Classical simulation of quantum circuits based on stabilizer decompositions

A dissertation submitted by

Yifei Huang

In partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Physics

TUFTS UNIVERSITY

May 2021

Advisor: Peter Love

Abstract

In this dissertation we will describe various algorithms for simulating quantum circuits from the prospective of the Gottesman-Knill theorem. More specifically, we focus on several decompositions based on stabilizer states and how their l_0 norm could be used to characterize the cost of simulation algorithms we propose.

Our first work extends the Bravyi-Gosset algorithm to odd prime dimensions. We generalize their approximate stabilizer rank for weak simulation and bound the rank of the magic states by their stabilizer fidelity. We also relate the canonical form of qudit stabilizer states to Gauss sum evaluations and give an $O(n^3)$ algorithm for calculating the inner product of two n-qudit stabilizer states.

In the second work, we propose a classical simulation method based on decomposing unitary gates into a sum of stabilizer projectors. By only decomposing the non-Clifford gates, we take advantage of the Gottesman-Knill theorem and build a bridge between stabilizer-based simulation and Feynman-path simulation. We give two variants of this method: stabilizer-based path integral recursion (SPIR) and stabilizer projector contraction (SPC). The SPIR method improves the previous bound of recursive Feynman simulation from $O((2d)^n)$ to $O((2d_{nc})^n)$, where d_{nc} is the non-Clifford depth of the circuit.

Our third work combines teleportation-inspired method for shallow circuits with the SPC algorithm we proposed. Teleportation-inspired simulation corresponds to a different way of contracting the tensor network of a circuit, which we will call turning circuit sideways. For our SPC algorithm, unitary gates turned sideways may not be unitary, therefore, the stabilizer projector rank may not be full rank, thus improving upon the $O(2^{\tilde{n}})$ scaling of the original teleportation-based method. By

using SPC algorithm on its sideways circuits, we obtain a time cost of $O(2^{(\tilde{n}-\tilde{n}_t/2)})$ to calculate a single amplitude, where $\tilde{n}$ is the number of qubits in the sideways circuit and $\tilde{n}_t$ is the minimal number of sideways T gates (or any sideways diagonal non-Clifford gates) in the non-Clifford layers of the sideways circuit. The fact the the more T gates there are, the faster our algorithm is, benefits from incorporating monotone simulation methods, which takes advantage of diagonal gates. We also discuss the trade-off for the scaling when circuits are turned sideways differently.

To my parents and my beloved wife Wanyue.

Acknowledgments

The work described in this dissertation is the result of many people's helpful advice, useful suggestion and enthusiastic contribution. First of all, my advisor Peter Love, who has been supporting me without any reservation throughout the 5 years I have been in the group. Under his supervision, he has provided me with tremendous guidance in terms of science, writing, presentation and so on. Meanwhile, I have had the absolute freedom to work on problems I am interested in and the way I want to. Next I would like to thank Lucas Kocia, who helped me so much while I was starting to engage in research activities. With their help, my scientific skills have become much more mature in every aspect.

I would like to thank David Gosset for being on my committee and making very helpful suggestions for revision of the draft. It is an incredible experience to have someone who I cite all the time reading my thesis work and commenting on it. I want to thank my other committee members: Cristian Staii, Krzysztof Sliwa and Gary Goldstein, for showing great interests, asking great questions and making useful comments about my work. I have had classes with each one of them, they are all inspiring teachers in their own way.

I would also like to thank the group members for discussing ideas and enlarge and deepen my understanding of the field, especially Juspreet Singh Sandhu, Will Kirby, Andrew Tranter and Michael Kreshchuk. It has been a wonderful journey to see the group doubles its size for 4 years in a row.

I'd also like to thank my roommates, fellow grad students, my college friends who I have been keeping in touch with, friends I met on the tennis courts, and others who helped me get to grad school, helped me in grad school, or simply managed to put up with me while I was in graduate school. It has been a long journey, they

have made it fun and easier for me.

Yifei Huang

TUFTS UNIVERSITY

May 2021

Contents

Abstract ii

Acknowledgments v

List of Tables xi

List of Figures xii

Chapter 1 Introduction 1

1.1 Qubits, gates and quantum circuits 2

1.2 Gottesman-Knill theorem 3

1.2.1 Pauli matrices and stabilizer states 3

1.2.2 Clifford gates and the Clifford Hierarchy 4

1.2.3 Updates to the stabilizer generators regarding computational-basis measurements 6

1.2.4 Aaronson-Gottesman Tableau algorithm 8

1.3 Strong simulation and weak simulation 9

1.3.1 Strong Simulation 10

1.3.2 Weak Simulation 10

1.4 Schrodinger simulation vs. Feynman simulation and simulation algorithms in the monotone class 13

1.4.1 Calculating single amplitude vs. the whole wave function . . 13

1.4.2 Recursive Feynman path-integral algorithm 14

1.4.3 Feynman-Schrodinger Hybrid simulation 15

1.4.4 Tensor network diagrams 16
1.5 Classical Simulation algorithm based on stabilizer rank 18
1.5.1 Strong Simulation for qubits 18
1.5.2 Sampling from a resource initial state and Clifford circuit . . 21
1.6 Stabilizer Projector decomposition of density matrices and Robustness of Magic 22
1.7 Objectives and outline of the dissertation 24

