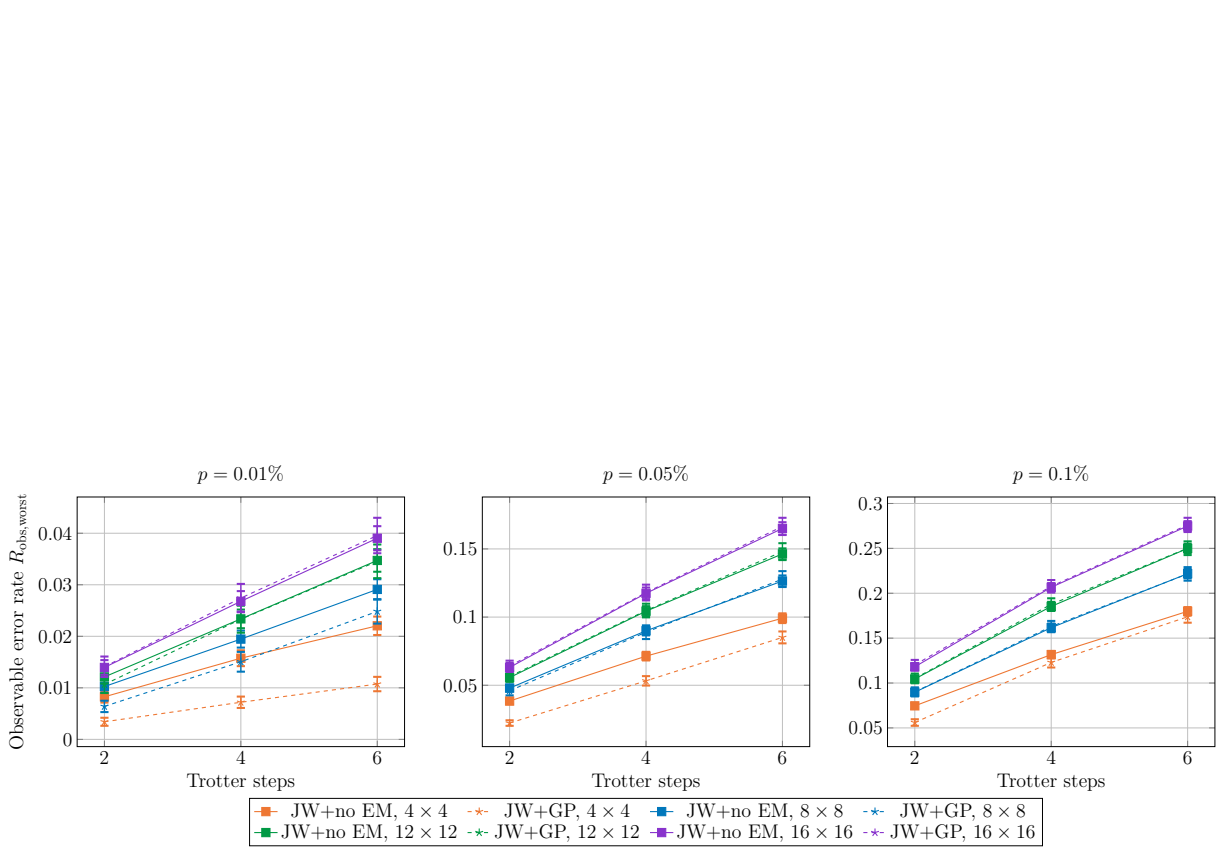


Figure B.9: **Effect of global parity postselection (GP) in the JW encoding for the standard depolarizing error model.** We consider fermionic lattice sizes of 4×4 up to 16×16 and occupation number measurements. One Trotter step consists of an application of all terms in the Hamiltonian and $R_{\text{obs,worst}}$ is the maximum local observable error rate. Error bars are computed using bootstrap resampling with 1000 resamples.