QAT

A Deterministic Decoration Scheme for Measurement-Based Variational Quantum Eigensolvers

Master of Physics of Complex Systems

Candidate:
Martina Leonetti

Supervisors:
Prof. Harold Ollivier
Prof. Vittorio Penna

Contents

Chapter 1

Abstract

The Variational Quantum Eigensolver (VQE) is a leading hybrid quantum-classical algorithm for finding ground states of quantum Hamiltonians on noisy quantum devices. Traditional circuit-based VQE implementations use sequences of quantum gates, but these require deep circuits that are challenging for current quantum hardware. An alternative approach is Measurement-Based VQE (MB-VQE), which performs the same variational optimization using single-qubit measurements on pre-prepared entangled resource states (graph states) instead of gate operations. However, measurement-based quantum computing requires deterministic computation, which is guaranteed only when the underlying graph state satisfies the generalized flow (gflow) condition - ensuring that the final output state is predictable despite the inherent randomness of individual quantum measurements.

This work analyzes existing MB-VQE methods, particularly the edge-decoration scheme, and identifies gflow violation in decorated graph states that lead to non-deterministic quantum computation. We focus on stabilizer Hamiltonians whose ground states are exactly known, and investigate how MB-VQE performs when small perturbations are added, making the perturbed ground state unknown and requiring variational approximation methods. To address the determinism issue, we develop a modified edge-decoration approach using dynamic qubit relabeling that preserves gflow conditions throughout the measurement process by sequentially applying decorations to qubit pairs while dynamically reassigning input/output roles to avoid shared-vertex conflicts. Numerical simulations on perturbed Toric code, linear cluster, and other graph-based stabilizer Hamiltonians demonstrate that our deterministic MB-VQE approach successfully approximates the perturbed ground states while maintaining computational reliability.

Chapter 2

Introduction

Quantum computers process information using quantum mechanical phenomena such as superposition and entanglement, enabling computational approaches fundamentally different from classical methods. Celebrated algorithms including Shor's factoring algorithm [1] and Grover's search speedup [2] demonstrate exponential or quadratic quantum advantages for specific tasks. However, current Noisy Intermediate-Scale Quantum (NISQ) devices [3] suffer from high error rates and limited coherence times, preventing the execution of these deep quantum algorithms.

Despite these limitations, quantum computers show promise for investigating problems in molecular structure, materials science, and condensed matter physics, where even approximate quantum solutions could provide valuable insights beyond classical capabilities. The Variational Quantum Eigensolver (VQE) [4] has emerged as the leading algorithmic approach for NISQ devices. VQE combines quantum hardware with classical optimization: a parameterized quantum circuit, known as the ansatz, prepares trial quantum states representing candidate solutions, the energy expectation value is measured, and classical optimization adjusts the circuit parameters to minimize this energy. This hybrid approach has been successfully applied to diverse quantum systems, from small molecular systems such as H_2 [4], LiH, and BeH_2 [5] to quantum many-body models and optimization problems.

However, VQE faces a fundamental scalability problem. Larger and more complex quantum systems require deeper quantum circuits with more gate operations to adequately represent their quantum states. Unfortunately, deeper circuits accumulate more errors and require longer coherence times, quickly exceeding the capabilities of NISQ devices. This creates a bottleneck where the most computationally interesting quantum systems require circuits too deep for current quantum hardware.

Measurement-Based Quantum Computing (MBQC) [6] offers a fundamentally different approach that could potentially address this circuit depth problem. Instead of applying sequences of quantum gates, MBQC performs computation through adaptive single-qubit measurements on a highly entangled resource state [7], specifically a graph state, where measurement results on some qubits determine which measurements to perform on subsequent qubits. Rather than building deep circuits gate-by-gate, MBQC requires only the preparation of an entangled resource state followed by single-qubit measurements, which are typically faster and less error-prone than two-qubit gates.

However, MBQC introduces a new challenge: determinism. Quantum measurements are inherently random, but computational output states must be deterministic and reproducible. MBQC achieves determinism through classical processing of measurement outcomes and adaptive corrections, but this is only guaranteed when the graph state, the resource state of the computation, satisfies a mathematical condition called *generalized flow* (gflow) [8]. When gflow is violated, the computation becomes non-deterministic.

Recent work has begun exploring Measurement-Based VQE (MB-VQE), which combines the variational approach of VQE with the measurement-based paradigm of MBQC. Early proposals include the edge-decoration scheme [9], which modifies graph states—quantum states represented as networks where qubits are vertices and the edges represent the entanglement between them—to encode variational parameters through measurement angles. However, systematic comparison of these methods and understanding of their gflow properties remains limited.

This work addresses these limitations by focusing on *stabilizer Hamiltonians*—quantum many-body systems whose ground states can be exactly determined in their unperturbed form (such as cluster states or topological codes like the Toric code model). When small perturbations are added to these systems, the exact ground state is no longer known and must be approximated. These systems are ideal testbeds for MB-VQE methods because the unperturbed ground state provides a natural starting point for variational optimization, while the underlying graph structure can be directly mapped to MBQC resource states.

Our contributions include: systematic analysis of the existing edge-decoration MB-VQE scheme; identification of gflow violations that compromise deterministic computation; development of a modified edge-decoration approach using dynamic qubit relabeling to ensure determinism; and numerical validation on perturbed stabilizer Hamiltonians including Toric code and linear cluster models.

These advances provide a viable solution for deterministic measurement-based variational quantum algorithms and their application to quantum many-body physics.

The thesis is structured as follows: Chapter 3 provides the theoretical background, Chapter 4 presents the models of quantum computation, with focus on MBQC, Chapter 5 presents the NISQ era and the variational quantum algorithms framework, in particular the Variational Quantum Eigensolver. Chapter 6 introduces Measurement-Based Variational Quantum Eigensolvers, analyzes existing approaches including edge-wise and node-wise decoration schemes, and identifies the determinism challenges that arise in edge-wise decoration due to gflow violations. Chapter 7 provides a detailed analysis of these determinism issues and proposes a novel solution using dynamic qubit relabeling to ensure reliable computation, with numerical validation on perturbed stabilizer Hamiltonians. Chapter 8 presents discussion, conclusions and future directions.

Chapter 3

Theoretical Background

This chapter provides the necessary quantum information foundation for readers unfamiliar with the field. I introduce the principles that distinguish quantum systems from classical ones, then develop the mathematical framework of the stabilizer formalism, concepts essential for understanding the quantum computing innovations presented in subsequent chapters [11].

3.1 Fundamentals of Quantum Information

Quantum information theory recasts information processing in terms of quantum mechanics and is more than classical computation, relying on the many unique quantum phenomena [11].

3.1.1 Quantum States and Operations

To understand quantum computation, we must first grasp how quantum information differs fundamentally from classical information. While classical bits are definitively either 0 or 1, quantum bits can exist in superpositions of both states simultaneously. Quantum states live in Hilbert space (a complex vector space with an inner product), which provides the mathematical foundation for describing quantum mechanical phenomena. In Dirac notation quantum states are denoted as kets $|\psi\rangle$ and the corresponding dual bras $\langle\psi|$. Physical states are unit vectors ($\langle\psi\rangle\,\psi = 1$), and states that differ by the global phase ($e^{i\theta}\,|\psi\rangle$) are physically equivalent, a property that makes sense collectively, but has no classical analogy.

A quantum bit (qubit) inhabits a two-dimensional Hilbert space $\mathcal{H} = \mathbb{C}^2$ with computational basis $\{|0\rangle\,, |1\rangle\}$. Its general state is as follows:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{3.1}$$

where $\alpha, \beta \in \mathbb{C}$ satisfies $|\alpha|^2 + |\beta|^2 = 1$. Unlike classical bits, which must be either 0 or 1, qubits exist in a continuous space of possible superpositions between these basis states. This superposition can be visualized geometrically using the Bloch sphere, where every point on the surface represents a possible qubit state. This geometric representation reveals that qubits don't just have "more" states than classical bits—they have a continuum of possible states. The Bloch sphere 3.1 provides a geometric representation, parameterizing any pure-qubit state as:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{3.2}$$

where $\theta \in [0, \pi]$ represents the polar angle and $\phi \in [0, 2\pi)$ the azimuthal angle on the sphere.

There are many ways in which a qubit can be realized physically: it can be carried by a single atom, or a single electron, or a single photon.

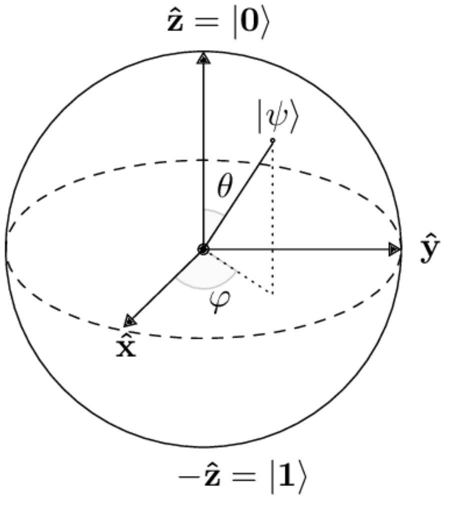

$\hat{\mathbf{z}} = |\mathbf{0}\rangle$
$|\psi\rangle$
θ
$\hat{\mathbf{y}}$
φ
$\hat{\mathbf{x}}$
$-\hat{\mathbf{z}} = |\mathbf{1}\rangle$

Figure 3.1: Bloch Sphere

For n qubits, the state space grows exponentially as $\mathcal{H}^{(n)} = (\mathbb{C}^2)^{\otimes n}$ with dimension 2^n, highlighting a fundamental advantage of quantum information: the ability to represent complex amplitudes 2^n with just n physical qubits. A general n-qubit state takes the form:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle \tag{3.3}$$

where $\sum_i |\alpha_i|^2 = 1$ and $|i\rangle$ represents the computational basis state corresponding to the binary representation of the integer i.

Quantum operations are implemented through three fundamental classes of operators:

1. Unitary operators ($U^\dagger U = UU^\dagger = I$): Govern reversible evolution of closed quantum systems, preserving state norms and superpositions. 2. Hermitian operators ($A = A^\dagger$): Represent physical observables with real eigenvalues. 3. Projection operators ($P^2 = P, P = P^\dagger$): Implement measurements and filter quantum states, causing the nonunitary state collapse that bridges quantum and classical domains.

Single-qubit operations include the Pauli gates:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3.4}$$

The X gate acts as a quantum NOT operation, flipping $|0\rangle$ to $|1\rangle$ and vice versa. The Z gate introduces a relative phase, leaving $|0\rangle$ unchanged while mapping $|1\rangle$ to $-|1\rangle$. The Y gate combines bit-flip and phase operations. Together with the identity I, these four operators form a basis for all single-qubit operations.

The Hadamard gate creates superpositions:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{3.5}$$

It transforms $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, effectively mapping between the computational bases (Z) and Hadamard (X).

Multi-qubit gates like the Controlled-NOT (CNOT) introduce entanglement:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3.6}$$

This gate flips the target qubit if and only if the control qubit is $|1\rangle$. When applied to $|+\rangle|0\rangle$, it creates the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

3.1.2 Entanglement and Measurement

Entanglement represents a purely quantum correlation between subsystems with no classical analog, what Einstein called *spooky action at a distance*. For composite systems, an entangled state cannot be expressed as a tensor product of individual subsystem states:

$$|\psi\rangle \neq |\psi_A\rangle \otimes |\psi_B\rangle \otimes \cdots \tag{3.7}$$

The Bell state exemplifies two-qubit entanglement:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{3.8}$$

In this state, measuring one qubit instantly determines the state of the other, regardless of spatial separation. This perfect correlation persists in any measurement basis, leading to violations of Bell inequalities and confirming the non-local nature of quantum mechanics.

For multi-qubit systems, more complex entangled states exist, such as the GHZ state:

$$|\text{GHZ}\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}) \tag{3.9}$$

and the W state:

$$|W\rangle = \frac{1}{\sqrt{n}}(|10...0\rangle + |01...0\rangle + ... + |00...1\rangle) \tag{3.10}$$

Measurement in quantum mechanics bridges quantum states and classical outcomes, introducing fundamental randomness. Formally, measurements are described by a set of operators $\{M_m\}$ satisfying the completeness relation:

$$\sum_m M_m^\dagger M_m = I \tag{3.11}$$

For a state $|\psi\rangle$, the probability of obtaining outcome m is:

$$p(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle \tag{3.12}$$

After obtaining outcome m, the system collapses to:

$$|\psi_m\rangle = \frac{M_m |\psi\rangle}{\sqrt{p(m)}} \tag{3.13}$$

This quantum state collapse represents a nonunitary, irreversible process fundamentally different from unitary evolution.

Projective measurements, the most common type, use orthogonal projectors P_i corresponding to an observable's eigenspaces. For an observable A with spectral decomposition $A = \sum_i \lambda_i P_i$, the measurement results are the eigenvalues λ_i with probabilities determined by the projection of the state onto each eigenspace. The expectation value is:

$$\langle A \rangle = \langle \psi | A | \psi \rangle = \sum_i \lambda_i p(i) \tag{3.14}$$

This measurement formalism explains both the probabilistic nature of quantum mechanics and the extraction of classical information from quantum systems.

3.2 Stabilizer Formalism

The stabilizer formalism, introduced by Gottesman [12], provides an efficient way to describe certain quantum states without storing exponentially many complex coefficients. Instead of characterizing an n-qubit state through 2^n amplitudes, this approach describes states using a polynomial number of operators that leave the state invariant.

3.2.1 The Pauli Group and Stabilizer States

The key insight is to characterize quantum states through operators that fix them. A *stabilizer* for a state $|\psi\rangle$ is an operator S such that $S|\psi\rangle = +|\psi\rangle$ — the state is a $+1$ eigenstate of S. These operators literally "stabilize" the state by leaving it unchanged.

The relevant operators come from the Pauli group $\mathcal{P}_n$, which consists of all tensor products of single-qubit Pauli matrices $\{I, X, Y, Z\}$ with global phases $\{\pm 1, \pm i\}$:

$$\mathcal{P}_n = \{c \cdot P_1 \otimes P_2 \otimes \cdots \otimes P_n \,|\, P_i \in \{I, X, Y, Z\}, c \in \{\pm 1, \pm i\}\} \tag{3.15}$$

To uniquely specify a state, we need a complete set of commuting stabilizers. Since Pauli operators either commute or anticommute, we can find sets of mutually commuting operators. A *stabilizer group* is such a commuting set that generates enough constraints to uniquely determine the state. For n qubits, we need exactly n independent stabilizing operators.

For example, $|0\rangle$ is stabilized by Z (since $Z|0\rangle = +|0\rangle$), while the Bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is uniquely determined by the two commuting stabilizers $X \otimes X$ and $Z \otimes Z$.

For the Bell state, for example. we can verify:

$$(X \otimes X)|\Phi^+\rangle = \frac{1}{\sqrt{2}}(X|0\rangle \otimes X|0\rangle + X|1\rangle \otimes X|1\rangle) \tag{3.16}$$

$$= \frac{1}{\sqrt{2}}(|1\rangle \otimes |1\rangle + |0\rangle \otimes |0\rangle) \tag{3.17}$$

$$= \frac{1}{\sqrt{2}}(|11\rangle + |00\rangle) = |\Phi^+\rangle \tag{3.18}$$

Similarly, $(Z \otimes Z)|\Phi^+\rangle = |\Phi^+\rangle$.

The efficiency of the stabilizer formalism becomes clear in terms of storage: instead of 2^n complex amplitudes, we need only n Pauli operators. Each operator requires $O(n)$ bits to specify (2 bits per qubit for $\{I, X, Y, Z\}$ plus bits for the phase), giving total storage of $O(n^2)$ bits—exponentially better than the standard $O(n \cdot 2^n)$ representation.

Geometrically, k independent stabilizer constraints reduce the 2^n-dimensional Hilbert space to a 2^{n-k}-dimensional subspace. Each constraint halves the allowed space. This dimensional control enables quantum error correction by encoding logical information in protected stabilized subspaces.

Clifford operations are quantum gates that preserve the Pauli group under conjugation ($C\mathcal{P}_n C^\dagger = \mathcal{P}_n$), meaning they map Pauli operators to Pauli operators and consequently transform stabilizer states into stabilizer states. The Gottesman-Knill theorem [12] demonstrates that quantum circuits restricted to stabilizer state preparation, Clifford gates, and Pauli measurements can be efficiently simulated on classical computers in polynomial time, establishing that non-Clifford resources are necessary for quantum computational advantage.

3.2.2 Stabilizer Hamiltonians

Finding ground states of generic k-local Hamiltonians is computationally challenging because the Hilbert space grows exponentially with system size; the associated decision version—the k-local Hamiltonian problem—is QMA-complete, i.e. hard even for quantum computers [13]. Classical approaches include exact diagonalization for very small systems, Density-Matrix Renormalization Group (DMRG) for 1-D chains, tensor-network extensions (PEPS, MERA) in higher dimensions, Quantum Monte Carlo (QMC) and Coupled-

Cluster methods effective for weakly correlated molecules [14, 15]. Quantum approaches comprise the Variational Quantum Eigensolver (VQE) for NISQ hardware, Quantum Phase Estimation (QPE) on future fault-tolerant machines, and Adiabatic Quantum Computing (AQC) [16, 17].

However, *stabilizer Hamiltonians* of the form $H_{\text{stab}} = -\sum_\alpha S_\alpha$ (where S_α are stabilizer operators) can be solved efficiently on classical computers due to the Gottesman-Knill theorem, as their eigenstates are stabilizer states. Examples include the Toric Code and cluster state Hamiltonians. In contrast, for instance, the transverse-field Ising model $H_{\text{TFIM}} = -J\sum_i Z_i Z_{i+1} - h\sum_i X_i$ is not a stabilizer Hamiltonian due to non-commuting terms, making it computationally hard to simulate classically. In the folliwing subsection we will present the Toric Code, whose Hamiltonian is a stabilizer Hamiltonian.

3.2.3 Toric Code

The toric code, introduced for the first time in the work [18], is both a topological and a stabilizer quantum code. To understand the toric code, we first need to understand what stabilizer and topological codes are. In quantum error correction, we want to protect quantum information from noise and errors by encoding the quantum information we actually care about - that is stored into "logical qubits". into a larger space of "physical qubits" - the actual quantum hardware components that can be manipulated and measured. A stabilizer code is defined by a set of operators called stabilizers, which we introduced in the theoretical background. A topological code stores logical qubits in global properties of the entire system rather than in individual components. In few words, this means that errors must affect large regions of the system to corrupt the logical information, making it robust against local perturbations. This was just a brief highlight for understaning the importance of the toric code, we can focus on what we need for our specific problem. The toric code is defined on a 2D square lattice $L \times L$ placed on a torus surface 3.2. Physical qubits are placed on the edges of the square lattice. $2L^2$ physical qubits are required to encode 2 logical qubits.

The code uses two types of stabilizer operators to detect errors: star operators and plaquette operators. Star operators are defined at each vertex s of the lattice:

$$\hat{A}_s = \prod_{i \in \text{edges}(s)} \hat{X}_i \tag{3.19}$$

These are products of Pauli-$\hat{X}$ operators on the four edges connected to

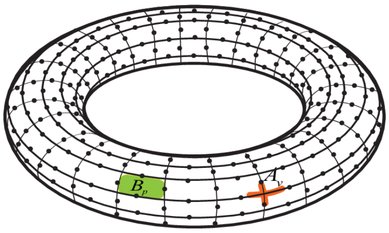


Figure 3.2: Torus suface of the toric code

vertex s. Plaquette operators are defined on each square face p of the lattice:

$$\hat{B}_p = \prod_{i \in \text{boundary}(p)} \hat{Z}_i \tag{3.20}$$

These are products of Pauli-$\hat{Z}$ operators on the four edges that form the boundary of plaquette p. All stabilizer operators commute with each other: $[\hat{A}_s, \hat{A}_{s'}] = [\hat{B}_p, \hat{B}_{p'}] = [\hat{A}_s, \hat{B}_p] = 0$ for all vertices s, s' and plaquettes p, p'. The code space consists of all quantum states that are $+1$ eigenstates of every stabilizer operator. Since the lattice is on a torus, periodic boundary conditions are present on the lattice. In figure 3.3 the two-dimensional lattice is illustrated where N_x is the number of rows of indipendent vertices and N_y the number of columns of indipendent vertices on the lattice. From now on, we will consider $N_x = N_y = 2$.

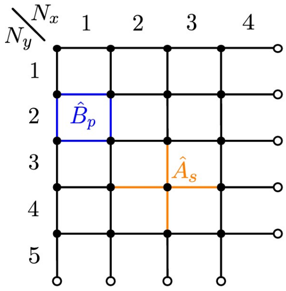


Figure 3.3: Toric code on a square lattice [9]. The star operator $\hat{A}_s$ acts on the four edges adjacent to a vertex, while the plaquette operator $\hat{B}_p$ acts on the four edges forming plaquette p.

The toric code Hamiltonian is constructed from the stabilizer operators:

$$\hat{H} = -\sum_s \hat{A}_s - \sum_p \hat{B}_p \tag{3.21}$$

This Hamiltonian describes a system where the ground state corresponds to the error-free code space. According to the theory of quantum error correction in which we don't go into details, when all stabilizer operators have eigenvalue $+1$, no errors are present and energy is minimized. Errors flip some stabilizers to eigenvalue -1, increasing the energy and naturally driving the system back toward the error-free ground state.

The Hamiltonian has four degenerate ground states $|r,t\rangle_L$ with $r, t = 0, 1$, called logical states. The reason of this four-fold degeneracy is related to its encoding of 2 logical qubits. Since each logical qubit can be in states $|0\rangle$ or $|1\rangle$, the two logical qubits form four combinations: $|0,0\rangle_L$, $|0,1\rangle_L$, $|1,0\rangle_L$, and $|1,1\rangle_L$. All four states are simultaneous eigenstates of every stabilizer operator with eigenvalue $+1$:

$$\hat{A}_s|r,t\rangle_L = |r,t\rangle_L \quad \forall s \tag{3.22}$$

$$\hat{B}_p|r,t\rangle_L = |r,t\rangle_L \quad \forall p \tag{3.23}$$

Therefore, they are degenerate eigenstates of the Hamiltonian $\hat{H}$. Since the stabilizer operators satisfy two constraints ($\prod_s \hat{A}_s = \prod_p \hat{B}_p = 1$), there are only 6 independent stabilizers ($2N_xN_y - 2$) instead of 8. This is insufficient to fully define the 8-qubit ground state, so two additional stabilizers are needed.

The standard choice is to add the logical operators $\hat{X}_{L_1}$, $\hat{X}_{L_2}$ or $\hat{Z}_{L_1}$, $\hat{Z}_{L_2}$ 3.4 3.5. These commute with all plaquette and star operators, that is why they are chosen to complete the stabilizer characterization of one of the four degenerate ground states. Explicitly, $\hat{Z}_{L,1}$ ($\hat{Z}_{L,2}$) applies the $\hat{Z}$ operator to all horizontal (vertical) edges of an arbitrarily chosen column (row). Similarly, $\hat{X}_{L,1}$ ($\hat{X}_{L,2}$) applies the $\hat{X}$ operator to all horizontal (vertical) edges of an arbitrarily chosen row (column) [9].

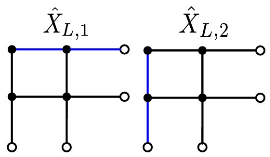


Figure 3.4: [9] $\hat{X}_{L_1}$ and $\hat{X}_{L_2}$ logical operators. Blue lines indicate the qubits on which the operators act.

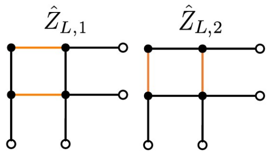


Figure 3.5: [9] $\hat{Z}_{L_1}$ and $\hat{Z}_{L_2}$ logical operators. Orange lines indicate the qubits on which the operators act.

Chapter 4

Models of Quantum Computation

The concept of quantum computation emerged in the early 1980s when researchers like Benioff and Feynman recognized that quantum mechanics could fundamentally enhance computational capabilities [19, 20]. This insight led to the development of different computational models, each exploiting quantum properties in distinct ways.

The circuit model, pioneered by Deutsch, became the dominant paradigm through its intuitive gate-based approach [21]. Shor's 1994 factoring algorithm demonstrated the circuit model's power for specific problems, leading to widespread adoption in quantum algorithm development [?]. However, the circuit model's requirement for deep, sequential gate operations poses significant challenges for noisy intermediate-scale quantum (NISQ) devices.

Alternative computational models emerged to address these limitations. Measurement-based quantum computing exploits pre-prepared entangled graph states and adaptive measurements, potentially reducing circuit depth requirements. Adiabatic quantum computing leverages quantum annealing principles, offering a different approach to optimization problems.

Understanding these different models is crucial because they offer complementary approaches to quantum algorithm design and hardware implementation. Each model presents unique trade-offs between algorithmic expressiveness, noise resilience, and physical realizability. This chapter examines the mathematical foundations of these quantum paradigms.

4.1 Circuit-Based Quantum Computation

Most current quantum processors adopt the *gate-based* (or *circuit-based*) model, executing an algorithm as a time-ordered sequence of quantum gates acting on qubits [11].

4.1.1 Fundamental Elements

A circuit diagram is a set of horizontal wires—one per qubit—that all begin in the basis state $|0\rangle$. Gates are drawn as boxes or symbols on the wires, and time flows from left to right, so the left-most gate acts first. Single-qubit gates such as Hadamard (H), Pauli-X (X), Pauli-Y (Y), Pauli-Z (Z), phase (S) and $\pi/8$ (T) create and rotate superpositions on a single qubit.

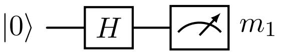


Figure 4.1: A Hadamard gate prepares $(|0\rangle + |1\rangle)/\sqrt{2}$; the final measurement collapses the superposition to a classical bit m_1.

Two-qubit gates are indispensable because they generate correlations that single-qubit rotations cannot achieve. One example is the controlled-NOT (CNOT), which flips the target qubit when the control is in $|1\rangle$. A quantum circuit represents a series of unitary operations applied to a quantum state. For n qubits, the state is represented as a vector in a 2^n-dimensional Hilbert space. Each gate corresponds to a unitary matrix U satisfying $U^\dagger U = \mathbb{I}$, ensuring reversibility and conservation of probability in accordance with quantum mechanics.

The evolution of a quantum state through a circuit is described by the sequential application of unitary operators:

$$|\psi_{\text{final}}\rangle = U_k \cdots U_2 U_1 |\psi_{\text{init}}\rangle$$

A *finite* family of quantum gates is called **universal** when *any* operation allowed by quantum mechanics can be built out of them. Formally, for a register of n qubits, every unitary matrix acting on the 2^n-dimensional state space can be *approximated arbitrarily well* by a finite sequence of gates chosen only from that family.

A widely used universal set consists of just four blocks:

- the **Hadamard** gate, H,

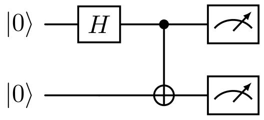
- the **phase** gate, S,
- the $\pi/8$ gate, T, and
- the two-qubit **controlled-NOT** (CNOT).

Because any quantum circuit can be rewritten in terms of these four gates, experimental platforms focus on implementing them (or an equivalent universal set) with high fidelity.

4.1.2 Example: Bell State Preparation

Consider the preparation of a Bell state—a maximally entangled two-qubit state.

1. Initialize two qubits in the computational basis state $|00\rangle$.
2. Apply a Hadamard gate to the first qubit, creating the superposition state $\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$.
3. Apply a CNOT gate with the first qubit as control and the second as target, resulting in the entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.
4. Upon measurement, the system yields either 00 or 11 with equal probability.



4.1.3 Challenges and Limitations

Even though the circuit model is conceptually simple, today's hardware struggles with some problems:

1. **Fragility to the environment (noise and decoherence).**
   A qubit keeps its quantum properties only for a short *coherence time*—from a few microseconds in superconducting chips to a few seconds in trapped ions. Two-qubit gates fail roughly once every 10^2–10^3 operations, so long circuits quickly drown in errors.
2. **Imperfect entangling gates.**
   Operations such as CNOT demand nanometre-level control of lasers or

microwave pulses. Any mis-tuning spreads through the computation, making algorithms that need thousands of CNOTs unfeasible on *NISQ* (Noisy Intermediate-Scale Quantum) devices.

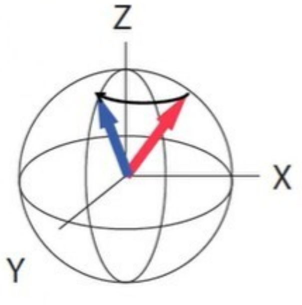
3. **Error-correction overhead.**
   Protecting information requires many physical qubits for *one* reliable logical qubit—typically 10^2–10^3. Shor's algorithm on a 2048-bit RSA key would need millions of physical qubits, whereas current prototypes offer about 10^3.

4. **Timing and connectivity constraints.**
   Qubits are not all-to-all connected; extra *swap* gates are needed to bring distant qubits together, lengthening the circuit and adding further error sources. All gates must also fit within the short coherence window.

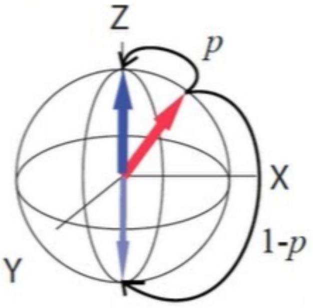
4.2 Measurement-Based Quantum Computation

Measurement-based quantum computation (MBQC), also known as the "one-way quantum computer", is an universal model of quantum computation introduced by Raussendorf and Briegel in 2001 [6]. Instead of executing the quantum computation via a sequence of reversible unitary gates—as in the circuit model—MBQC relies on preparing a single, highly entangled *resource state*, and then performing the entire computation solely through sequential single-qubit projective measurements on that state. This means that local measurements on qubits are used to drive the computation.

In general, quantum mechanical systems can evolve in time in two ways: via unitary evolution and via projective evolution.



Unitary evolution:
deterministic, reversible



Projective measurement:
probabilistic, irreversible

- **Unitary evolution:** Governed by the Schrödinger equation, producing continuous, deterministic, and reversible changes according to:

$$|\psi(t)\rangle = e^{-i\hat{H}t/\hbar}|\psi(0)\rangle \tag{4.1}$$

where $\hat{H}$ is the system Hamiltonian. For any later time, we will know exactly what state we have, if we know the initial state. Moreover, if we have the tools to change $\hat{H}$ o $-\hat{H}$, then we can undo the evolution and get back to where we started.

- **Projective measurements:** If we make a measurement on the system, there is no way of knowing what we're going to get, the outcome is probabilistic: the projective measurements cause discontinuous and probabilistic state collapse.

The framework of MBQC is known as the "one-way quantum computer" since the measurements destructively consume the entanglement, rendering the cluster state usable only once as a computational resource.

Despite the differences, both models are computationally equivalent—any quantum algorithm efficiently implementable in one model can be implemented efficiently in the other.

4.2.1 Graph States

A graph state is a special class of multi-qubit entangled state used as resource states in MBQC. Formally, a graph state $|G\rangle$ is determined by an undirected graph $G = (V, E)$, where V is the set of vertices that correspond to qubits and E is the set of edges, where each edge represents the entanglement between two qubits.

A graph state $|G\rangle$ is prepared as follows:
All qubits are set to $|+\rangle$, where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and then a Controlled-Z (CZ) gate is applied between every pair of qubits connected by an edge of the graph. Equivalently, $|G\rangle$ can be defined via the stabilizer formalism as the unique state which is the simultaneous $+1$ eigenstate of a commuting set $\{K_v\}$, for each vertex $v \in V$, where the stabilizer operator is:

$$K_v = X_v \prod_{w \in N_v} Z_w \tag{4.2}$$

where N_v denotes the neighborhood of the vertex v, that is, all vertices connected to v by an edge.

4.2.2 Cluster states

A cluster state [7] is a particular instance of graph states, where the underlying graph is a connected subset of a d-dimensional lattice. Vertices are arranged in a regular lattice (commonly a square lattice in 2D or a cubic lattice in 3D) and connected by edges that connect nearest neighbors, so that the states are generated in lattices of qubits with Ising-type interactions.
The 2D cluster state is a computationally universal "material", we can implement any quantum computation on it, solely by local measurements.

On the experimental side, cluster states have been produced in different quantum platforms, such as photonic qubits, trapped ions, and superconducting qubits.

A possible description of cluster states is based on the stabilizer formalism. For example the 1D cluster being then just the unique quantum state with +1 eigenvalue to n independent operators $\{S_i\}_{i=1}^n$ of the form $S_i = Z_{i-1}X_iZ_{i+1}$.

Two-Qubit Cluster State

The simplest meaningful cluster state involves just two qubits:

$$\begin{aligned} |C_2\rangle &= CZ_{1,2}|+\rangle_1|+\rangle_2 && (4.3)\\ &= CZ_{1,2}\left(\frac{|0\rangle_1+|1\rangle_1}{\sqrt{2}}\right)\left(\frac{|0\rangle_2+|1\rangle_2}{\sqrt{2}}\right) && (4.4)\\ &= \frac{1}{2}(|0\rangle_1|0\rangle_2+|0\rangle_1|1\rangle_2+|1\rangle_1|0\rangle_2-|1\rangle_1|1\rangle_2) && (4.5)\\ &= \frac{|0+\rangle_{1,2}+|1-\rangle_{1,2}}{\sqrt{2}} && (4.6) \end{aligned}$$

This state is equivalent to a Bell pair (a maximally entangled state) if we apply a Hadamard gate to the second qubit:

$$(I_1 \otimes H_2)|C_2\rangle = \frac{|00\rangle_{1,2}+|11\rangle_{1,2}}{\sqrt{2}} =: |\text{Bell}\rangle \qquad (4.7)$$

4.2.3 Single qubit unitary on a one dimensional cluster state

To understand how to translate a circuit-based single qubit unitary in the measurement-based framework, consider a two-qubit chain where we measure qubit 0 at angle θ in the XY plane. The measurement operator is

$M_0(\theta) = \cos(\theta)X_0 + \sin(\theta)Y_0$ with eigenstates $|\theta^+\rangle = |0\rangle + e^{i\theta}|1\rangle$ and $|\theta^-\rangle = |0\rangle - e^{i\theta}|1\rangle$ (up to normalization). We measure qubit 0 in state

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ while qubit 1 is in the $|+\rangle$ state:

$$M_0(\theta)CZ_{01}|\psi\rangle_0|+\rangle_1 = M_0(\theta)(\alpha|00\rangle + \alpha|01\rangle + \beta|10\rangle - \beta|11\rangle)$$

There are two possible measurement outcomes: 0 or 1. If we obtain outcome 0, this means the state is projected onto the eigenstate $|\theta^+\rangle = |0\rangle + e^{i\theta}|1\rangle$, and the state of the system becomes:

$$\begin{aligned} M_0^0 CZ_{01}|\psi\rangle_0|+\rangle_1 &= (\langle 0| + e^{i\theta}\langle 1|)(\alpha|00\rangle + \alpha|01\rangle + \beta|10\rangle - \beta|11\rangle) \\ &= \alpha|0\rangle + \alpha|1\rangle + \beta e^{i\theta}|0\rangle - \beta e^{i\theta}|1\rangle \\ &= \alpha|+\rangle + \beta e^{i\theta}|-\rangle \\ &= HR_z(-\theta)|\psi\rangle \end{aligned} \tag{4.8}$$

If we obtain outcome 1, this means the state is projected onto the eigenstate $|\theta^-\rangle = |0\rangle - e^{i\theta}|1\rangle$, and the state of the system becomes:

$$\begin{aligned} M_0^1 CZ_{01}|\psi\rangle_0|+\rangle_1 &= (\langle 0| - e^{i\theta}\langle 1|)(\alpha|00\rangle + \alpha|01\rangle + \beta|10\rangle - \beta|11\rangle) \\ &= \alpha|0\rangle + \alpha|1\rangle - \beta e^{i\theta}|0\rangle + \beta e^{i\theta}|1\rangle \\ &= \alpha|+\rangle - \beta e^{i\theta}|-\rangle \\ &= HR_z(-\theta)Z|\psi\rangle \end{aligned} \tag{4.9}$$

The only difference between equations (4.8) and (7.4) is that in the first case there is no Z gate, while in the second case a Z gate appears. This Z is a so called by-product operator, and it's an undesired Pauli operator that comes up when the measurement outcome is 1.

In more compact form we have:

$$M_0(\theta)CZ_{01}|\psi\rangle_0|+\rangle_1 = HR_z(-\theta)Z^{m_0}|\psi\rangle$$

where m_0 is the measurement outcome (0 or 1).

This process is called quantum half teleportation: the quantum information from the first qubit moves to the second qubit, with a Hadamard transformation applied and a possible Pauli-Z correction depending on what we measured. When we store logical information in qubit 1 and then measure it, you might expect that information to be lost forever since measurement is normally irreversible. But surprisingly, the entanglement between the two qubits allows the quantum information to completely transfer to the second qubit, preserving it despite the measurement.

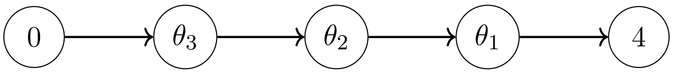
We know that in circuit-based quantum computation a unitary can be decomposed into a product of three rotations (Euler decomposition):

$$U = R_x(-\theta_1)R_z(-\theta_2)R_x(-\theta_3)$$

where R_x and R_z are the rotation matrices:

$$R_x(\theta) = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \quad R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

Consider a one-dimensional cluster chain of 5 qubits where qubit 0 is the input and qubit 4 is the output:



$$|\psi'\rangle = M_3M_2M_1M_0CZ_{34}CZ_{23}CZ_{12}CZ_{01}|\psi\rangle_0|+\rangle_1|+\rangle_2|+\rangle_3|+\rangle_4$$

Due to random measurement outcomes, MBQC requires **adaptive measurements**. The measurement angles must be corrected based on previous outcomes m_j:

$$\begin{array}{ll} \text{Qubit 0:} & 0 \mapsto 0 \\ \text{Qubit 1:} & \theta_1 \mapsto (-1)^{m_0}\theta_1 \\ \text{Qubit 2:} & \theta_2 \mapsto (-1)^{m_1}\theta_2 \\ \text{Qubit 3:} & \theta_3 \mapsto (-1)^{m_0+m_2}\theta_3 \end{array}$$

The final result is:

$$|\psi'\rangle = X^{m_3+m_1}Z^{m_2+m_0}U|\psi\rangle$$

where $U = R_x(-\theta_3)R_z(-\theta_2)R_x(-\theta_1)$ and $X^{m_3+m_1}Z^{m_2+m_0}$ are the *byproduct operators.*

4.2.4 Determinism in MBQC

As showed in the the previous examples, there are two key MBQC principles:

1. Deterministic computation requires adaptive measurements.

2. Adaptation of angles can be realized only up to byproduct operators.

Byproduct Correction The byproduct operators that emerge from the measurements must be corrected. In the framework of graph states, that we recall being the resource states for MBQC, depending on the graph topology we may not be able to eliminate the randomness. So, given a graph state, we're not always able to perform measurements on this state in such a way that the determinism is guaranteed. The success of this correction strategy depends crucially on the structure of the underlying graph state. This introduces a fundamental constraint: *the graph structure must satisfy specific mathematical conditions to enable deterministic computation.* A necessary condition is known as *flow* [25].

Flow Condition

Given a graph state representation of an MBQC resource, we can identify three types of qubits:

- **Input qubits** (I): where the computation begins
- **Output qubits** (O): where the final result is obtained
- **Auxiliary qubits**: all other qubits that are measured during computation

Definition 1 (Flow)**.** *An open graph state (G, I, O, λ) with measurements in the (X, Y) plane possesses a flow if there exists a function $f : O^c \to I^c$ (mapping measured qubits to correction qubits) and a partial order $>$ on vertices such that for all measured qubits $i \in O^c$:*

(F1) $i \sim f(i)$ (measured qubit is adjacent to its corrector)

(F2) $i < f(i)$ (measured qubit is processed before its corrector)

(F3) $\forall k \in N_G(f(i)) \setminus \{i\}$, $i < k$ (neighbors of corrector are processed after measured qubit)

Flow ensures both the temporal and causal measurement ordering required for adaptive measurements, and the geometric conditions necessary for byproduct correction. Specifically: (1) each measured qubit has a designated "corrector" qubit connected to it, (2) measurements occur in a specific temporal order that respects causality, and (3) byproduct propagation can proceed without conflicts.

Generalized Flow (Gflow)

The standard flow condition is sufficient but not necessary for deterministic MBQC, and is restricted to (X, Y) measurements with one-to-one correction mapping. **Generalized flow** (gflow) extends this framework by:

- Allowing measurements in all three planes: (X, Y), (X, Z), and (Y, Z)
- Permitting each measured qubit to have a *set* of correcting qubits rather than just one

Definition 2 (Generalized Flow)**.** *An open graph state* (G, I, O, λ) *has gflow if there exists a function* $g : O^c \to \mathcal{P}(I^c)$ *(mapping measured qubits to correction sets) and a partial order* $<$ *satisfying:*

- *(G1) If* $j \in g(i)$ *and* $i \neq j$*, then* $i < j$
- *(G2) If* $j \leq i$ *and* $i \neq j$*, then* $j \notin Odd(g(i))$
- *(G3) If* $\lambda(i) = (X, Y)$*:* $i \notin g(i)$ *and* $i \in Odd(g(i))$
- *(G4) If* $\lambda(i) = (X, Z)$*:* $i \in g(i)$ *and* $i \in Odd(g(i))$
- *(G5) If* $\lambda(i) = (Y, Z)$*:* $i \in g(i)$ *and* $i \notin Odd(g(i))$

Here, $\text{Odd}(K) = \{u \in V \mid |N_G(u) \cap K| \equiv 1 \pmod 2\}$ denotes vertices with an odd number of neighbors in set K.

The key insight is that gflow provides a *necessary and sufficient* condition for deterministic MBQC: a graph state enables deterministic computation if and only if it possesses gflow. This makes gflow the definitive criterion for assessing whether both adaptive measurements and byproduct correction can be successfully implemented on a given MBQC resource state.

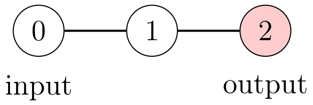
In the framework of the stabilizer formalism, it becomes intuitive to understand how the stabilizers of a graph state that admits gflow ensure determinism by enabling byproduct correction.

As we saw, there are two types of byproduct operators that arise from $R(\theta)$ measurements (rotations in the XY plane), each requiring different correction strategies:

- **X byproducts**: Can be easily corrected by swapping measurement bases $R(\theta) \leftrightarrow R(-\theta)$ on the respective qubit through adaptive measurements, since $XR(\theta)X = R(-\theta)$.

- **Z byproducts**: Require correction through the stabilizer structure of the graph state by commuting them to neighboring unmeasured qubits.

Consider for example a 3-qubit linear cluster state:



where qubit 0 is the input, qubit 2 is the output, and qubit 1 is measured during computation.

When we measure qubit 0 and obtain outcome $m_0 = 1$ (negative eigenstate), a Z byproduct operator appears, as demonstrated in the two-qubit chain example. This results in an unwanted Z_0 operator acting on the state.

The correction mechanism exploits the stabilizer structure of the graph state. For the 3-qubit linear cluster, the stabilizers are:

$$S_0 = X_0 Z_1 \tag{4.10}$$
$$S_1 = Z_0 X_1 Z_2 \tag{4.11}$$
$$S_2 = Z_1 X_2 \tag{4.12}$$

To correct the unwanted Z_0 byproduct, we observe that Z_0 appears in the stabilizer $S_1 = Z_0 X_1 Z_2$. Since stabilizers have eigenvalue $+1$ on the graph state by definition, we can write:

$$S_1|\mathrm{GS}\rangle = Z_0 X_1 Z_2 |\mathrm{GS}\rangle = |\mathrm{GS}\rangle$$

This means that applying Z_0 to the graph state is equivalent to applying $X_1 Z_2$:

$$Z_0|\mathrm{GS}\rangle = X_1 Z_2 |\mathrm{GS}\rangle$$

Therefore, the unwanted Z_0 byproduct is *commuted* through the stabilizer relation to neighboring unmeasured qubits, transforming into $X_1 Z_2$ byproducts. This is the general mechanism: Z byproducts from projections onto negative eigenstates can be shifted to neighboring qubits using stabilizer rules. Now we have two new byproducts to handle: X_1 can be corrected by adaptive measurements on qubit 1 (swapping $R(\theta_1) \to R(-\theta_1)$), while Z_2 appears on the output qubit where it can be corrected through classical post-processing.

This stabilizer-based correction mechanism automatically reveals the mathematical condition for determinism in MBQC: **the flow condition**.

Through our example, we can see that flow captures two essential requirements:

Correction Sets: Each measured qubit must have designated "corrector" qubits that can handle its byproducts. In our example: qubit 0 sends corrections to qubit 1 (the X_1 byproduct from the stabilizer transformation), and qubit 1 sends corrections to qubit 2 (any further byproducts are pushed to the output).

Measurement Order: There must be a specific temporal order that ensures corrector qubits are available when needed. In our case: qubit 0 must be measured before qubit 1. This is crucial because when qubit 0 produces the X_1 byproduct, qubit 1 must still be unmeasured so it can perform the adaptive correction.

All these rules can be visualized pictorially in Figure 4.2.

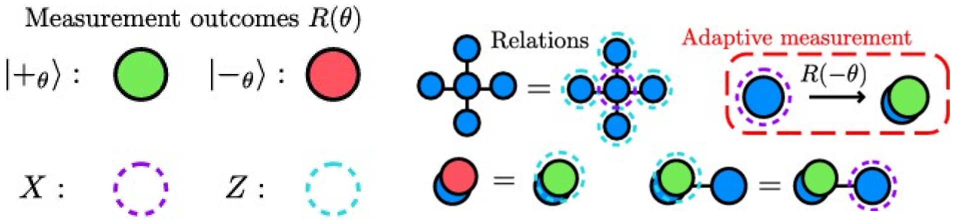


Figure 4.2: Figure from [10], a graphical representation of MBQC and byproduct corrections. Blue circles are the qubits

Chapter 5

Variational Quantum Algorithms

The term **NISQ** (*Noisy Intermediate-Scale Quantum*) was coined by John Preskill in 2018 to define the current class of quantum computing platforms. NISQ devices are characterized by two key limitations: they are **noisy**, referring to environmental influences such as thermal noise and electromagnetic field noise that result in computation errors, and **intermediate-scale**, containing 50 to a few hundred qubits—sufficient to exceed classical simulation for specific tasks but inadequate for full-scale quantum error correction.

The NISQ era represents a transitional phase where devices begin performing certain tasks faster than classical machines, achieving *quantum advantage* or *quantum supremacy.* Classical computers struggle particularly with simulating highly entangled many-particle quantum systems, making *quantum dynamics* a promising area for quantum computational advantages.

Despite their potential, NISQ devices face fundamental challenges that limit their computational capabilities and practical applications:

1. **High Error Rates:** Current quantum computers exhibit error rates much higher than conventional electronics, with insufficient quantum resources for powerful error correction protocols [26].

2. **Short Coherence Times:** Qubits spontaneously relax toward their ground state in microseconds to milliseconds, limiting successful computational operation windows.

3. **Limited Qubit Numbers:** Available qubits are insufficient for full-fledged quantum error correction protocols required for fault-tolerant

computation. Hundreds to thousands of physical qubits are still required for each logical qubit.

4. **Connectivity Constraints:** In most architectures, qubits cannot interact arbitrarily, limiting the complexity and efficiency of implementable quantum circuits.

5. **Measurement Errors:** Read-out fidelities remain at 99–99.8% (mid-circuit even lower), often dominating total error budgets [27].

6. **Gate Calibration Drift:** Quantum devices require frequent recalibration as gate parameters drift due to environmental fluctuations and hardware aging.

7. **Limited Circuit Depth:** The combination of decoherence and gate errors restricts quantum circuits to shallow depths, typically 10-100 gates before errors dominate computation.

Given these limitations, two primary approaches emerge for quantum computing's future. The first is implementing quantum error correction (QEC), where many physical qubits constitute a single robust logical qubit. The second approach accepts inherent noise and finds applications that survive despite it. Variational Quantum Algorithms represent a prominent example of this second strategy.

Building upon the NISQ era challenges, Variational Quantum Algorithms (VQAs) emerge as practical approaches to extract computational value from current quantum hardware. These hybrid algorithms leverage both classical and quantum computing resources to find approximate solutions to optimization and eigenvalue problems, with applications spanning physics, chemistry, finance, and logistics.

VQAs serve as a bridge between classical algorithms and fault-tolerant quantum algorithms of the future. While current NISQ devices lack the qubit count, coherence times, and error rates needed for full-scale fault-tolerant quantum algorithms, VQAs are specifically designed to work within these constraints. The fundamental insight behind VQAs is offloading computationally intensive optimization to classical computers while using quantum hardware for tasks it performs best: preparing and manipulating quantum states difficult to simulate classically.

The theoretical foundation of VQAs rests on the variational principle, a fundamental theorem in quantum mechanics providing mathematical guarantees for their effectiveness. For any Hamiltonian $\hat{H}$ with ground state energy

E_0 and any normalized quantum state $|\psi\rangle$, the variational principle states that $E_0 \leq \langle\psi|\hat{H}|\psi\rangle$. This inequality ensures that the expectation value of the Hamiltonian in any trial state provides an upper bound to the true ground state energy.

By parametrizing a family of quantum states $|\psi(\boldsymbol{\theta})\rangle$ and minimizing the cost function $C(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|\hat{H}|\psi(\boldsymbol{\theta})\rangle$, we systematically approach the ground state energy from above. The optimal parameters $\boldsymbol{\theta}^*$ that minimize $C(\boldsymbol{\theta})$ yield the best approximation to the ground state within the chosen parametric family.

VQAs consist of four essential steps:

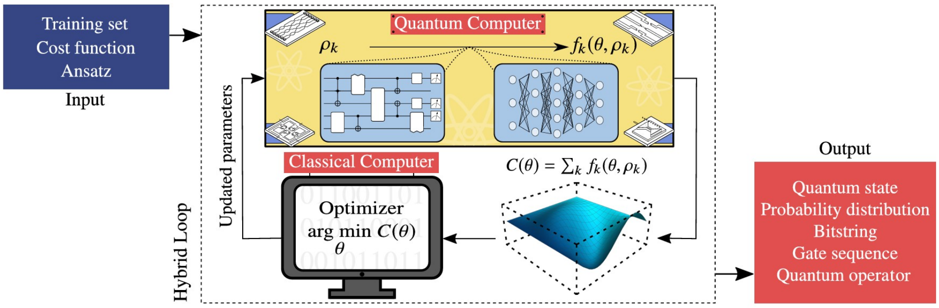


Figure 5.1: Schematic diagram of a Variational Quantum Algorithm (VQA) [29]

1. **Parameterized quantum circuit preparation:** Start with a parameterized quantum circuit (ansatz) where parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$ are rotation angles associated with quantum gates. This creates a mapping $\boldsymbol{\theta} \mapsto |\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0\rangle^{\otimes n}$, where $U(\boldsymbol{\theta})$ is a sequence of parameterized unitary rotations.

2. **Classical optimization:** Optimize parameters $\boldsymbol{\theta}$ using classical optimization algorithms to minimize a problem-specific cost function $C(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|\hat{H}|\psi(\boldsymbol{\theta})\rangle$, where $\hat{H}$ encodes the problem to be solved.

3. **Convergence assessment:** Monitor convergence through gradient analysis $\nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta})$ or cost function changes: $|C(\boldsymbol{\theta}_{i+1}) - C(\boldsymbol{\theta}_i)| < \epsilon$ for tolerance ϵ.

4. **Solution extraction:** Use optimized parameters $\boldsymbol{\theta}^*$ to prepare the final state $|\psi(\boldsymbol{\theta}^*)\rangle$ and extract solutions through classical post-processing.

Gradient computation requires special consideration for quantum circuits. Unlike classical neural networks with automatic differentiation, quantum circuits use the parameter shift rule. For a parameterized gate with parameter θ_i, the gradient is:

$$\frac{\partial}{\partial \theta_i} \langle \psi(\boldsymbol{\theta}) | \hat{H} | \psi(\boldsymbol{\theta}) \rangle = \frac{1}{2} \left[\langle \psi(\boldsymbol{\theta}^+) | \hat{H} | \psi(\boldsymbol{\theta}^+) \rangle - \langle \psi(\boldsymbol{\theta}^-) | \hat{H} | \psi(\boldsymbol{\theta}^-) \rangle \right]$$

where $\boldsymbol{\theta}^{\pm} = \boldsymbol{\theta} \pm \frac{\pi}{2}\mathbf{e}_i$. Parameter updates follow: $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \alpha \nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}^{(k)})$, where α is the learning rate.

The ansatz choice critically determines VQA performance by defining the explorable quantum state space. An ansatz creates a mapping from classical parameters to quantum states: $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0\rangle^{\otimes n}$. Most ansätze use layered structures: $U(\boldsymbol{\theta}) = U_L(\boldsymbol{\theta}^{(L)}) \cdot U_{L-1}(\boldsymbol{\theta}^{(L-1)}) \cdots U_1(\boldsymbol{\theta}^{(1)})$, where each layer applies operations to all qubits systematically.

Effective ansatz design balances three requirements: **expressivity** (range of representable quantum states), **trainability** (optimization landscape quality), and **implementability** (hardware compatibility with acceptable error rates). Hardware-efficient ansätze maximize available quantum resources while respecting device limitations, typically alternating single-qubit rotations and entangling gates. A common structure is:

$$U_j(\boldsymbol{\theta}^{(j)}) = U_{\text{entangling}} \prod_{i=1}^{n} R_y(\theta_i^{(j)}) R_z(\phi_i^{(j)})$$

The rotation gates $R_y(\theta)$ and $R_z(\phi)$ are fundamental single-qubit operations rotating qubit states around the Y and Z axes of the Bloch sphere:

$$R_y(\theta) = \exp\left(-i\frac{\theta}{2}\sigma_y\right) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \tag{5.1}$$

$$R_z(\phi) = \exp\left(-i\frac{\phi}{2}\sigma_z\right) = \begin{pmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{i\phi/2} \end{pmatrix} \tag{5.2}$$

Entangling layers create quantum correlations between qubits, typically using CNOT gates between connected qubits. This combination generates rich quantum state varieties while remaining implementable on near-term devices.

Problem-inspired ansätze incorporate domain knowledge about problem structure. For quantum chemistry, the Unitary Coupled Cluster (UCC) ansatz builds on established methods by applying excitation operators moving electrons between orbitals. Hardware-efficient ansätze can be shallow and noise-resilient but may require many parameters for complex states. Problem-inspired ansätze often converge faster through domain knowledge but may require deeper circuits more susceptible to noise.

However, VQAs face fundamental challenges. The most serious is **barren plateaus**, where cost function gradients become exponentially small over large parameter space regions. When gradient variance scales as $\mathrm{Var}\left[\frac{\partial C}{\partial \theta_i}\right] \in O\left(\frac{1}{2^n}\right)$, gradient signals become indistinguishable from measurement noise, requiring exponentially many measurements for progress. Additional challenges include complex optimization landscapes with numerous local minima, shot noise from quantum measurements, and hardware imperfections creating spurious local minima.

The Variational Quantum Eigensolver (VQE), introduced by Peruzzo et al. in 2014 [30], exemplifies VQA application to eigenvalue problems. VQE determines ground state energies of many-body quantum systems by combining the variational principle with parameterized quantum circuits. The algorithm implements hybrid quantum-classical optimization leveraging the insight that while classical computers struggle with exponential scaling of quantum many-body problems, quantum hardware naturally represents and manipulates these states.

Hamiltonians decompose into Pauli operators:

$$\mathcal{H} = \sum_{i\alpha} h^i_\alpha \sigma^i_\alpha + \sum_{ij\alpha\beta} h^{ij}_{\alpha\beta} \sigma^i_\alpha \sigma^j_\beta + \dots$$

where h are real coefficients, Roman indices identify subsystems, and Greek indices identify Pauli operator types.

Exploiting observable linearity:

$$\langle \mathcal{H} \rangle = \sum_{i\alpha} h^i_\alpha \langle \sigma^i_\alpha \rangle + \sum_{ij\alpha\beta} h^{ij}_{\alpha\beta} \langle \sigma^i_\alpha \sigma^j_\beta \rangle + \dots$$

VQE handles diverse physical systems through polynomial-term Hamiltonians, including electronic structure Hamiltonians, quantum Ising models, Heisenberg models, and k-sparse Hamiltonians. Quantum devices efficiently compute expectation values for $2^n \times 2^n$ dimensional Hamiltonians, circumventing the computationally intractable N-representability problem (QMA-Hard [31]). The algorithm shows promising results for small molecular systems and continues developing with extensions for excited states, adaptive ansätze, and error mitigation techniques [32, 33].

Beyond VQE, several other VQAs address different problem classes. The Quantum Approximate Optimization Algorithm (QAOA) targets combinatorial optimization problems by alternating between problem and mixing Hamiltonians. Variational Quantum Classifiers (VQC) apply VQA principles to machine learning tasks, using quantum circuits as feature maps and cost functions based on classification accuracy. These algorithms share the fundamental VQA structure while adapting to specific problem requirements, demonstrating the framework's versatility across domains.

Traditional VQAs rely on gate-based quantum circuits, but alternative computational models offer potential advantages for NISQ devices. As discussed in the previous chapter on quantum computational models, Measurement-Based Quantum Computation (MBQC) represents one such alternative, using pre-prepared entangled states and adaptive measurements instead of gate sequences. This paradigm shift from gates to measurements opens new possibilities for variational algorithms, potentially addressing some limitations of circuit-based approaches through reduced circuit depth requirements and improved noise resilience. The combination of VQA principles with MBQC leads to Measurement-Based Variational Quantum Eigensolvers, which will be explored in detail in the following chapter.

Chapter 6

Measurement-Based Variational Quantum Eigensolvers

Building upon the variational quantum algorithm framework established in the previous chapter, this chapter explores an alternative implementation approach that leverages measurement-based quantum computation (MBQC) instead of traditional gate-based circuits. Measurement-Based Variational Quantum Eigensolvers (MB-VQE) represent a paradigm shift in how variational algorithms can be realized on quantum hardware, offering potential advantages for near-term devices while introducing new challenges related to computational determinism.

As discussed in Chapter 4, MBQC provides a fundamentally different approach to quantum computation compared to the gate-based circuit model. Instead of applying sequences of unitary gates to manipulate quantum states, MBQC leverages pre-prepared entangled resource states and performs computation through adaptive single-qubit measurements. This approach can be particularly advantageous for NISQ devices, as it moves the complexity from temporal gate sequences to spatial resource state preparation.

Ferguson et al. [9] first introduced Measurement-Based Variational Quantum Eigensolvers (MB-VQE), combining the variational optimization framework of VQE with the measurement-based computational paradigm. Traditional VQE uses unitary gates to manipulate quantum states, while MB-VQE uses measurements that consume auxiliary qubits to modify the remaining quantum state. This approach offers several potential advantages for near-term quantum devices and opens pathways to quantum states that may be expensive to prepare with conventional circuits.

The key insight behind MB-VQE is that many quantum computations can be more efficiently realized through measurements on pre-prepared entangled states rather than through sequential gate operations. This is particularly relevant for NISQ devices, where gate errors accumulate over time and circuit depth is severely limited. By moving computational complexity into the resource state preparation phase, MB-VQE can potentially achieve the same variational optimization objectives with reduced susceptibility to temporal noise sources.

6.1 Edge Decoration Approach

Ferguson et al. introduced two complementary approaches for implementing MB-VQE: edge decoration and direct circuit translation. Edge decoration provides a direct method for constructing variational state families within the measurement-based framework. This approach starts with an ansatz graph state $|\psi_a\rangle$ that approximates the target ground state of the Hamiltonian of interest. The variational flexibility is introduced by decorating the edges of this base graph with auxiliary qubits that are subsequently measured in parameterized bases.

For each pair of connected qubits (m, n) in the original ansatz graph, auxiliary qubits are added according to specific patterns. These auxiliary qubits are connected through controlled-Z gates and then measured in rotated bases $R(\theta) = \{(|0\rangle \pm e^{i\theta}|1\rangle)/\sqrt{2}\}$. The angles θ serve as variational parameters that can be optimized to minimize the energy expectation value.

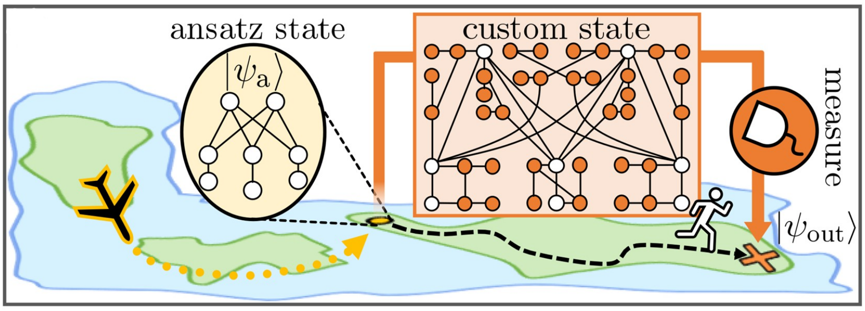


Figure 6.1: MB-VQE edge decoration approach [9]

Edge decoration creates a tailored entangled state that allows exploration of appropriate corners of the Hilbert space. The power of this decoration lies in its ability to induce complex entanglement modifications that don't correspond to simple unitary operations. A single auxiliary qubit connected to m output qubits can implement the operation $e^{i\theta/2\hat{Z}_1\otimes\hat{Z}_2\otimes\cdots\otimes\hat{Z}_m}$ through one measurement. The equivalent circuit implementation would require $O(m)$ two-qubit gates.

The direct translation method converts existing circuit-based VQE into equivalent MB-VQE by separating the original VQE circuit into Clifford gates and parametric single-qubit gates. Clifford gates can be performed classically according to the Gottesman-Knill theorem and are absorbed into the custom-state preparation, while parametric gates transform into measurements of auxiliary qubits in rotated bases.

MB-VQE offers several advantages over circuit-based VQE. Circuit depth reduction is the primary benefit, as traditional VQE requires gate sequences accumulating errors over time, while MB-VQE moves computation into resource state preparation, often performed more efficiently. The measurement phase requires only single-qubit operations, avoiding errors associated with long gate sequences. Resource allocation differs between approaches: circuit-based VQE trades time for space requiring sequential operations on the same qubits, while MB-VQE trades space for time using more qubits to reduce temporal complexity. Parallelization offers another advantage, as many MB-VQE measurements can be performed simultaneously, unlike gate sequences respecting causal ordering.

6.2 Determinism Challenges in Edge-Wise Decoration

However, Ferguson's edge decoration approach faces a critical limitation related to flow conditions—mathematical structures ensuring deterministic computation in MBQC despite measurement randomness. As established in Chapter 4, measurement-based quantum computing requires that the underlying graph state satisfies generalized flow (gflow) conditions to guarantee deterministic computation. When auxiliary qubits are added to graph edges sharing common vertices, the resulting graph structure can violate these conditions, breaking the deterministic guarantee fundamental to MBQC.

The problem manifests when decorating multiple edges that share common vertices. While single-edge decoration maintains valid gflow conditions, shared-vertex configurations create conflicts where byproduct corrections from one decoration interfere with measurements from another decoration. This leads to non-deterministic computation where the final quantum state depends on random measurement outcomes rather than the chosen variational parameters.

Consider a three-qubit linear graph state where vertex 2 participates in decorations for both edges 1-2 and 2-3. When measurement outcomes create Z byproducts that must be corrected through stabilizer relations, the correction pathways can conflict. A byproduct from measuring an auxiliary qubit in one decoration may require correction through a qubit that has already been measured in another decoration, violating the temporal ordering required by gflow conditions.

This determinism issue represents a fundamental theoretical limitation that prevents edge-wise decoration schemes from achieving the computational equivalence with circuit-based models that MBQC requires. The violation of gflow conditions means the algorithm cannot reliably reproduce the same quantum state across multiple runs with identical parameters, undermining the reliability essential for variational optimization.

6.3 Node-Wise Decoration Solution

Recent work by Schroeder et al. [10] addresses the determinism problem in MB-VQE using a scheme called node-wise decoration. This approach differs from edge-wise decoration by adding auxiliary qubits to vertices rather than edges of the graph state, creating layered structures that preserve the topological properties needed for flow conditions while maintaining variational capabilities.

The node-wise decoration scheme takes the original ansatz graph $G_0 = (V_0, E_0)$ and creates multiple identical copies arranged in layers. Each decoration layer G_ℓ maintains the same internal connectivity as G_0 plus vertical connections to layer $(\ell - 1)$. This construction ensures proper flow conditions: for any measured qubit in layer ℓ, the correction function maps to the corresponding qubit in layer $(\ell + 1)$.

The byproduct correction mechanism operates through stabilizer relations of the extended graph state. When measurements produce negative eigenstates, they introduce Z byproducts that can be commuted to the next layer using stabilizer equations. This systematic correction ensures deterministic computation while preserving polynomial overhead characteristics.

For linear ansatz graphs with n vertices, node-wise decoration with L layers creates $(L + 1) \times n$ rectangular cluster states requiring $(L + 1)n$ total qubits. Each decoration layer introduces n variational parameters, giving Ln total parameters for optimization. The dimensional expansion represents the price of determinism: one-dimensional ansatz graphs require two-dimensional cluster state resources, with direct implications for hardware implementation.

The comparison between edge-wise and node-wise approaches reveals different trade-offs. Edge-wise decoration maintains original dimensionality but faces exponential overhead due to postselection requirements when flow conditions are violated. Node-wise decoration guarantees polynomial overhead through systematic flow-based corrections but requires dimensional expansion. For practical implementations, this makes node-wise the currently viable approach for scalable deterministic MB-VQE.

6.4 Alternative Approaches and Future Directions

While node-wise decoration provides a working solution to the determinism problem, it comes with the cost of dimensional expansion that may limit its practical scalability. The requirement for higher-dimensional cluster states increases both the number of required qubits and the complexity of resource state preparation, potentially offsetting some of the advantages that MB-VQE offers over circuit-based approaches.

This limitation motivates continued research into alternative approaches that could maintain the benefits of edge-wise decoration while ensuring deterministic computation. Potential directions include modified decoration patterns that avoid shared-vertex conflicts, hybrid approaches combining elements of both edge-wise and node-wise schemes, and novel correction protocols that could restore determinism to edge-wise decorated graphs.

The development of deterministic MB-VQE algorithms represents a crucial step toward making measurement-based quantum computation practically viable for variational quantum algorithms. As quantum hardware continues to evolve, the relative advantages of measurement-based versus gate-based approaches may shift, making it essential to have robust implementations of both paradigms available for quantum algorithm developers.

The exploration of these measurement-based approaches represents an active frontier in quantum algorithm development, offering complementary strategies for extracting computational value from near-term quantum devices while working within current hardware constraints. The next chapter will analyze these determinism issues in detail and propose novel solutions to address the limitations of existing edge-wise decoration schemes.

Chapter 7

Analysis

In this chapter, we analyze the existing edge-decoration scheme for MB-VQE proposed by Ferguson et al. [9] in 2021. We first describe the scheme and its components, then identify the flow violation that hinders deterministic computation and develop a method to address this problem.

7.1 Edge-wise decoration scheme

Let us start by considering the edge-wise decoration scheme proposed in [9]. Consider two qubits m and n that are initially prepared in the $|+\rangle$ state. A controlled-Z gate is applied between them to create the edge, yielding $|G\rangle = \mathrm{CZ}|+\rangle_m|+\rangle_n$. This state $|G\rangle$ is a graph state and represents the simplest possible example of a graph state one can conceive, consisting of two nodes connected by a single edge.

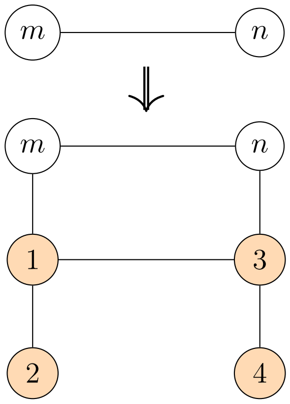


Figure 7.1: Edge-wise decoration applied to a single edge (m,n)

What the word *decoration* mean in this context? The decoration process involves adding auxiliary qubits to the original graph state. The decoration (see Figure 7.1) introduces four auxiliary qubits, labeled $i = 1, 2, 3, 4$ (orange circles), which are connected to the original output qubits m and n (white circles). All auxiliary qubits are initialized in the $|+\rangle$ state and connected through CZ gates, transforming our simple two-qubit graph state into a decorated structure of six qubits, that in [9] is called a *custom* state.

What is the role of the auxiliary qubits? Measuring auxiliary qubits in a specific order and in rotated bases $R(\theta)$ with specific angles directly controls the entanglement between qubits m and n.

Recalling that the rotated measurement basis is defined as:

$$R(\theta) \equiv \left\{ \frac{|0\rangle \pm e^{i\theta}|1\rangle}{\sqrt{2}} \right\}$$

we can consider two extremal cases:

Case 1: $\theta = 0$ (X-basis measurements) Measuring all auxiliary qubits with $\theta = 0$ (that is in the basis $R(0) = \{|+\rangle, |-\rangle\}$) allows to recover the original maximally entangled state, which is equivalent to applying no decoration at all:

$$\begin{aligned} \mathrm{CZ}_{mn}|+\rangle_m|+\rangle_n &= \frac{1}{2}(|00\rangle_{mn} + |01\rangle_{mn} + |10\rangle_{mn} - |11\rangle_{mn}) \\ &= |0\rangle_m|+\rangle_n + |1\rangle_m|-\rangle_n \end{aligned}$$

Case 2: $\theta = \pi/2$ (Y-basis measurements) Measuring all auxiliary qubits with $\theta = \pi/2$ (that is in the basis $R(\pi/2) = \{| + i\rangle, | - i\rangle\}$) completely disentangles the system, returning the separable state:

$$|+\rangle_m|+\rangle_n = \frac{1}{2}(|0\rangle_m + |1\rangle_m) \otimes (|0\rangle_n + |1\rangle_n)$$

By choosing intermediate values of θ, we can continuously interpolate between these two extremes, controlling the entanglement between qubits m and n.

To fully characterize this entanglement control, we analyzed the final quantum state $|\psi_{m,n}\rangle$ of the output qubits m and n after measuring all auxiliary qubits in rotated bases by arbitrary angles θ_i ($i = 1, \dots, 4$) in the XY plane.

The resulting state can be expressed in the computational basis as:

$$|\psi_{m,n}\rangle = C_{00}|0_m0_n\rangle + C_{01}|0_m1_n\rangle + C_{10}|1_m0_n\rangle + C_{11}|1_m1_n\rangle$$

The probability amplitudes are determined by the following analytical expressions:

$$
\begin{aligned}
C_{00} &= 1 + \cos\theta_2 \sin\theta_3 \sin\theta_4 + \cos\theta_4 \sin\theta_1 \sin\theta_2 + \cos\theta_1 \cos\theta_3 \sin\theta_2 \sin\theta_4 \\
C_{01} &= \frac{1}{2}(\cos^2\theta_4 + \cos\theta_4 - 1) + \sin\theta_1 \sin\theta_2 + i \sin\theta_4(\cos\theta_2 \cos\theta_3 - \cos\theta_1 \sin\theta_2 \sin\theta_3) \\
C_{10} &= \cos\theta_2 + \sin\theta_3 \sin\theta_4 + i \sin\theta_2(\cos\theta_1 \cos\theta_4 - \cos\theta_3 \sin\theta_1 \sin\theta_4) \\
C_{11} &= -\cos\theta_2 \cos\theta_4 - i \cos\theta_1 \sin\theta_2 + \sin\theta_4(\sin\theta_1 \sin\theta_2 \sin\theta_3 - i \cos\theta_3)
\end{aligned}
$$

These expressions represent unnormalized probability amplitudes for each computational basis state, with the physical quantum state obtained by normalization: $|\psi_{m,n}\rangle_{\text{norm}} = |\psi_{m,n}\rangle / \sqrt{\sum_{ij} |C_{ij}|^2}$.

7.2 Flow conditions in decoration schemes

A natural question arises: can we identify a valid gflow on this decorated graph to ensure deterministic computation, as explained in the subsection 4.2.4?

Examining the decorated graph in Figure 7.1, we need to establish both correction sets and a measurement order for the four auxiliary qubits. The answer is yes—a valid gflow exists, as shown in Table 7.1.

Measured Qubit	**Correction Set**
2	$\{1\}$
4	$\{3\}$
1	$\{m\}$
3	$\{n\}$

Table 7.1: Correction sets for single-edge decoration

The measurement order is: $2, 4 \prec 1, 3$, where qubits 2 and 4 are measured first (in any order), followed by qubits 1 and 3 (in any order).

This confirms that the single-edge decoration operates deterministically. In other words, we can measure the auxiliary qubits and guarantee that the final output state is independent of the random measurement outcomes—the hallmark of deterministic MBQC. However, the situation becomes significantly different when we extend this approach to larger graphs where edges share common vertices.

Let's consider the same decoration in 7.1. Instead of a single edge, we now consider two edges that share a common vertex. Our initial ansatz graph state consists of qubits 1, 2, and 3 connected by entangling gates, yielding $|\mathrm{G}\rangle = \mathrm{CZ}_{12}\mathrm{CZ}_{23}|+\rangle_1|+\rangle_2|+\rangle_3$. This is the graph state we will decorate.

As shown in Figure 7.2, vertex 2 participates in decorations for both edges 1-2 and 2-3. This means that each edge gets its own decoration gadget (four auxiliary qubits each), but they must interact through the shared vertex 2.

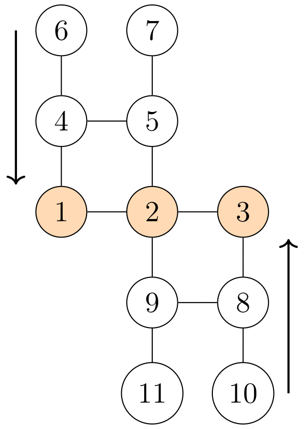


Figure 7.2: Output qubits 1–2–3 (orange) with decorations on edges 1-2 and 2-3 sharing vertex 2. The upper gadget (qubits 4,5,6,7) decorates edge 1-2, while the lower gadget (qubits 8,9,10,11) decorates edge 2-3.

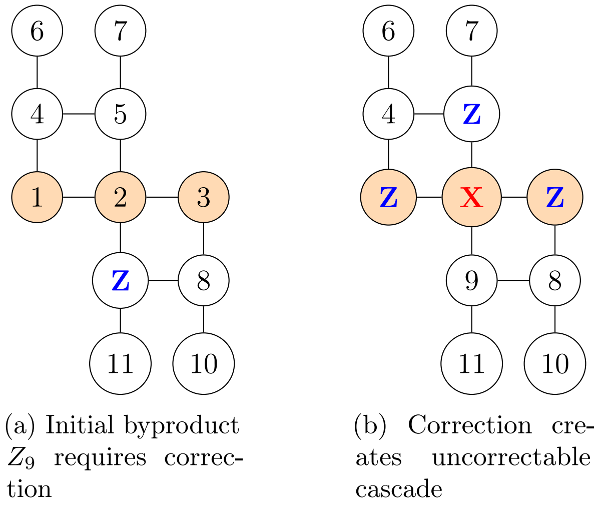
We maintain the same measurement order of the auxiliary qubits as in the single-edge case. In particular, as shown in Figure 7.2 (black arrows), qubits 6, 7, 11 and 10 are measured first, then qubits 4, 5, 9 and 8 are measured.

7.2.1 The gflow violation

Here is what goes wrong: as introduced in the theoretical framework (4.2.4), when measuring auxiliary qubits, unfavorable outcomes ($s_i = 1$, corresponding to projecting onto $|\theta^-\rangle$) generate Z byproducts that must be corrected through the stabilizer structure.

Suppose, looking at Figure 7.3, that a measurement on qubit 9 creates byproduct Z_9. To correct this, we must use a stabilizer of the graph state that contains Z_9. In our decorated graph, the relevant stabilizer is $X_2Z_1Z_3Z_5Z_9$. Applying this stabilizer eliminates Z_9 but creates new byproducts on qubits 1, 3, and 5.

The crucial problem is that byproduct Z_5 (in the upper gadget) cannot be corrected without affecting already-processed measurements. According to the gflow conditions, we would need qubit 7 to be available for correction (since the stabilizer for qubit 5 involves qubit 7), but qubit 7 was already measured before qubit 5, violating the temporal ordering required by gflow. This creates a cascade of corrections that cannot be resolved within the prescribed measurement order.



(a) Initial byproduct Z_9 requires correction

(b) Correction creates uncorrectable cascade

Figure 7.3: Stabilizer correction creates byproduct Z_5 that cannot be corrected due to measurement order constraints.

To be more precise mathematically, let us demonstrate this failure of determinism by examining all possible measurement outcomes for qubits 5 and 9.

Denote $|G\rangle$ the initial resource state in Figure 7.2. After measuring qubits 5 and 9, we project onto the outcomes $|\theta^{\pm}\rangle_5$ and $|\theta^{\pm}\rangle_9$. These are defined as:

$$|\theta^{+}\rangle_i = \frac{1}{\sqrt{2}}(|0\rangle_i + e^{i\theta}|1\rangle_i) \quad \text{(measurement outcome “0”)} \tag{7.1}$$

$$|\theta^{-}\rangle_i = \frac{1}{\sqrt{2}}(|0\rangle_i - e^{i\theta}|1\rangle_i) \quad \text{(measurement outcome “1”)} \tag{7.2}$$

where i denotes the qubit index. The key point is that outcome “1” introduces a Z byproduct that must be corrected using stabilizer relations.

For outcomes $(0,0)$ and $(1,1)$ on qubits $(9,5)$:

$$\begin{aligned}
\langle\theta^+|_9\langle\theta^+|_5|G\rangle &= \langle\theta^+|_9\langle\theta^+|_5|G\rangle \qquad (7.3)\\
\langle\theta^-|_9\langle\theta^-|_5|G\rangle &= \langle\theta^+|_9\langle\theta^+|_5 Z_9 Z_5|G\rangle\\
&= \langle\theta^+|_9\langle\theta^+|_5 X_2 Z_1 Z_3|G\rangle\\
&= \langle\theta^+|_9\langle\theta^+|_5|G\rangle \qquad (7.4)
\end{aligned}$$

For outcomes $(1,0)$ and $(0,1)$ on qubits $(9,5)$:

$$\begin{aligned}
\langle\theta^-|_9\langle\theta^+|_5|G\rangle &= \langle\theta^+|_9\langle\theta^+|_5 Z_9|G\rangle\\
&= \langle\theta^+|_9\langle\theta^-|_5 X_2 Z_1 Z_3|G\rangle \qquad (7.5)\\
\langle\theta^+|_9\langle\theta^-|_5|G\rangle &= \langle\theta^-|_9\langle\theta^+|_5 X_2 Z_1 Z_3|G\rangle \qquad (7.6)
\end{aligned}$$

The results show that outcomes $(0,0)$ and $(1,1)$ yield the same final state, while outcomes $(1,0)$ and $(0,1)$ yield a different final state. Since the final quantum state depends on the specific measurement outcomes, the computation is no longer deterministic—this is the mathematical signature of gflow violation.

This limitation of the edge-decoration scheme motivates the need for a new approach that can handle shared vertices while maintaining deterministic computation.

Just to verify this limitation and confirm that no alternative measurement ordering could admit a valid gflow, we used the `graphix` library. This library is particularly interesting and useful concerning computations in MBQC. In particular, there is a function that constructs measurement patterns from graph specifications which is called `generate_from_graph` and automatically searches for gflow assignments. Testing the configuration in Figure 7.2 confirms that no valid gflow exists for this shared-vertex decoration scheme. The small code is provided in Appendix A.2.

7.2.2 Node-wise decoration scheme

Now that we have a better understanding of the edge decoration scheme and its limitations, it is worth noting that this same problem has already been observed in the literature by [10], who addressed it with a technique called the *node-wise decoration scheme.*

Let us examine this node-wise decoration approach to understand why the solution proposed in [10] successfully preserves the gflow condition.

The node-wise decoration scheme is illustrated in Figure 7.4. In this approach, the decoration is organized in layers, where the first digit of each white node represents its layer (e.g., nodes 10-12 belong to layer 1, nodes 20-22 to layer 2). The measurement order proceeds from bottom to top, as indicated by the arrow.

Remarkably, this scheme addresses the same three-qubit linear graph as in Figure 7.2, but the shared vertex in the middle (qubit 1) no longer causes gflow violations. The key insight is that the layered structure ensures that any Z byproduct can always be corrected using stabilizers from qubits in subsequent measurement layers, creating a systematic correction pathway that maintains determinism.

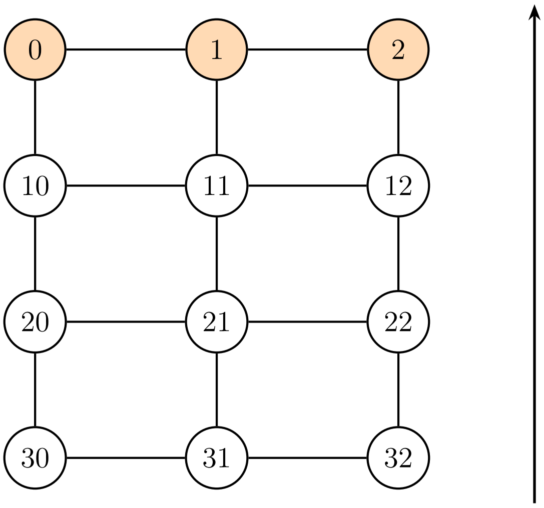
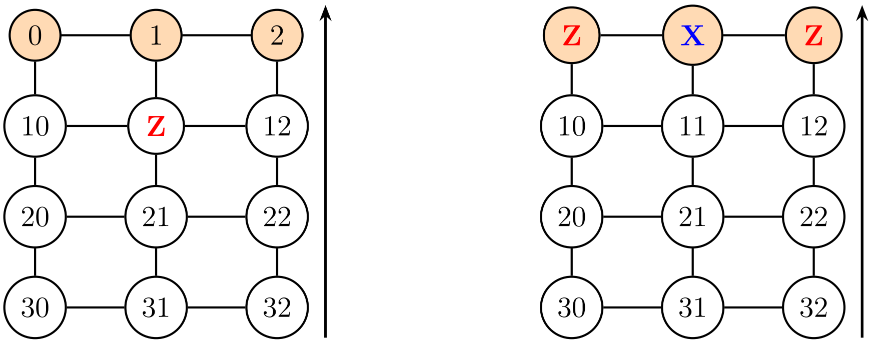


Figure 7.4: Node-wise decoration with 3 layers. Orange nodes are physical output qubits, white nodes are decoration qubits measured from bottom to top.

Figure 7.5 illustrates how byproduct corrections work in the node-wise scheme. When a Z byproduct appears on qubit 11 (left panel), it can be corrected by applying the appropriate stabilizer that involves qubit 1 in the next layer (right panel). This correction mechanism always succeeds because the layered structure ensures that correction qubits are always available in subsequent layers.

To further illustrate the robustness of the layered approach, Figure 7.6 shows how byproducts appearing in deeper layers can be systematically corrected. Even when a Z byproduct appears on qubit 21 in layer 2, it can be corrected using the stabilizer of qubit 11 in layer 1, which then



(a) Z byproduct on qubit 11

(b) After correction using stabilizer of qubit 1

Figure 7.5: Byproduct propagation in node-wise decoration: a Z byproduct on qubit 11 is corrected by applying the stabilizer of qubit 1 in the neighboring layer

propagates the correction upward to the physical qubits where it can be handled appropriately.

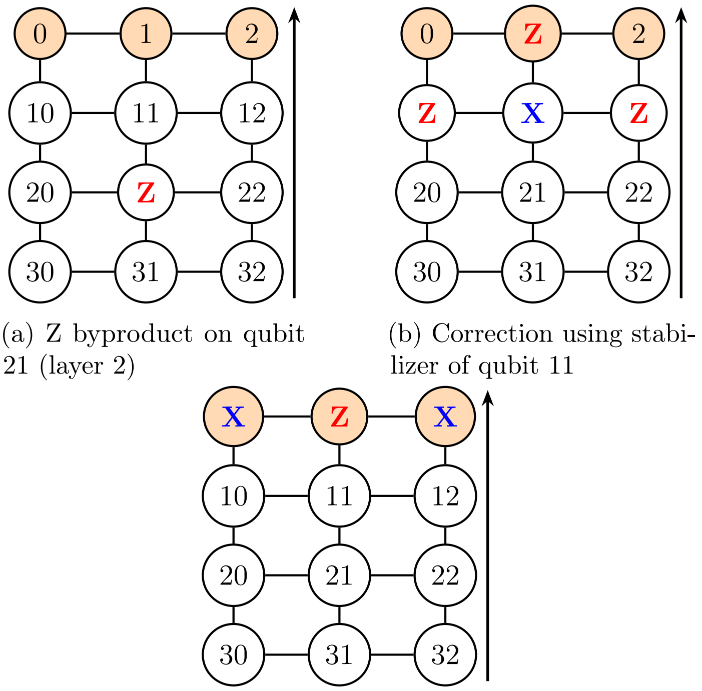
In general, each Z byproduct is corrected using the stabilizer of the corresponding qubit in the subsequent measurement layer. This systematic approach ensures that all byproducts can be handled deterministically, regardless of which layer they originate from. This is why the layered approach proposed by [10] is very promising.

However, there is one possible limitation: this decoration scheme differs from the edge-wise one: the edge-wise scheme maintains a planar graph, whereas the node-wise scheme requires an additional dimension for the graph. For example, in Figure 7.2 the graph is linear, demanding the preparation of a regular 2D cluster state as the resource state. For a 2D graph, a three-dimensional cluster state becomes necessary, which may pose challenges for experimental implementation.

In Appendix A.3 we provide a code that verifies determinism in this approach. The implementation confirms that states remain equivalent across multiple runs of the simulations. At each run the measurement outcomes are obviously different since they are probabilistic, whereas the output state is always the same validating the gflow properties.

7.2.3 Toward a solution for gflow violation

Having analyzed these two decoration schemes and their respective limitations, we can now present our approach to solving the gflow violation problem in



(a) Z byproduct on qubit 21 (layer 2)

(b) Correction using stabilizer of qubit 11

(c) Final correction using stabilizers of qubits 0 and 2

Figure 7.6: Cascade correction in node-wise decoration: a Z byproduct in layer 2 is systematically corrected by propagating through the layers

edge-wise decoration.

Our initial solution was to apply the decorations sequentially, processing each edge pattern separately while maintaining the original distinction between qubit types: auxiliary qubits remain the measured qubits, and the output qubits are always the qubits of the initial ansatz graph being decorated.

However, as we will show, this sequential decoration approach based on the decomposition of masurement patterns proved to be theoretically inadequate.

This limitation motivated the development a different approach: the proposed solution consists of dynamically updating the input and output qubit assignments during the computation. Both the procedure will be explained in the following sections.

7.3 Pattern decomposition approach

To address the gflow violation problem in edge-wise decoration, we first considered a sequential approach: instead of decorating all edges simultaneously, we decorate them one by one to avoid shared-vertex conflicts.

The idea is simple. Given a graph $G = (V, E)$ in initial state $|\psi_{\text{in}}\rangle$, we:

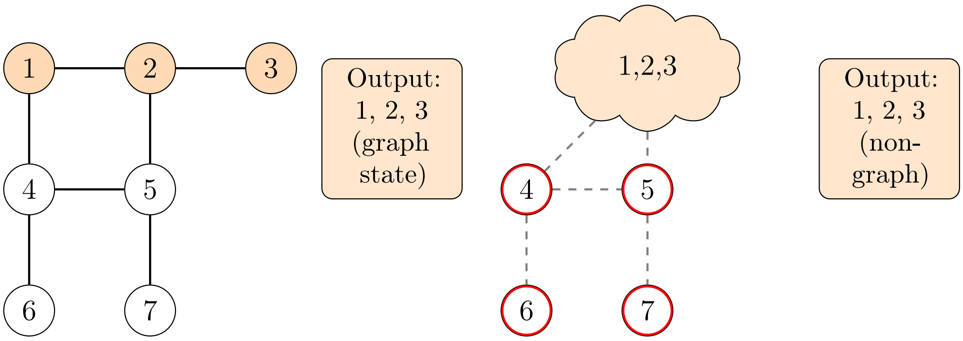
1. Select edge $(a, b) \in E$ and apply the decorative gadget
2. Measure the auxiliary qubits according to the decoration pattern
3. Use the resulting state as input for the next decoration
4. Repeat until all edges are decorated

This can be made more efficient using graph coloring: edges of the same color share no vertices, so we can process all edges of one color simultaneously before moving to the next color.

However, this approach has a problem: after the first decoration, the output qubits are no longer in a graph state.

When we measure auxiliary qubits, their random outcomes affect the output qubits in a measurement-dependent way. This destroys the stabilizer structure that defines graph states. The output qubits become an arbitrary entangled state that cannot be described by stabilizer generators.

This is illustrated in Figure 7.7. Before measurement, qubits 1, 2, 3 form a graph state with well-defined stabilizers. After measuring auxiliary qubits 4, 5, 6, 7, these qubits (1,2 and 3) are in a complex entangled state (shown as a cloud) that no longer has the graph state structure.

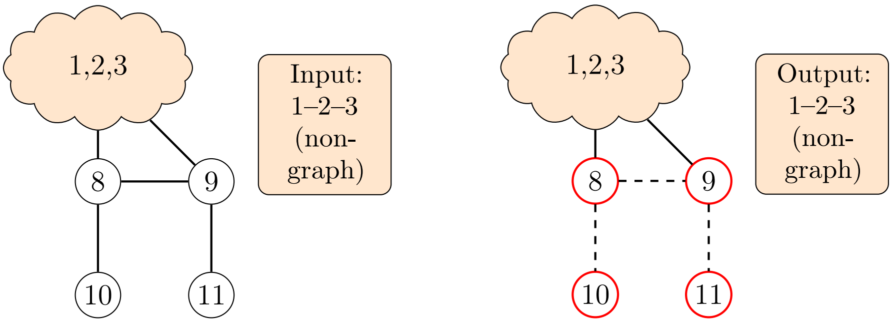


(a) Before measurement

(b) After measurement

Figure 7.7: First stage of pattern decomposition showing transition from graph state to non-graph state after partial measurements.

The problem becomes clear when we try to apply the second decoration in Figure 7.8. The decoration pattern assumes the input is a graph state, but our input is now an arbitrary entangled state.



(a) Before measurement

(b) After measurement

Figure 7.8: Second stage of pattern decomposition attempting to process non-graph state input

The absence of stabilizers is the core issue. In graph states, each vertex has a well-defined stabilizer operator that enables byproduct correction. For example, in the initial graph state of Figure 7.7, qubit 2 has stabilizer $X_2Z_1Z_3$. If a measurement produces a Z byproduct on qubit 5, we can apply this stabilizer to eliminate this byproduct.

However, once we measure the auxiliary qubits and transition to the non-graph state shown in the cloud, these stabilizers no longer exist. The state cannot be characterized by simple local stabilizer operators. If we need to correct a Z byproduct on qubit 8 during the second decoration, we have no stabilizer operator available to perform the correction.

This led us to develop a different approach that maintains the graph state structure throughout the entire decoration process by dynamically reassigning input and output qubits.

7.4 Edge decoration with dynamic qubit relabeling

Since the first approach could not be implemented, we now analyze a second possible strategy to address the gflow problem in the edge decoration scheme.

The main idea is simple: instead of keeping fixed input and output qubits, we dynamically update the qubit register at each decoration step. We start with an initial qubit register V prepared in a stabilizer state, for example a graph state $|G\rangle$. At each step, we select two qubits from the register as inputs for the decoration. We apply the decoration gadget, measure the input qubits and some ancilla qubits, and keep the remaining ancilla qubits as new outputs that replace the inputs in the register.

This dynamic relabeling allows us to maintain the stabilizer structure throughout the process. The selected input qubits can form an edge in the initial graph or be isolated vertices—the procedure works in both cases.

The step-by-step procedure is as follows:

Initial condition: At step $k = 0$, we start with an initial stabilizer state $|\psi_0\rangle$ defined on qubit register V_0.

1. **Setup and input selection:** At step $k \geq 1$, we have a quantum state $|\psi_k\rangle$ defined on qubit register V_k. We select any two qubits $a_k, b_k \in V_k$ and designate them as input qubits for the current decoration.

2. **Decoration and measurement:** We introduce four ancillary qubits $A_k = \{a_1^{(k)}, a_2^{(k)}, a_3^{(k)}, a_4^{(k)}\}$, each initialized in $|+\rangle$. A local Clifford unitary U_k entangles the input qubits with all ancillae through CZ and Hadamard gates. We then perform XY-plane measurements on the input qubits a_k, b_k and the first two ancillae $a_1^{(k)}, a_2^{(k)}$, recording outcomes as $m_k = (m_{a_k}, m_{b_k}, m_{a_1^{(k)}}, m_{a_2^{(k)}})$.

3. **Output correction and relabeling:** The unmeasured ancillae $a_3^{(k)}$ and $a_4^{(k)}$ become our output qubits. We apply Pauli corrections based on measurement outcomes and update the register as $V_k = (V_k \setminus \{a_k, b_k\}) \cup \{a_3^{(k)}, a_4^{(k)}\}$. These outputs are available for selection in subsequent decoration steps.

7.4.1 Example: 3-qubit Linear Graph State

We illustrate the dynamic qubit relabeling procedure on a simple three-qubit linear graph state. We begin with $V_0 = \{1, 2, 3\}$ and initial state

$$|\psi_0\rangle = CZ_{12}\, CZ_{23}\, |+\rangle^{\otimes 3},$$

representing edges 1–2 and 2–3.

Decoration step 1: We select qubits 1 and 2 as input qubits for our first decoration. We add four ancillae $A_1 = \{a_1^{(1)}, a_1^{(2)}, a_1^{(3)}, a_1^{(4)}\}$, all initialized in $|+\rangle$, and create the decorative gadget by introducing entanglement between

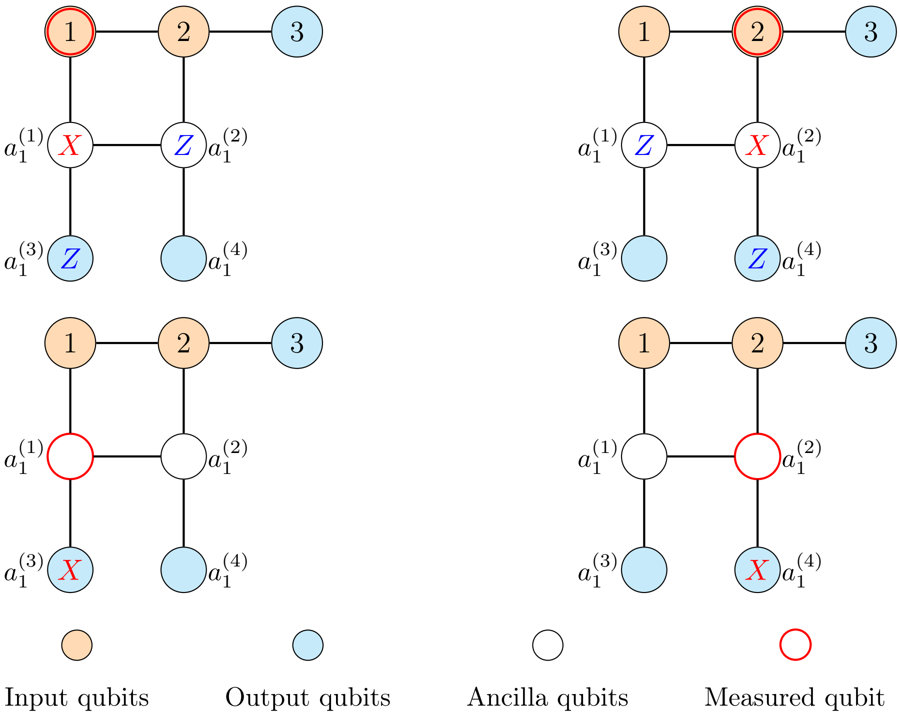


Figure 7.9: Sequential measurement and byproduct correction pattern for decoration step 1.

the ancillary qubits and connecting qubits 1 and 2 to the gadget structure. We then perform measurements on qubits 1, 2, $a_1^{(1)}$, and $a_1^{(2)}$ in the XY-plane.

The key insight is that Pauli corrections are applied using only the stabilizers of the decorative gadget structure. For byproducts on the measured qubits, we use the following stabilizers:

- Byproduct on qubit 1: $X_{a_1^{(1)}} Z_{a_1^{(3)}} Z_{a_1^{(2)}} Z_1$
- Byproduct on qubit 2: $X_{a_1^{(2)}} Z_{a_1^{(4)}} Z_{a_1^{(1)}} Z_2$
- Byproduct on $a_1^{(1)}$: $X_{a_1^{(3)}} Z_{a_1^{(1)}}$
- Byproduct on $a_1^{(2)}$: $X_{a_1^{(4)}} Z_{a_1^{(2)}}$

Finally, we relabel the two output qubits as $o_1 := a_1^{(3)}$ and $o_2 := a_1^{(4)}$, updating the register to $V_1 = \{o_1, o_2, 3\}$.

Decoration step 2: We now select o_2 (an output from step 1) together with qubit 3 as our new input pair. The gray box represents the non-graph

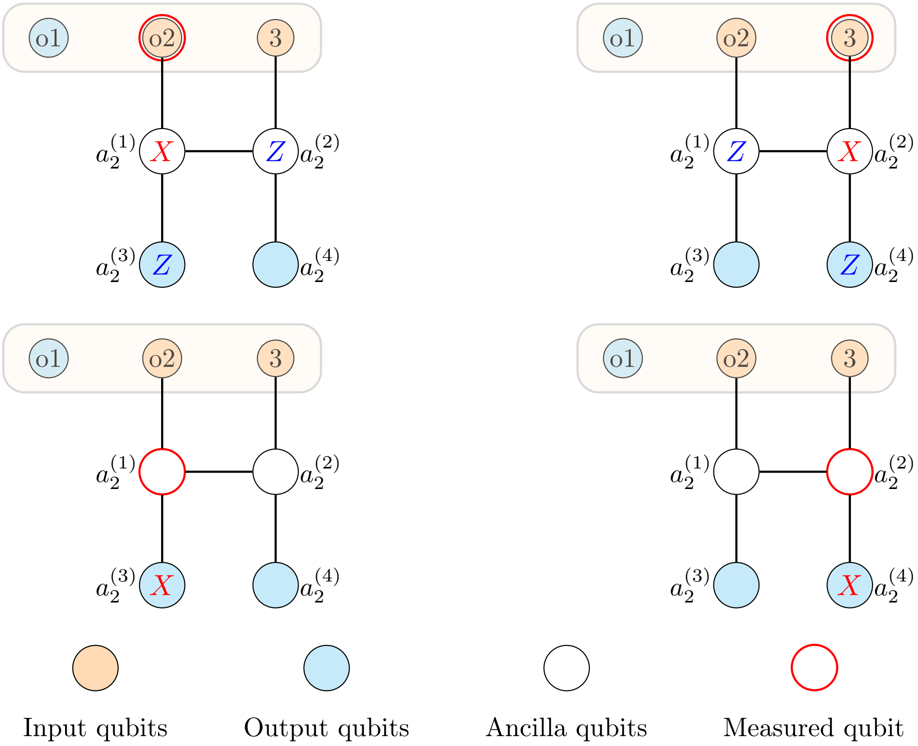


Figure 7.10: Sequential measurement and byproduct correction for decoration step 2.

state resulting from the previous decoration. Crucially, we do not require these qubits to form an edge in a graph state—they simply need to be part of the current quantum state.

We add new ancillae $A_2 = \{a_2^{(1)}, a_2^{(2)}, a_2^{(3)}, a_2^{(4)}\}$ and create the gadget structure by entangling them. After performing measurements on o_2, 3, $a_2^{(1)}$, and $a_2^{(2)}$ in the XY-plane, we apply Pauli corrections using the stabilizers from the newly created decoration gadget:

- Byproduct on o_2: $X_{a_2^{(1)}} Z_{a_2^{(3)}} Z_{a_2^{(2)}} Z_{o_2}$
- Byproduct on qubit 3: $X_{a_2^{(2)}} Z_{a_2^{(4)}} Z_{a_2^{(1)}} Z_3$
- Byproduct on $a_2^{(1)}$: $X_{a_2^{(3)}} Z_{a_2^{(1)}}$
- Byproduct on $a_2^{(2)}$: $X_{a_2^{(4)}} Z_{a_2^{(2)}}$

We then relabel the output qubits as $o_3 := a_2^{(3)}$ and $o_4 := a_2^{(4)}$, resulting in the final register $V_2 = \{o_1, o_3, o_4\}$.

This procedure can be extended to decorate all edges in the original graph. While this example involves only two steps, the technique can be applied systematically across any graph topology.

7.5 Analysis of Determinism in dynamic qubit relabeling

The dynamic qubit relabeling scheme guarantees deterministic computation through a key insight: determinism in measurement-based quantum computing does not require the input qubits to maintain specific graph state structures, but rather depends on the local availability of correction stabilizers within each decoration step.

Consider the k-th decoration step applied to input qubits (a_k, b_k) with auxiliary qubits $A_k = \{a_1^{(k)}, a_2^{(k)}, a_3^{(k)}, a_4^{(k)}\}$. The decoration operation creates a local stabilizer group S_k through the Clifford unitary U_k that entangles the input qubits with the auxiliary qubits according to the gadget structure. This stabilizer group is completely determined by the local decoration topology and is independent of the global structure of the input state $|\psi_k\rangle$.

A crucial difference from standard edge-wise decoration is that we measure not only the auxiliary qubits but also the two input qubits (a_k, b_k) that were originally part of the ansatz graph or outputs from previous decorations. This measurement strategy, combined with dynamic relabeling, is what enables deterministic computation.

The stabilizer generators for the decoration gadget are:

$$\begin{aligned} S_k = \langle & X_{a_1^{(k)}} Z_{a_k} Z_{a_3^{(k)}} Z_{a_2^{(k)}}, && (7.7)\\ & X_{a_2^{(k)}} Z_{b_k} Z_{a_4^{(k)}} Z_{a_1^{(k)}}, && (7.8)\\ & X_{a_3^{(k)}} Z_{a_1^{(k)}}, && (7.9)\\ & X_{a_4^{(k)}} Z_{a_2^{(k)}} \rangle && (7.10) \end{aligned}$$

Each measurement outcome $m_i \in \{0, 1\}$ on a measured qubit i potentially introduces a Z byproduct $Z_i^{m_i}$ that must be corrected. For each measured qubit—including both input qubits and auxiliary qubits—there exists one stabilizer generator in S_k that contains the corresponding Z operator, ensuring a correction path. For instance, if measuring auxiliary qubit $a_1^{(k)}$ yields outcome $m_1 = 1$, the resulting Z byproduct $Z_{a_1^{(k)}}$ can be corrected using the stabilizer $X_{a_3^{(k)}} Z_{a_1^{(k)}}$, which transforms the byproduct into an X correction on the output qubit $a_3^{(k)}$.

This approach differs from the problematic shared-vertex configurations in standard edge-wise decoration, where byproducts from one decoration can propagate to qubits that are simultaneously involved in other decorations, creating conflicts where correction stabilizers are not available.

By measuring the input qubits and dynamically reassigning the output qubits, we maintain deterministic computation while avoiding the gflow violations that plague the original edge-decoration scheme of [9].

There are some advantages of this procedure, in particular:

- **Modularity:** Each decoration acts locally and independently. Decorations can be applied in any order to any pair of qubits, enabling step-by-step construction of patterns.
- **Efficient tracking:** Since each step involves only a few qubits, measurement results and corrections are easy to track.
- **Circuit-like structure:** Each decoration gadget acts as a two-qubit quantum gate. We apply these "gates" sequentially to different qubit pairs, mirroring the structure of gate-based quantum circuits, as shown in Figure 7.11.
- **Generality:** The method applies to any local gadget operating on qubit pairs with ancillae, so there is the possibility to generalize the decorative gadget and choose more suitable ones.

The qubit relabeling approach can be visualized as a sequential cascade of decoration gadgets viewed as quantum 2 qubit gates. Figure 7.11 illustrates this process.

7.6 MB-VQE Simulation

We tested numerically the approach developed in section 7.4 using perturbed stabilizer Hamiltonians as test cases. Stabilizer Hamiltonians provide ideal benchmarks for validating our deterministic MB-VQE approach because: (1) their unperturbed ground states are exactly known graph states, providing natural ansätze for our decoration scheme; (2) small perturbations destroy the stabilizer structure, requiring variational approximation; and (3) for

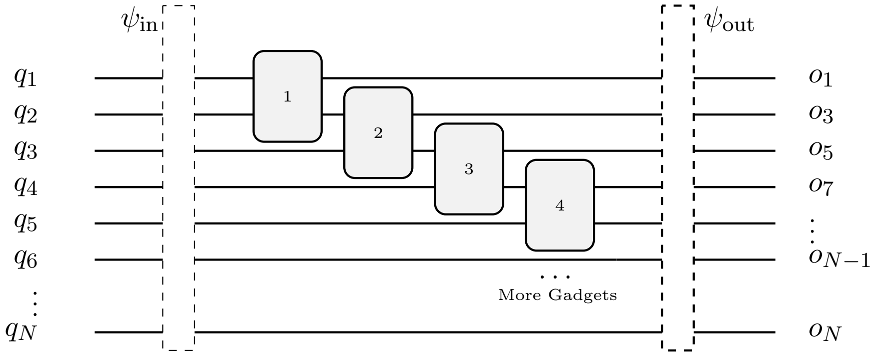


Figure 7.11: Diagrammatic representation of sequential gadget decorations with dynamic update of the role of the qubit for N qubits. Starting with N input qubits $q_1, q_2, \ldots, q_N$, each gadget connects one output from the previous decoration to the next original ansatz input qubit.

small number of qubits, exact solutions remain computationally tractable for validation against our variational results.

We tested the dynamic qubit relabeling approach using as the first target Hamiltonian the perturbed 8-qubit Toric Code Hamiltonian.

Recall that unperturbed 8-qubit Toric Code Hamiltonian $\hat{H}_0$ (described in 3.2.3) is:

$$\hat{H}_0 = -\sum_s \hat{A}_s - \sum_p \hat{B}_p \tag{7.11}$$

Let us now consider a perturbation of this Hamiltonian:

$$\hat{H} = \hat{H}_0 + \lambda \hat{H}_{\text{pert}} \tag{7.12}$$

The perturbative Hamiltonian takes the form of a homogeneous longitudinal magnetic field:

$$\hat{H}_{\text{pert}} = -\lambda \sum_{i=1}^{8} \hat{Z}_i \tag{7.13}$$

This term acts along the computational basis $\hat{Z}$-direction, directly favoring the $|1\rangle$ state for each qubit.

The total Hamiltonian interpolates between two distinct physical regimes:

- For $\lambda = 0$: The system is governed purely by the toric code Hamiltonian $\hat{H}_0$, and the ground-state manifold is four-fold degenerate: $\{|00\rangle_L, |01\rangle_L, |10\rangle_L, |11\rangle_L\}$
- For $\lambda \gg 1$: The perturbative term dominates and the ground state

becomes the completely separable state $|\psi_{\text{GS}}\rangle = |11\dots11\rangle$

The key insight of our MB-VQE approach is to use different initial graph states depending on the perturbation strength. We identify a threshold λ_t that determines which graph state to use as the initial ansatz:

- For $\lambda < \lambda_t$: We start with the graph state corresponding to the unperturbed ground state $|0,0\rangle_L$
- For $\lambda > \lambda_t$: We use isolated vertices representing the separable state $|1\rangle^{\otimes 8}$

Implementation Details: For the 8-qubit toric code, we apply the dynamic qubit relabeling scheme by sequentially decorating the graph state representation of $|0,0\rangle_L$. The graph state is first extracted using the *stim* library to convert stabilizer operators into the corresponding graph representation. We then apply edge decorations using our dynamic qubit relabeling protocol, where each decoration step processes a pair of qubits and dynamically updates the qubit register. The final ansatz uses 4 variational parameters per decoration step plus 3 single-qubit rotation parameters ($R_x(\theta)$, $R_y(\phi)$, and $R_z(\gamma)$) applied to each output qubit to increase expressivity.

For all our numerical simulations, we evaluate both energy accuracy and state fidelity. To quantify how well our MB-VQE states approximate the true ground states, we use the state fidelity:

$$F = |\langle\psi_{\text{exact}}|\psi_{\text{MB-VQE}}\rangle|^2 \tag{7.14}$$

The fidelity measures the overlap between the exact ground state $|\psi_{\text{exact}}\rangle$ and our optimized ansatz state $|\psi_{\text{MB-VQE}}\rangle$, with $F = 1$ indicating perfect agreement and $F = 0$ indicating orthogonal states. For practical analysis, we often use the infidelity:

$$\text{Infidelity} = 1 - F = 1 - |\langle\psi_{\text{exact}}|\psi_{\text{MB-VQE}}\rangle|^2 \tag{7.15}$$

which directly measures the "distance" between the quantum states, with smaller values indicating better state preparation.

Figure 7.12 shows the initial ansatz graph for our decoration scheme in the weak perturbation regime.

Figure 7.13 demonstrates why the adaptive threshold strategy is essential for achieving good performance across different perturbation regimes.

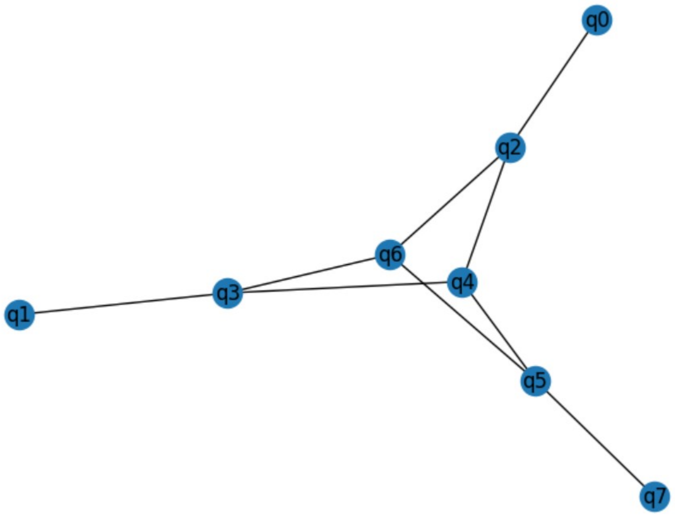


Figure 7.12: Graph state corresponding to the logical state $|0,0\rangle_L$ of the 2×2 toric code. This graph serves as the initial ansatz for our dynamic qubit relabeling decoration scheme in the weak perturbation regime ($\lambda < \lambda_t$). Each vertex represents a physical qubit, and edges represent entangling CZ gates in the graph state preparation.

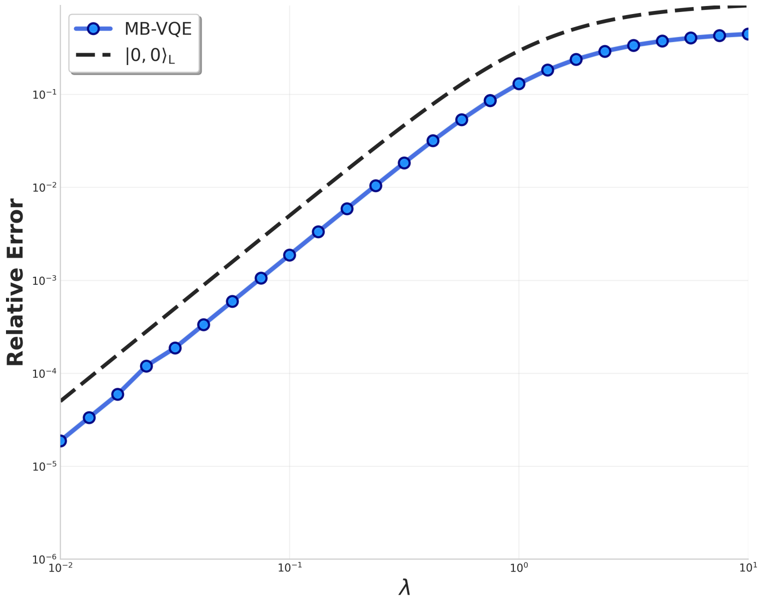


Figure 7.13: MB-VQE performance without threshold strategy, showing the necessity of adaptive ansatz selection. The blue curve shows poor performance for large λ values when using only the $|0,0\rangle_L$ graph state initialization throughout the entire range. The performance plateaus for $\lambda \gtrsim 0.9$ because the entangled ansatz cannot capture the separable nature of the ground state in the strong perturbation regime.

We systematically tested threshold values between 0.5 and 2.0, evaluating the performance by computing the state fidelity for each λ. The optimal threshold $\lambda_t = 0.9$ minimizes the maximum infidelity across the entire perturbation range.

We test MB-VQE on the 2×2 toric code with 25 λ values from 0.01 to 10 (logarithmic spacing). For each λ, we run 3 optimization attempts using L-BFGS-B and report the best result. The implementation uses the dynamic qubit relabeling scheme with deterministic measurement outcome correction.

Energy Performance

Figure 7.14 compares the relative energy errors of three different approaches across the full perturbation range.

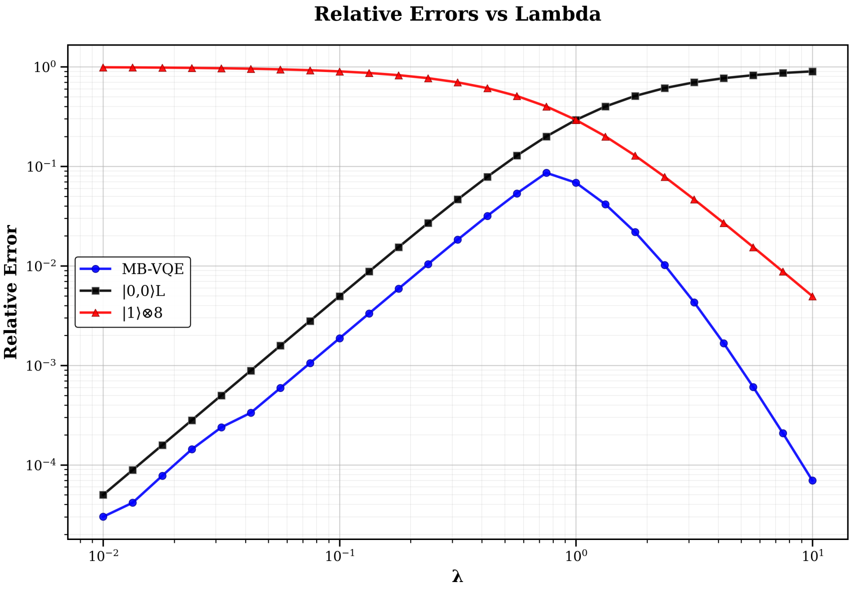


Figure 7.14: Relative energy errors vs perturbation strength for the 2×2 toric code, demonstrating the effectiveness of adaptive threshold strategy. The blue curve shows our MB-VQE approach with adaptive threshold, which consistently outperforms both baselines: the unperturbed logical state $|0,0\rangle_L$ (black curve) and the separable state $|1\rangle^{\otimes 8}$ (red curve).

The blue curve represents our MB-VQE performance:

$$\varepsilon_{\text{MB-VQE}} = \frac{|E_{\text{MB-VQE}}(\lambda) - E_{\text{exact}}(\lambda)|}{|E_{\text{exact}}(\lambda)|} \tag{7.16}$$

To demonstrate the effectiveness of our approach, we compare against two static baselines. The black curve shows the performance when using only the unperturbed logical state $|0,0\rangle_L$ for all λ values:

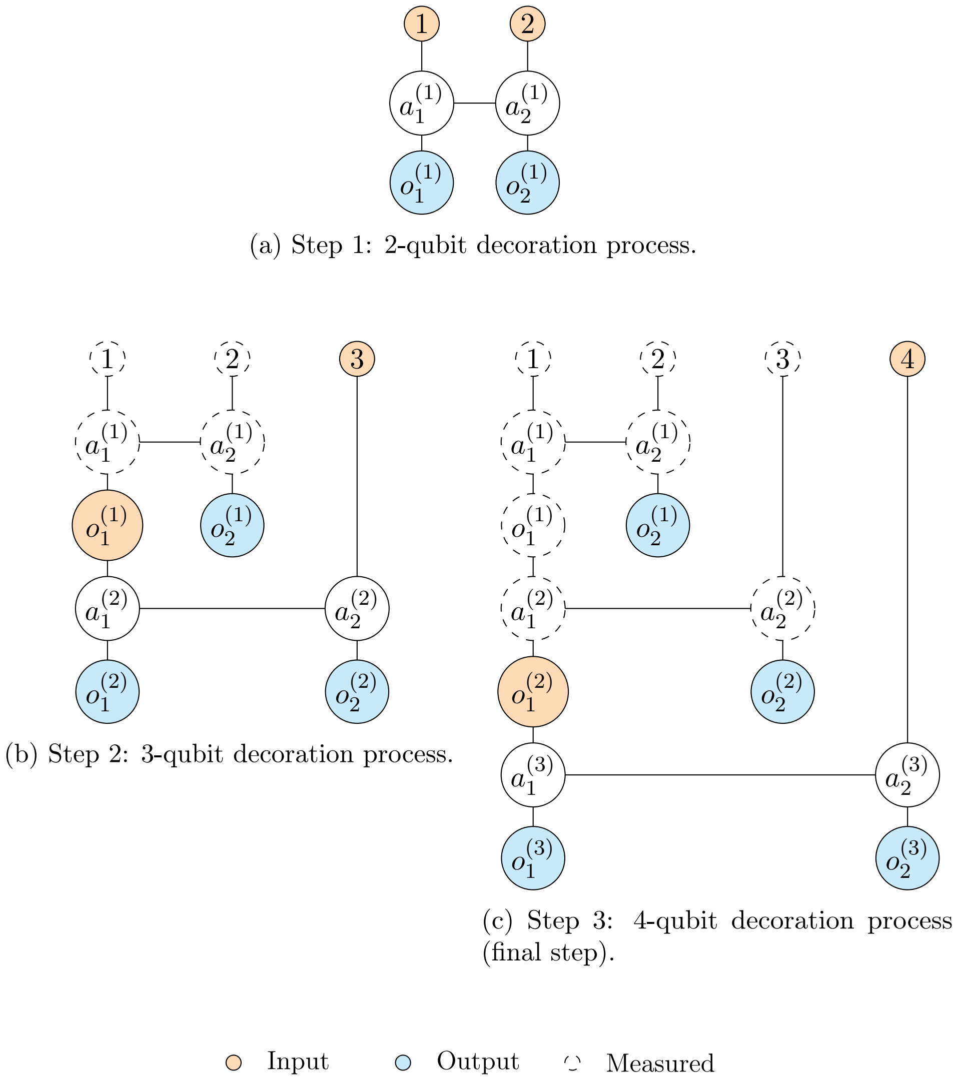
$$\varepsilon_{\text{unpert}} = \frac{|{}_L\langle 0,0|H(\lambda)|0,0\rangle_L - E_{\text{exact}}(\lambda)|}{|E_{\text{exact}}(\lambda)|} \tag{7.17}$$

The red curve shows the performance when using only the separable state $|1\rangle^{\otimes 8}$ for all λ values:

$$\varepsilon_{\text{separable}} = \frac{|\langle 1|^{\otimes 8} H(\lambda)|1\rangle^{\otimes 8} - E_{\text{exact}}(\lambda)|}{|E_{\text{exact}}(\lambda)|} \tag{7.18}$$

The results reveal several important insights. Our MB-VQE approach outperforms both baselines throughout the perturbation range.

In the highly perturbative regime ($\lambda > \lambda_t$), we use as initial graph state a set of isolated qubits with no entanglement between them. Figure 7.15 illustrates the decoration process for this regime using a simple 4-qubit example, showing how the dynamic qubit relabeling method processes the isolated vertices through sequential decoration steps. This procedure naturally extends to larger systems, such as our 8-qubit toric code, by continuing the sequential decoration process until all qubits have been incorporated.



(a) Step 1: 2-qubit decoration process.

(b) Step 2: 3-qubit decoration process.

(c) Step 3: 4-qubit decoration process (final step).

Figure 7.15: Sequential decoration process for isolated qubits in the strong perturbation regime ($\lambda > \lambda_t$). This 4-qubit example illustrates the general strategy that extends to larger systems. The dynamic qubit relabeling ensures deterministic computation by maintaining proper byproduct correction pathways at each step. Dashed circles indicate qubits that have been measured and removed from the active register, while orange and cyan circles show the current input and output qubits respectively.

7.7 Linear Cluster Hamiltonian

To further validate the effectiveness of our MB-VQE approach with dynamic qubit relabeling, we apply it to another stabilizer Hamiltonian: the 6-qubit transverse field cluster model with open boundary conditions. This system provides an additional test case that enables us to assess the versatility of our method across different quantum many-body systems.

The Hamiltonian is:

$$\hat{H}(\lambda) = -\sum_{i=1}^{6} Z_{i-1}\, X_i\, Z_{i+1} - \lambda \sum_{i=1}^{6} Z_i \tag{7.19}$$

We consider again a uniform perturbation controlled by the parameter λ, where the perturbative term $-\lambda \sum_{i=1}^{6} Z_i$ applies a longitudinal magnetic field to all qubits. This perturbation drives the system between two distinct regimes: for $\lambda = 0$, the ground state is the cluster state defined by the stabilizer generators, while for $\lambda \gg 1$, the ground state becomes the product state $|1\rangle^{\otimes 6}$.

The stabilizer generators for the unperturbed cluster state are:

$$S_1 = XZIIII \qquad S_2 = ZXZIII \tag{7.20}$$

$$S_3 = IZXZII \qquad S_4 = IIZXZI \tag{7.21}$$

$$S_5 = IIIZXZ \qquad S_6 = IIIIZX \tag{7.22}$$

These generators satisfy the eigenvalue equation $\hat{S}_i|C\rangle = |C\rangle$ for all i, defining the unique cluster state (up to global phase) as their common $+1$ eigenstate.

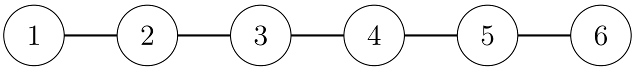


Figure 7.16: Linear chain topology for the 6-qubit cluster Hamiltonian with open boundary conditions. This one-dimensional arrangement serves as the base graph for our dynamic qubit relabeling decoration scheme, providing a simple yet non-trivial test case for validation.

We test our approach using 50 values of λ from 10^{-2} to 10 with logarithmic spacing. For each value of λ, we optimize the MB-VQE ansatz using our dynamic qubit relabeling scheme and compare both the energy and the quantum state obtained with the exact results from full diagonalization.

Figure 7.17 shows that MB-VQE achieves good energy accuracy across the entire parameter range. However, achieving the correct ground state energy does not guarantee that the optimized quantum state is close to the true ground state.

To assess the quality of the quantum states produced by our ansatz, we

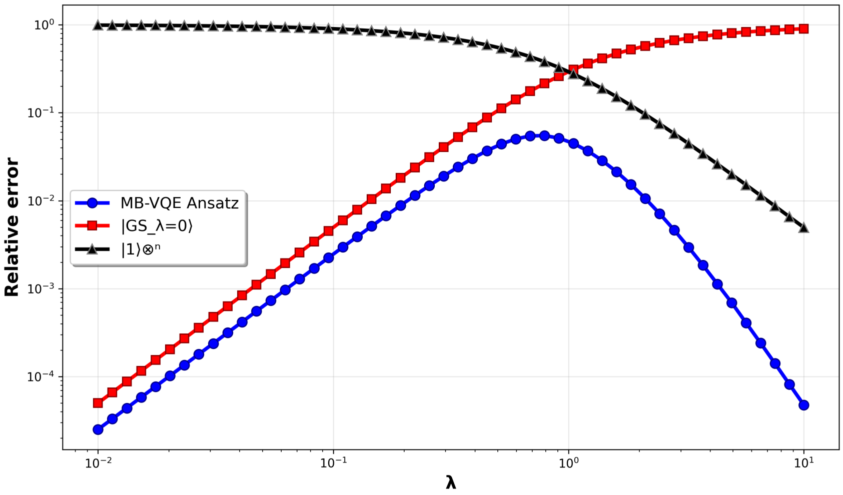


Figure 7.17: Relative energy error for the 6-qubit cluster Hamiltonian as a function of perturbation strength λ, demonstrating excellent energetic performance across all regimes. The blue curve shows $\varepsilon = |E_{\text{MB-VQE}} - E_{\text{exact}}|/|E_{\text{exact}}|$ for each tested value of λ. MB-VQE maintains low relative energy errors across the entire parameter range, demonstrating good energetic accuracy from the cluster state regime ($\lambda \ll 1$) to the product state regime ($\lambda \gg 1$).

analyze the infidelity between the exact ground state and our optimized states:

The infidelity analysis in Figure 7.18 demonstrates that our MB-VQE approach produces quantum states that are close to the true ground states across most of the parameter range. The maximum infidelity occurs near $\lambda \approx 1$, which corresponds to the critical region where the ground state structure changes rapidly.

7.7.1 Gaussian-Distributed Perturbations

We extended the analysis to include spatially inhomogeneous perturbations with a different perturbation type. Instead of applying uniform Z-field perturbations, we consider Gaussian-distributed X-field perturbations.

The modified Hamiltonian with X-field perturbations takes the form:

$$\hat{H}(\{\lambda_i\}) = -\sum_{i=1}^{6} Z_{i-1}\, X_i\, Z_{i+1} - \sum_{i=1}^{6} \lambda_i X_i \tag{7.23}$$

For each qubit i, we sample individual perturbation strengths λ_i from a

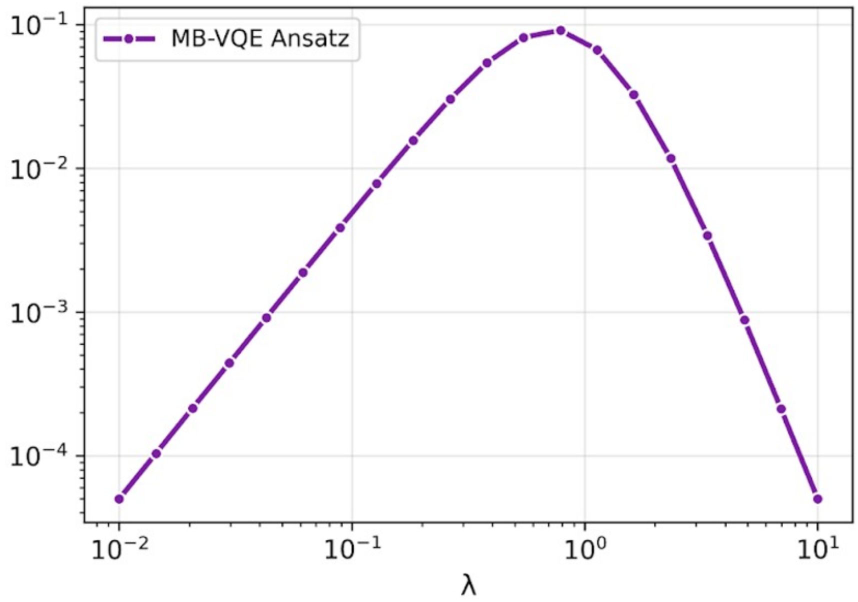


Figure 7.18: State infidelity as a function of perturbation strength λ for the 6-qubit cluster Hamiltonian, revealing the challenging nature of the quantum phase transition region. The blue curve shows $1 - |\langle\psi_{\text{exact}}|\psi_{\text{MB-VQE}}\rangle|^2$ for each tested value of λ. The peak infidelity occurs around $\lambda \approx 1$, which corresponds to the critical region where the ground state structure changes rapidly between the cluster state and product state phases.

Gaussian distribution:

$$\lambda_i \sim \mathcal{N}(\lambda, 0.1\lambda) \tag{7.24}$$

where λ represents the mean perturbation strength and the standard deviation is set to 0.1λ, introducing a 10% relative variation across qubits.

The X-field perturbation fundamentally changes the ground state structure. For large uniform X-fields ($\lambda_i = \lambda \gg 1$), each qubit satisfies the eigenvalue equation:

$$X_i|+\rangle_i = |+\rangle_i \tag{7.25}$$

where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ is the $+1$ eigenstate of the Pauli-X operator. Therefore, the ground state in the strong X-field limit becomes the completely separable state:

$$|\psi_{\text{GS}}\rangle = \bigotimes_{i=1}^{6} |+\rangle_i = |+\rangle^{\otimes 6} \tag{7.26}$$

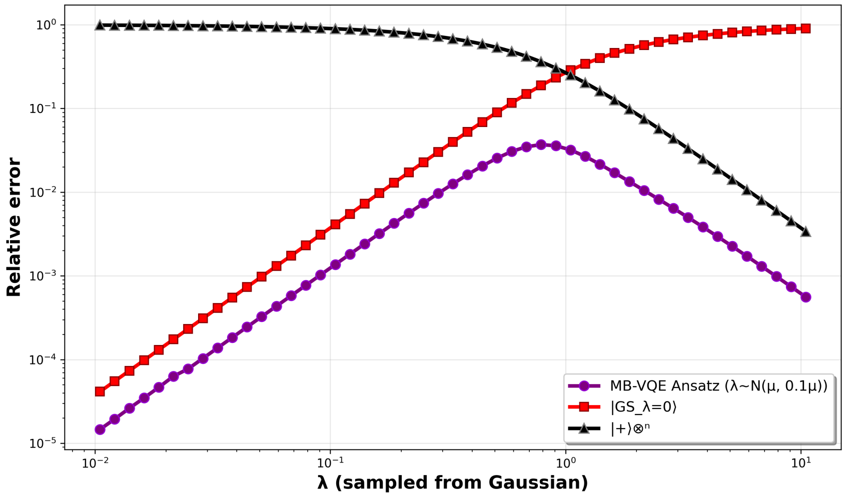
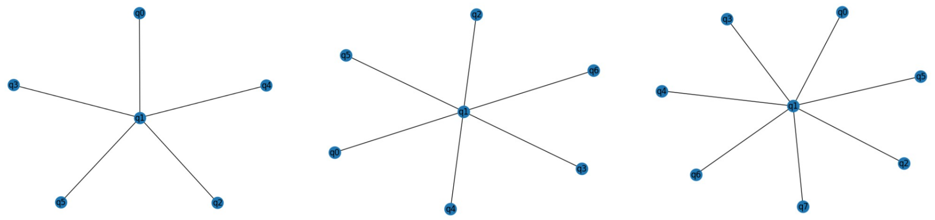


Figure 7.19: MB-VQE performance under Gaussian-distributed X-field perturbation strengths for the 6-qubit cluster Hamiltonian, demonstrating robustness to disorder and different field orientations. The spatial inhomogeneity and change in field direction (from Z to X) test the versatility of our dynamic qubit relabeling approach beyond the standard uniform perturbations, showing maintained performance quality across the transition from cluster state to $|+\rangle^{\otimes 6}$ product state.

7.8 Star Graph Hamiltonians

We investigate the performance of the MB-VQE algorithm on star graph topologies, characterized by a central hub qubit connected to multiple peripheral qubits. We examine systems with 6, 7, and 8 qubits to assess scalability and performance trends as illustrated in Figure 7.20.



(a) 6-qubit star graph (b) 7-qubit star graph (c) 8-qubit star graph

Figure 7.20: Star graph topologies with central qubit q_0 connected to peripheral qubits for different system sizes.

For the 6-qubit star graph, the stabilizer Hamiltonian adopts the characteristic form:

$$\hat{H}_0^{\text{star-6}} = -\hat{X}_0\hat{Z}_1\hat{Z}_2\hat{Z}_3\hat{Z}_4\hat{Z}_5 - \hat{Z}_0\hat{X}_1 - \hat{Z}_0\hat{X}_2 - \hat{Z}_0\hat{X}_3 - \hat{Z}_0\hat{X}_4 - \hat{Z}_0\hat{X}_5 \quad (7.27)$$

The first term represents the central qubit stabilizer involving all peripheral qubits, while the subsequent terms correspond to the individual peripheral qubit stabilizers. This structure creates a natural asymmetry where the central qubit plays a coordinating role in the global stabilizer structure. The star graph state $|S\rangle$ (denoting the stabilized star configuration) is defined as the simultaneous $+1$ eigenstate of all stabilizer generators:

$$\hat{X}_0\hat{Z}_1\hat{Z}_2\hat{Z}_3\hat{Z}_4\hat{Z}_5|S\rangle = |S\rangle \quad (7.28)$$

$$\hat{Z}_0\hat{X}_j|S\rangle = |S\rangle \quad \text{for } j = 1, 2, 3, 4, 5 \quad (7.29)$$

For the 7-qubit and 8-qubit configurations, additional peripheral qubits are incorporated following the same architectural principle, preserving the star connectivity pattern.

We introduce perturbations through uniform magnetic fields $\hat{H}_{\text{pert}} = -\sum_i \hat{Z}_i$, constructing the interpolating Hamiltonians $\hat{H}(\lambda) = \hat{H}_0^{\text{star}} + \lambda\hat{H}_{\text{pert}}$ to study the transition from the star graph state to the paramagnetic phase. In the strong perturbation limit ($\lambda \gg 1$), the ground state becomes:

$$|\psi_{\text{GS}}\rangle = |1\rangle^{\otimes N} \quad (7.30)$$

where N is the total number of qubits (6, 7, or 8).

The comparative analysis in Figures 7.21 and 7.22 shows that performance remains essentially equivalent across the different system sizes. The 6, 7, and 8-qubit star graphs exhibit similar energy errors and infidelity values, indicating that adding peripheral qubits to the star structure does not significantly impact the performance of our dynamic qubit relabeling approach in this size range.

7.9 Complete Graph Hamiltonians

We examine complete graph topologies where every qubit maintains direct connectivity with all other qubits, representing maximally entangled stabilizer states.

We analyze 6-qubit and 8-qubit complete graphs as shown in Figure 7.23.

For the 6-qubit complete graph, the stabilizer Hamiltonian exhibits full

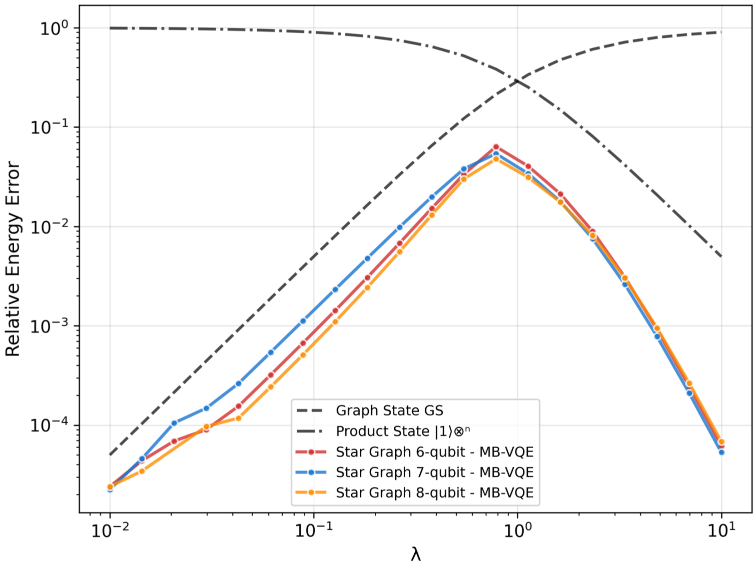


Figure 7.21: Energy performance comparison for star graph configurations with 6, 7, and 8 qubits as a function of perturbation strength λ. The three curves show the relative energy error $\varepsilon = |E_{\text{MB-VQE}} - E_{\text{exact}}|/|E_{\text{exact}}|$ for each system size. The performance remains essentially equivalent across different system sizes, indicating that the hierarchical star structure does not show significant scaling advantages in this size range.

permutation symmetry:

$$\begin{aligned}\hat{H}_0^{\text{complete-6}} &= -\hat{X}_0\hat{Z}_1\hat{Z}_2\hat{Z}_3\hat{Z}_4\hat{Z}_5 - \hat{Z}_0\hat{X}_1\hat{Z}_2\hat{Z}_3\hat{Z}_4\hat{Z}_5 - \hat{Z}_0\hat{Z}_1\hat{X}_2\hat{Z}_3\hat{Z}_4\hat{Z}_5 \\ &\quad - \hat{Z}_0\hat{Z}_1\hat{Z}_2\hat{X}_3\hat{Z}_4\hat{Z}_5 - \hat{Z}_0\hat{Z}_1\hat{Z}_2\hat{Z}_3\hat{X}_4\hat{Z}_5 - \hat{Z}_0\hat{Z}_1\hat{Z}_2\hat{Z}_3\hat{Z}_4\hat{X}_5\end{aligned} \tag{7.31}$$

Each stabilizer generator involves precisely one qubit in the X measurement basis while all remaining qubits are measured in the Z basis. The complete graph state $|G\rangle$ (denoting the global complete graph configuration) satisfies the eigenvalue equations:

$$\hat{S}_j|G\rangle = |G\rangle \quad \text{for all } j = 0, 1, 2, 3, 4, 5 \tag{7.32}$$

where $\hat{S}_j$ represents the j-th stabilizer generator.

We apply identical perturbations $\hat{H}_{\text{pert}} = -\sum_i \hat{Z}_i$, yielding the interpolating Hamiltonians $\hat{H}(\lambda) = \hat{H}_0^{\text{complete}} + \lambda\hat{H}_{\text{pert}}$. In the strong perturbation

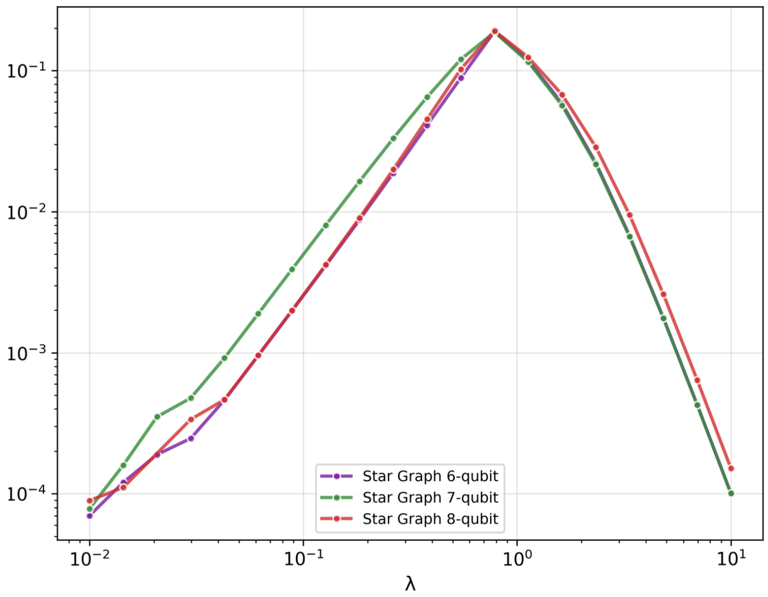


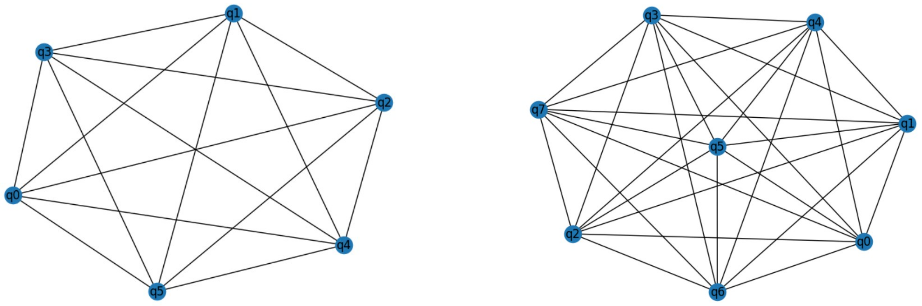
Figure 7.22: Infidelity comparison for star graph configurations with 6, 7, and 8 qubits as a function of perturbation strength λ. The three curves show the state infidelity $1 - |\langle\psi_{\text{exact}}|\psi_{\text{MB-VQE}}\rangle|^2$ for each system size, confirming that performance remains consistent across different star graph sizes.

limit, the ground state becomes:

$$|\psi_{\text{GS}}\rangle = |1\rangle^{\otimes N} \tag{7.33}$$

where each qubit is in the $\hat{Z} = +1$ eigenstate.

The performance analysis in Figures 7.24 and 7.25 demonstrates lower relative energy errors for small λ values compared to other graph topologies. This improved performance can be attributed to the highly entangled nature of the complete graph initial state, which provides a better starting point for the variational optimization in the weak perturbation regime where the ground state retains significant entanglement structure.



(a) 6-qubit complete graph

(b) 8-qubit complete graph

Figure 7.23: Complete graph topologies where every qubit is connected to all other qubits, representing maximally entangled stabilizer states. This architecture provides the richest possible entanglement structure within the stabilizer formalism.

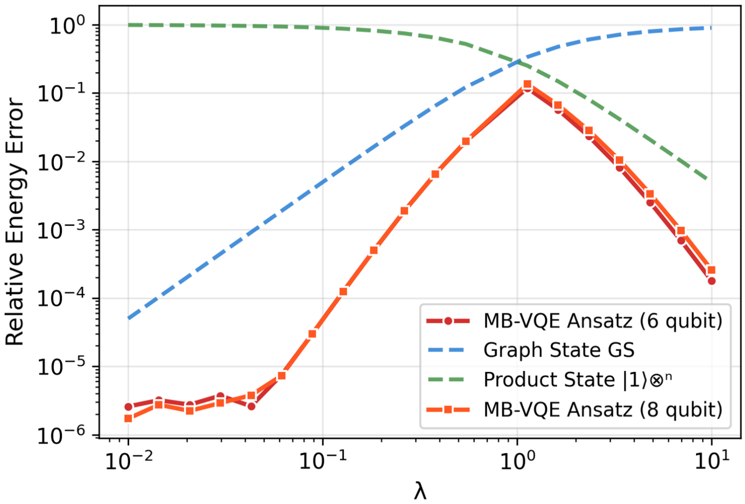


Figure 7.24: Energy performance comparison for complete graph configurations with 6 and 8 qubits as a function of perturbation strength λ. The two curves show lower relative energy errors for small λ values, reflecting the advantage of starting from highly entangled initial graph states.

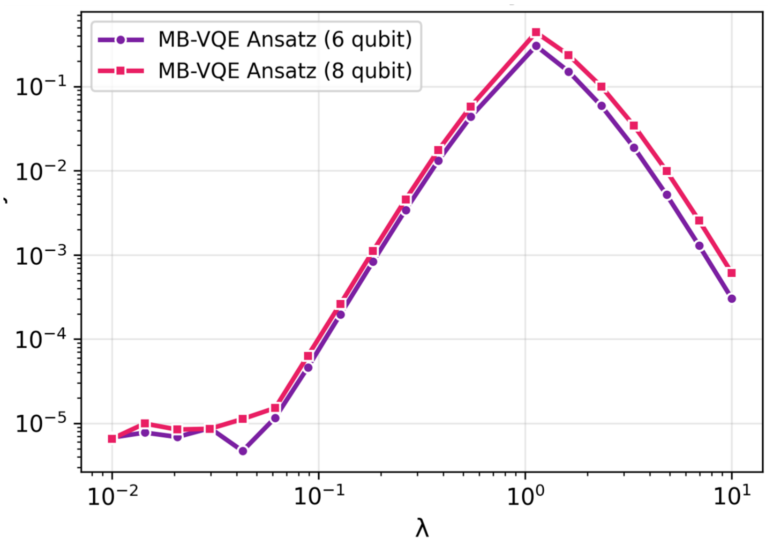


Figure 7.25: Infidelity comparison for complete graph configurations with 6 and 8 qubits as a function of perturbation strength λ. The two curves show the state infidelity $1 - |\langle\psi_{\text{exact}}|\psi_{\text{MB-VQE}}\rangle|^2$ for both system sizes, confirming the improved performance for small perturbations.

Chapter 8

Conclusions

This thesis addresses the determinism problem in edge-wise decoration scheme for Measurement-Based Variational Quantum Eigensolvers (MB-VQE), a theoretical limitation that prevents this scheme from achieving the computational equivalence with circuit-based models that MBQC requires.

The problem we addressed stems from a violation of generalized flow (gflow) conditions in existing edge-wise decoration schemes. Just as circuit-based quantum computation guarantees deterministic output states for given input parameters, MBQC requires flow conditions on the resource graph to ensure deterministic computation through measurement outcome correction. When auxiliary qubits are added to edges sharing common vertices, the resulting graph structure violates these conditions, breaking the deterministic guarantee. This theoretical limitation means the algorithm cannot reliably reproduce the same quantum state across multiple runs.

Our main contribution is a novel edge-wise decoration scheme with dynamic qubit relabeling that maintains deterministic computation while preserving the advantages of the original approach. Unlike the node-wise decoration solution by Schroeder et al., which requires additional spatial dimensions and more complex resource states, our method maintains the original graph dimensionality while ensuring gflow conditions through strategic qubit role reassignment during the computation process. Input and output qubits are dynamically updated at each decoration step, allowing byproduct corrections to proceed systematically without violating flow conditions.

We implemented and validated this approach using 4 variational parameters per edge decoration plus 3 single-qubit rotation parameters per output qubit. Our method was tested on multiple stabilizer Hamiltonians including the 8-qubit toric code, 6-qubit linear cluster, star graphs, and complete graphs.

The numerical simulations demonstrate that our approach achieves relative energy errors below 10^{-1} across different perturbation regimes, consistently outperforming both unperturbed graph states in weak perturbation regimes and purely perturbative ground states in strong perturbation regimes. These results validate both the theoretical soundness of our deterministic scheme and its practical effectiveness for ground state approximation.

Our validation was limited to systems with 6-8 qubits due to computational constraints of exact diagonalization, and our current implementation focuses on stabilizer Hamiltonians where the ground states can be represented as graph states.

Several promising directions emerge for future research. The method could be extended to larger quantum systems. Moreover, different decorative subgraphs topologies could be explored to optimize the balance between variational expressivity and parameter efficiency. The most promising extension involves applying our deterministic approach to non-stabilizer Hamiltonians. Following the methodology of Sun et al. [36], we could identify sets of commuting stabilizers that approximate ground states of interacting models like Ising and Heisenberg Hamiltonians. These would define graph states serving as starting points for variational ansätze, potentially extending our deterministic MB-VQE beyond stabilizer systems to capture perturbation effects in quantum many-body physics.

This work provides a concrete solution to determinism issues in edgewise decoration scheme, opening a path toward reliable measurement-based quantum algorithms. We developed and implemented simulation code using the graphix library, contributing practical tools for the quantum computing community.

Appendix A

A.1 Qiskit implementation of Pauli exponential extraction

This code verifies numerically that the circuit representation of the measurement pattern of the edge-wise decoration [9] for a single edge, obtained in the main text, is correct. The qubits 0 and 1 correspond to qubits m and n in the main text. The initial $\mathrm{CZ}|++\rangle$ state is prepared using Hadamard gates followed by a controlled-Z gate. Four Pauli evolution gates $e^{-i\theta_i/2 \cdot \mathrm{Pauli}}$ are then applied to simulate the effect of measuring auxiliary qubits at angles $\theta_1, \theta_2, \theta_3, \theta_4$.

The code verifies two extreme cases: setting all angles to 0 must yield the unchanged $\mathrm{CZ}|++\rangle$ state (since zero-angle measurements leave the state invariant), while setting all angles to $\pi/2$ must disentangle qubits 0 and 1 (since $\pi/2$-angle measurements eliminate entanglement between auxiliary qubits).

```python
import numpy as np
import networkx as nx
from qiskit import QuantumCircuit
from qiskit.circuit import Parameter
from qiskit.quantum_info import Statevector, Pauli, SparsePauliOp
from qiskit.circuit.library import PauliEvolutionGate

# Create graph and prepare graph state
G = nx.Graph()
G.add_nodes_from([0, 1])
G.add_edge(0, 1)
qc = QuantumCircuit(2)
qc.h([0, 1])
qc.cz(0, 1)

# Parameters and Pauli exponentials
v = [Parameter(f'theta_{i}') for i in range(4)]

qc.append(PauliEvolutionGate(SparsePauliOp(Pauli('X')),v[0]/2),[0])

qc.append(PauliEvolutionGate(SparsePauliOp(Pauli('X')),v[1]/2),[1])

qc.append(PauliEvolutionGate(SparsePauliOp(Pauli('XZ')),v[2]/2),[0,1])

qc.append(PauliEvolutionGate(SparsePauliOp(Pauli('XZ')),v[3]/2),[1,0])
```

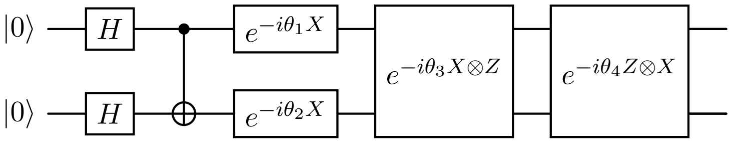


Figure A.1: MBQC Circuit with parametric gates

The circuit implements the following sequence:

1. Prepare $|++\rangle$ state using Hadamard gates on both qubits

2. Apply controlled-Z gate to create entanglement

3. Apply single-qubit X-rotations: $e^{-i\theta_1 X}$ and $e^{-i\theta_2 X}$

4. Apply two-qubit parametric gates: $e^{-i\theta_3 X\otimes Z}$ and $e^{-i\theta_4 Z\otimes X}$

```python
qc_test1 = qc.assign_parameters({v[i]: 0 for i in range(4)})
output_state1 = Statevector.from_instruction(qc_test1)
fidelity1 = state_fidelity(output_state1, cz_plus_state)

qc_test2 = qc.assign_parameters({v[i]: np.pi/2 for i in range(4)})
output_state2 = Statevector.from_instruction(qc_test2)
fidelity2 = state_fidelity(output_state2, plus_plus_state)

print(f"Fidelity (angles = 0): {fidelity1:.6f}")
print(f"Fidelity (angles = pi/2): {fidelity2:.6f}")
```

Fidelity (angles=0): 1.000

Fidelity (angles=$\pi/2$): 1.000

To verify that the obtained states are correct, I measured the fidelity between the circuit output and the expected reference states. The fidelity equals 1.000 for both extreme cases, confirming that the circuit representation accurately the MBQC pattern described in the main text.

A.2 Verification of gflow Existence edge-wise decoration

```python
import networkx as nx
from graphix.pattern import generate_from_graph
from graphix.flow import gflow_from_pattern

# 1) Build open graph and define I/O
G = nx.Graph()
G.add_edges_from([(0,1), (1,2), (2,3), (2,9), (3,8), (9,10), (8,10),
                  (1,4), (2,5), (4,5), (4,6), (5,7)])

inputs, outputs = [], [1,2,3]
angles = {i: 0.5 for i in range(10)}  # includes all nodes 0-9

# 2) Try to generate deterministic MBQC pattern
try:
  pat = generate_from_graph(G, angles, inputs=inputs, outputs=outputs)

  # 3) Extract gflow map and measurement layers
  g_map, layers = gflow_from_pattern(pat)
  print("gflow map:", g_map)
  print("measurement layers:", layers)

except Exception as e:
  print("Error: No gflow found for the specified graph structure.")
```

A.3 Verification of gflow Existence node-wise decoration

```python
def verify_mbqc_determinism(base_nodes, num_layers=3, trials=10):
    """Verify MBQC determinism across multiple measurement runs."""
    G = nx.Graph()
    G.add_nodes_from(base_nodes)
    G.add_edges_from([(base_nodes[i], base_nodes[i+1]) for i in range(len(base_nodes)-1)])

    decoration_nodes = []
    for layer in range(num_layers):
        layer_nodes = [node + 10*(layer+1) for node in base_nodes]
        decoration_nodes.extend(layer_nodes)
        G.add_nodes_from(layer_nodes)
        G.add_edges_from([(layer_nodes[i], layer_nodes[i+1]) for i in range(len(layer_nodes)-1)])
        prev_layer = base_nodes if layer == 0 else [node + 10*layer for node in base_nodes]
        G.add_edges_from([(prev_layer[i], layer_nodes[i]) for i in range(len(layer_nodes))])

    angles = {node: np.random.uniform(0, 2*np.pi) for node in decoration_nodes}
    pattern = generate_from_graph(G, angles, inputs=[], outputs=base_nodes)
    byproducts = extract_byproducts(pattern)
    states = []
    for i in range(trials):
        sim = PatternSimulator(pattern, backend="statevector")
        sim.run()
        psi = sim.backend.state.psi.flatten()
        state = psi * np.exp(-1j * np.angle(psi[np.argmax(np.abs(psi))]))
        states.append(np.round(state, 6))
    deterministic = all(np.allclose(states[0], s, atol=1e-12) for s in states[1:])
    return deterministic
base_nodes = [0, 1, 2]
verify_mbqc_determinism(base_nodes, num_layers=3, trials=5)
```

```
Physical nodes: [0, 1, 2] Decoration nodes: [10, 11, 12, 20, 21, 22, 30, 31, 32]
    Node 32: X→[22], Z→[12, 21]
    Node 30: X→[20], Z→[10, 21]
    Node 31: X→[21], Z→[11, 20, 22]
    Node 20: X→[10], Z→[0, 11]
    Node 21: X→[11], Z→[1, 10, 12]
    Node 22: X→[12], Z→[2, 11]
    Node 10: X→[0], Z→[1]
    Node 11: X→[1], Z→[0, 2]
    Node 12: X→[2], Z→[1]
```

A.4 Circuit Unitary Extraction

Although there are many methods for converting gate-based circuits into measurement patterns, the reverse process, converting measurement patterns into equivalent unitary gate sequences, is more challenging.

In the work of [35], a procedure is presented to extract the sequence of unitary gates that implements the same linear transformation as a given measurement pattern. Recall that a measurement pattern is a formal specification describing how to perform quantum computation through sequential measurements on a pre-prepared entangled state: (1) which qubits to measure, (2) in what order, (3) the measurement basis for each qubit, and (4) how measurement outcomes from previous qubits determine the basis choices for subsequent measurements.

For technical details see [35]. The method relies on exploiting the stabilizer formalism and the focused gflow of the graph state that serves as the resource for the computation.

We now consider the graph in Figure 7.1 and apply the circuit extraction procedure.

The first step is to identify the focused gflow in the graph state. For this goal, it is needed to identify the correction set of each measured qubit, and the partial order of the measurements. The correction sets, shown in Table A.1, determine how the measurement outcomes in individual qubits propagate corrections throughout the system.

v	$g(v)$
2	$\{1, n\}$
4	$\{3, m\}$
1	$\{m\}$
3	$\{n\}$

Table A.1: Correction sets for the qubits in Figure 7.1

This focused gflow establishes the causal structure of the measurement protocol: measuring qubit 2 implies corrections on qubits 1 and n, measuring qubit 4 affects qubits 3 and m, and so forth. The partial ordering $2, 4 < 1, 3$ [9] specifies that qubits 2 and 4 must be measured before qubits 1 and 3.

The auxiliary qubits are measured in rotated bases in the XY plane. A measurement at angle α in the XY plane can be decomposed as a rotation followed by measurement in the computational basis.

Let $|\pm_X\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ denote the eigenstates of the Pauli-X operator with eigenvalues ±1. The measurement eigenstates are obtained by rotating the X-basis states around the Z-axis:

$$|\pm_{XY,\alpha}\rangle = e^{-i\frac{\alpha}{2}Z}|\pm_X\rangle$$

The corresponding projection operators are:

$$\hat{P}_X^{\pm} = |\pm_X\rangle\langle\pm_X| = \frac{1}{2}(\mathbb{I} \pm X)$$
$$\hat{P}_{XY,\alpha}^{\pm} = |\pm_{XY,\alpha}\rangle\langle\pm_{XY,\alpha}| = e^{-i\frac{\alpha}{2}Z}\hat{P}_X^{\pm}e^{i\frac{\alpha}{2}Z}$$

Throughout this analysis, we use the compact notation $Z(\theta) = e^{-i\frac{\theta}{2}Z}$ for rotations around the Z-axis by angle θ.

Considering Figure 7.1 we define qubits 2 and 4 as input qubits, and we measure qubits 2, 4, 1 and 3. The qubits m and n are output qubits and therefore they are not measured. The stabilizers of the graph are:

$$S = \langle X_1Z_mZ_2Z_3, X_3Z_1Z_4Z_n, X_mZ_nZ_1, X_nZ_mZ_3\rangle$$

where the stabilizers of the input qubits are excluded for simplicity (they are never involved during the extraction procedure) and inglobated into the expression $|\psi, S\rangle$, that is the initial input state.

Following the established measurement order, the complete measurement sequence can be expressed as:

$$\prod_{i\in\{3,1,4,2\}} \hat{P}_{X_i}^{m_i}Z_i(\theta_i)|\psi, S\rangle = \hat{P}_{X_3}^{m_3}Z_3(\theta_3)\hat{P}_{X_1}^{m_1}Z_1(\theta_1)\hat{P}_{X_4}^{m_4}Z_4(\theta_4)\hat{P}_{X_2}^{m_2}Z_2(\theta_2)|\psi, S\rangle$$

where $|\psi, S\rangle$ represents the graph state stabilized by S, and $\hat{P}_{X_i}^{m_i} = \hat{P}_{X_i}^{+}$ if $m_i = +1$ and $\hat{P}_{X_i}^{m_i} = \hat{P}_{X_i}^{-}$ if $m_i = -1$.

The insight for extracting equivalent unitary circuits lies in systematic exploitation of stabilizer projectors. For each stabilizer generator $C_j \in S$, the stabilizer projector is $\hat{P}_{C_j} = \frac{1}{2}(\mathbb{I} + C_j)$. By definition of stabilizer states, this projector acts as the identity when applied to any state in the stabilizer space: $\hat{P}_{C_j}|\psi, S\rangle = |\psi, S\rangle$. This allows insertion of stabilizer projectors into the measurement sequence without altering the final result, after which they can be commuted through the measurement operators using algebraic manipulation.

Consider the stabilizer $C_j = X_nZ_mZ_3$. Inserting its projector into the measurement sequence yields:

$$\hat{P}_{X_3}^{m_3} Z_3(\theta_3) \hat{P}_{X_1}^{m_1} Z_1(\theta_1) \hat{P}_{X_4}^{m_4} Z_4(\theta_4) \hat{P}_{X_2}^{m_2} Z_2(\theta_2) \hat{P}_{X_n Z_m Z_3} |\psi, S\rangle$$

Since the stabilizer projector commutes with all measurement operators acting on different qubits (due to the stabilizer property), it can be moved forward to immediately follow the $Z_3(\theta_3)$ rotation:

$$\hat{P}_{X_3}^{m_3} Z_3(\theta_3) \hat{P}_{X_n Z_m Z_3} \hat{P}_{X_1}^{m_1} Z_1(\theta_1) \hat{P}_{X_4}^{m_4} Z_4(\theta_4) \hat{P}_{X_2}^{m_2} Z_2(\theta_2) |\psi, S\rangle$$

The crucial step involves applying the product rotation lemma [35], which enables commutation of rotations through projectors via operator conjugation. Specifically:

$$\begin{aligned} Z_3(\theta_3) \hat{P}_{X_n Z_m Z_3} &= e^{-i \frac{\theta_3}{2} Z_3} \hat{P}_{X_n Z_m Z_3} \\ &= e^{-i \frac{\theta_3}{2} Z_3 X_n Z_m Z_3} \hat{P}_{X_n Z_m Z_3} \\ &= e^{-i \frac{\theta_3}{2} X_n Z_m} \hat{P}_{X_n Z_m Z_3} \\ &= [X_n Z_m](\theta_3) \hat{P}_{X_n Z_m Z_3} \end{aligned}$$

where the identity $Z_3 X_n Z_m Z_3 = X_n Z_m$ follows from the anticommutation relation $\{Z_3, Z_3\} = \mathbb{I}$. Since the extracted unitary $[X_n Z_m](\theta_3)$ acts exclusively on qubits m and n while the measurement $\hat{P}_{X_3}^{m_3}$ acts on qubit 3, they commute, yielding:

$$[X_n Z_m](\theta_3) \hat{P}_{X_3}^{m_3} \hat{P}_{X_1}^{m_1} Z_1(\theta_1) \hat{P}_{X_4}^{m_4} Z_4(\theta_4) \hat{P}_{X_2}^{m_2} Z_2(\theta_2) |\psi, S\rangle$$

Applying the same procedure to the stabilizer $X_m Z_n Z_1$ produces:

$$[X_n Z_m](\theta_3) [X_m Z_n](\theta_1) \hat{P}_{X_3}^{m_3} \hat{P}_{X_1}^{m_1} \hat{P}_{X_4}^{m_4} Z_4(\theta_4) \hat{P}_{X_2}^{m_2} Z_2(\theta_2) |\psi, S\rangle$$

Now we should insert the projector of the stabilizer $X_3 Z_1 Z_4 Z_n$, but we encounter a complication: the presence of Z_1 prevents direct commutation through the measurement sequence. To resolve this, we multiply this stabilizer with the previously inserted $X_m Z_n Z_1$ to eliminate the problematic Z_1 terms:

$$X_3 Z_1 Z_4 Z_n \cdot X_m Z_n Z_1 = X_3 Z_4 X_m$$

where we used the identity relations $Z_n^2 = Z_1^2 = \mathbb{I}$. Applying the rotation lemma with this combined stabilizer gives:

$$[X_n Z_m](\theta_3)[X_m Z_n](\theta_1)[X_m](\theta_4)\hat{P}_{X_3}^{m_3}\hat{P}_{X_1}^{m_1}\hat{P}_{X_4}^{m_4}\hat{P}_{X_2}^{m_2}Z_2(\theta_2)|\psi, S\rangle$$

where the X_3 factor has been absorbed into the measurement projector $\hat{P}_{X_3}^{m_3}$ due to the idempotent property of projection operators.

Finally, processing the remaining stabilizers $X_1 Z_m Z_2 Z_3$ and $X_n Z_m Z_3$ through multiplication:

$$X_1 Z_m Z_2 Z_3 \cdot X_n Z_m Z_3 = X_1 Z_2 X_n$$

where we again used $Z_m^2 = Z_3^2 = \mathbb{I}$. This yields the final result:

$$[X_n Z_m](\theta_3)[X_m Z_n](\theta_1)[X_m](\theta_4)[X_n](\theta_2)\hat{P}_{X_3}^{m_3}\hat{P}_{X_1}^{m_1}\hat{P}_{X_4}^{m_4}\hat{P}_{X_2}^{m_2}|\psi, S\rangle$$

Let us have a look on the single terms of this expression:

- $[X_n Z_m](\theta_3)$ and $[X_m Z_n](\theta_1)$: Two-qubit unitary operations on output qubits n and m.
- $[X_m](\theta_4)$ and $[X_n](\theta_2)$: Single-qubit rotations on output qubits n and m.
- $\hat{P}_{X_i}^{m_i}$: X-basis measurements on auxiliary qubits.
- $|\psi, S\rangle$: Input quantum state, stabilized by the set of stabilizers S.

This procedure demonstrates that the MBQC computation can be rewritten as an equivalent gate-based circuit. The measurement pattern produces the same quantum state as the unitary circuit operations applied to the state of qubits m and n, that are initially in the state CZ$|+\rangle|+\rangle$,

$$\mathcal{U} = [X_n Z_m](\theta_3)[X_m Z_n](\theta_1)[X_m](\theta_4)[X_n](\theta_2)$$

followed by projective X measurements on the auxiliary qubits.

A simple numerical implementation is demonstrated through quantum circuits that directly apply the extracted Pauli exponentials to the qubits m and n. Using Qiskit's PauliEvolutionGate class, each extracted unitary operator $[P_i](\theta_j)$ can be implemented as a parametrized quantum gate Appendix A.1.

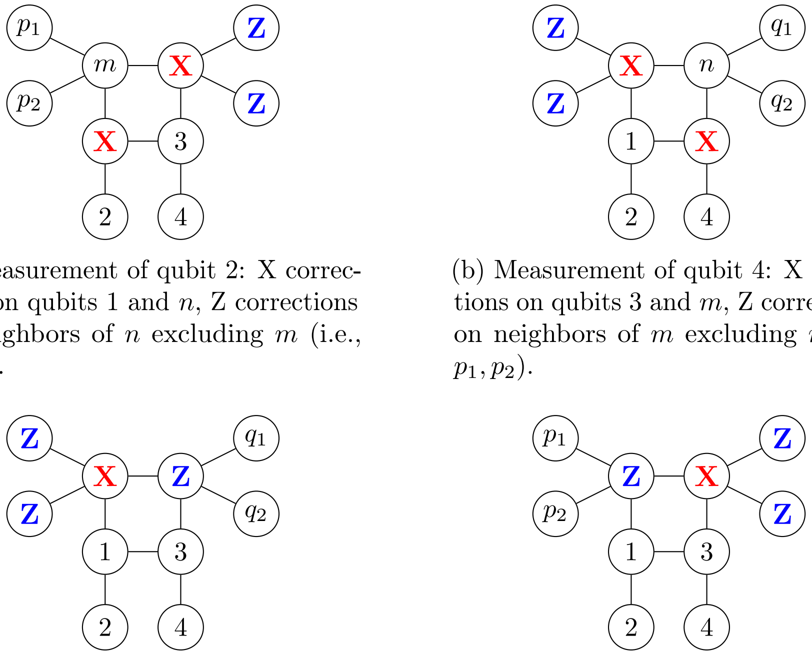
Generalization to Arbitrary Graph Structures

The unitary extraction procedure can be extended to all graph topologies. When the output vertices m, n are adjacent to extra vertices compared to the minimal configuration, the resulting Pauli exponentials must be modified to take into account the neighborhood of the qubits m and n :

$$\mathcal{U}_{\text{general}} = \left[X_n \, Z_{N_n \setminus \{m\}}\right](\theta_3) \cdot \left[X_m \, Z_{N_m \setminus \{n\}}\right](\theta_4) \cdot \left[X_n \, Z_{N_n}\right](\theta_2) \cdot \left[X_m \, Z_{N_m}\right](\theta_1)$$

where $N_k = \{i \mid (k, i) \in E\}$ is the set of neighbours of vertex k in graph $G = (V, E)$.

The form of the correction sests for each measurement follows from the focused gflow construction, as illustrated in Figure A.2.



(a) Measurement of qubit 2: X corrections on qubits 1 and n, Z corrections on neighbors of n excluding m (i.e., q_1, q_2).

(b) Measurement of qubit 4: X corrections on qubits 3 and m, Z corrections on neighbors of m excluding n (i.e., p_1, p_2).

(c) Measurement of qubit 1: X correction on m, Z corrections on all neighbors of m (i.e., n, p_1, p_2).

(d) Measurement of qubit 3: X correction on n, Z corrections on all neighbors of n (i.e., m, q_1, q_2).

Figure A.2: Pauli corrections applied after each sequential measurement, illustrating the correction pattern determined by the focused gflow.

We have characterized this circuit extraction procedure, which will be useful later for simulation. We now present the main problem we addressed.

If we consider the decoration procedure and apply it to larger graphs, decorating each edge by attaching ancillary qubits, how does the graph property modify? Can we still identify the gflow on this larger structure? The answer is no, and the absence of gflow prevents deterministic measurement-based quantum computation.

While the single-edge case works smoothly, when we consider two (or more) edges sharing a common vertex (or vertices) and we apply the decoration procedure followed by auxiliary qubit measurements, gflow is not guaranteed. This means that the measurement output states cannot be deterministic but depend on individual measurement outcomes.

The entire purpose of having a resource state for MBQC is ensure a deterministic measurement, the resource state must satisfy the necessary and sufficient condition of gflow.

Bibliography

[1] P. W. Shor. *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.* arXiv:quant-ph/9508027

[2] L. K. Grover. *A fast quantum mechanical algorithm for database search.* arXiv:quant-ph/9605043

[3] J. Preskill. *Quantum Computing in the NISQ era and beyond.* Quantum 2, 79 (2018). DOI: 10.22331/q-2018-08-06-79

[4] A. Peruzzo, J. McClean, P. Shadbolt, M.H. Yung, X.Q. Zhou, P.J. Love, A. Aspuru-Guzik, J.L. O'Brien. *A variational eigenvalue solver on a quantum processor.* Nature Communications 5, 4213 (2014). arXiv:1304.3061

[5] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J.M. Chow, J.M. Gambetta. *Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets.* Nature 549, 242–246 (2017). DOI: 10.1038/nature23879

[6] R. Raussendorf, H. J. Briegel. *A one-way quantum computer.* Physical Review Letters 86, 5188 (2001). DOI: 10.1103/PhysRevLett.86.5188

[7] H.J. Briegel, R. Raussendorf. *Persistent Entanglement in Arrays of Interacting Particles.* Physical Review Letters 86, 910 (2001). DOI: 10.1103/PhysRevLett.86.910

[8] D.E. Browne, E. Kashefi, M. Mhalla, S. Perdrix. *Generalized flow and determinism in measurement-based quantum computation.* New Journal of Physics, 9, 250 (2007). DOI: 10.1088/1367-2630/9/8/250

[9] R. R. Ferguson, L. Dellantonio, K. Jansen, A. Al Balushi, W. Dür, C. A. Muschik. *A measurement-based variational quantum eigensolver.* Physical Review Letters 126, 220501 (2021). arXiv: 2010.13940

[10] A. Schroeder, M. Heller, and M. Gachechiladze. *Deterministic Ansätze for the Measurement-based Variational Quantum Eigensolver.* arXiv:2312.13241 (2023). DOI: 10.48550/arXiv.2312.13241

[11] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, (2010).

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

[13] J. Kempe, A. Kitaev, O. Regev. *The Complexity of the Local Hamiltonian Problem.* SIAM Journal on Computing, 35(5):1070-1097 (2006). DOI: 10.1137/S0097539704445226

[14] Steven R. White *Density matrix formulation for quantum renormalization groups.* DOI: https://doi.org/10.1103/PhysRevLett.69.2863

[15] Wirawan Purwanto, Shiwei Zhang *Quantum Monte Carlo method for the ground state of many-boson systems.* arXiv:physics/0403146

[16] A. Peruzzo, J. McClean, P. Shadbolt, et al. *A variational eigenvalue solver on a photonic quantum processor.* Nature Communications, 5:4213 (2014). DOI: 10.1038/ncomms5213

[17] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, D. Preda. *A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem.* Science, 292(5516):472-475 (2001). DOI: 10.1126/science.1057726

[18] A. Yu. Kitaev. *Fault-tolerant quantum computation by anyons.* Annals of Physics 303, 2–30 (2003). DOI: 10.1016/S0003-4916(02)00018-0

[19] P. Benioff. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. Journal of Statistical Physics 22, 563–591 (1980). DOI: 10.1007/BF01011339

[20] R. P. Feynman. Simulating physics with computers. International Journal of Theoretical Physics 21, 467–488 (1982). DOI: 10.1007/BF02650179

[21] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. Proceedings of the Royal Society of London A 400, 97–117 (1985). DOI: 10.1098/rspa.1985.0070

[22] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. Nature 414, 883–887 (2001). DOI: 10.1038/414883a

[23] IBM Quantum Experience Team. IBM Quantum Experience: A new online quantum simulator and hardware. IBM Research Blog (2016). URL: https://research.ibm.com/blog/quantum-network

[24] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, et al. Quantum supremacy using a programmable superconducting processor. Nature 574, 505–510 (2019). DOI: 10.1038/s41586-019-1666-5

[25] V. Danos, E. Kashefi. *Determinism in the one-way model.* Physical Review A, 74(5), 052310 (2006). arXiv: quant-ph/0506062

[26] T. Wellens, M. Lüdecke, W. P. Schleich. *Noisy Intermediate-Scale Quantum (NISQ) Computers—How They Work and What They Can (Not) Do.* arXiv preprint arXiv:2301.11739 (2023). arXiv:2301.11739

[27] C. Wang *et al. 99.9%-fidelity in measuring a superconducting qubit.* arXiv:2412.13849 (2024).

[28] Quera. *What is Variational Quantum Algorithm.* Quera Glossary (2024).

[29] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles. *Variational Quantum Algorithms.* Nature Reviews Physics, 3(9), 625–644 (2021). arXiv:2012.09265

[30] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien. *A variational eigenvalue solver on a quantum processor.* Nature Communications, 5:4213, (2014).

[31] Y.-K. Liu, M. Christandl, F. Verstraete. *Quantum Computational Complexity of the N-Representability Problem: QMA Complete.* Physical Review Letters 98, 110503 (2007). arXiv:quant-ph/0609125

[32] X. Gao, E. Barnes, S. Kais. *The quantum generative adversarial network in a variational quantum eigensolver for efficient simulations.* New Journal of Physics 24, 033032 (2022). arXiv:2107.12387

[33] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G.H. Booth, J. Tennyson. *The variational quantum eigensolver: a review of methods and best practices.* Physics Reports 986, 1-128 (2022). arXiv:2111.05176

[34] Anna Schroeder, Matthias Heller, Mariami Gachechiladze *Deterministic Ansätze for the Measurement-based Variational Quantum Eigensolver.* arXiv:2312.13241

[35] Will Simmons. *Relating Measurement Patterns to Circuits via Pauli Flow.* arXiv:2109.05654

[36] Jiace Sun, Lixue Cheng, and Shi-Xin Zhang. *Stabilizer ground states for simulating quantum many-body physics: theory, algorithms, and applications.* arXiv:2403.08441v4 [quant-ph], 2024.