Chapter 2 Weak simulation of Clifford-dominated circuits and approximate stabilizer rank for qudits 26

2.1 Qudit Pauli group and Clifford gates 27
2.2 Qudit Magic states and T gates 31
2.2.1 Qudit T gates 32
2.2.2 Qudit Magic states 36
2.3 Qudit T gate Gadget 38
2.4 Canonical forms for Qudit stabilizer states and inner product algorithm 39
2.4.1 The inner product of two qudit stabilizer states 44
2.5 Nonorthogonal decompositions of qudit Magic states 46
2.6 Weak Simulation and Approximate Stabilizer Rank 48
2.7 Detailed calculations of $\phi(p, d)$ and $Z(\mathcal{L})$ 51
2.7.1 Magnitude and phase of magic state inner product with orbit representatives of nonorthogonal decompositions 51
2.7.2 Evaluation of $Z(\mathcal{L})$ 53

Chapter 3 Feynman path-integral Simulation with stabilizer projector decomposition of unitaries 56

3.1 l_0 measure for stabilizer pseudomixture 59
3.2 Stabilizer Projector decomposition for unitaries 61
3.3 Algorithms 63
3.3.1 Polynomial-space SPIR method 63
3.3.2 Exponential-space SPC method 65

3.3.3 Combination of SPC method with Feynman-Schrodinger hybrid algorithm algorithm 67
3.3.4 Combination of SPC method with Sum-over-Cliffords algorithm 68
3.4 Comparison to Bravyi-Gosset algorithm for a family of circuits . . . 69
3.5 Quantum supremacy experiment 71

Chapter 4 Improving teleportation-based simulation by stabilizer-based method on sideways gates 74

4.1 Stabilizer projector algorithm for sideways circuits 77
4.1.1 Clifford properties of sideways gates 78
4.1.2 Time scaling based on stabilizer projector rank for non-unitary sideways gates 80
4.2 Further reduction of stabilizer projector rank 83
4.2.1 Reduction for single-qubit diagonal Clifford gates 84
4.2.2 Reduction for two-qubit non-diagonal gates 85
4.3 Sideways simulation for different gate sequences: parallel vs. sequential 86

Chapter 5 Conclusions and open problems 91

5.1 Summary of contributions 91
5.1.1 Qudit Bravyi-Gosset weak simulation 91
5.1.2 Boosting recursive Feynman path-integral simulation by stabilizer projector decomposition 92
5.1.3 Taking advantage of sideways gates in stabilizer projector simulation 93
5.2 Future directions 94
5.2.1 Obtaining sparse decomposition by l_1 minimization 94
5.2.2 Approximate simulation with the stabilizer projector methods 95

Appendices 97

Appendix A More SPM decompositions of various resource states 98

Appendix B Conventions and definitions for stabilizer states **100**

Appendix C Gates in the Google supremacy experiment and their Sum-over-Clifford decompositions **101**

Appendix D Stabilizer projector decomposition for non-Clifford gates in the Google supremacy experiment **103**

Appendix E Stabilizer projector rank for $\sqrt{W} \otimes \sqrt{W}$ **106**

Bibliography **108**

List of Tables

1.1 Actions of the generators of Clifford gates on Pauli operators. 6

2.1 The matrices M_d, optimal value of p and approximate stabilizer rank scaling comparison for $d = 2, 3, 5, 7$. Here $\kappa = -2 \log_d \alpha$ so that $d^{\kappa t} = \alpha^{-2t}$. Here the ω for $d = 5$ and $d = 7$ rows are $e^{2\pi i/5}$ and $e^{2\pi i/7}$ respectively. 50

3.1 Upper bounds of SPM rank/RoM squared for various resource states. States $\left|\tilde{M}_k\right\rangle = \cos\left(\frac{\pi}{2^k}\right)|0\rangle + \sin\frac{\pi}{2^k}|1\rangle$ and therefore $|H\rangle = \left|\tilde{M}_3\right\rangle$. . . 60

3.2 Upper bounds on stabilizer projector rank and probability of occurrence among all two-qubit gates for the non-Clifford gates in the Google supremacy experiment. See Appendix D for one of the decompositions achieving these upper bounds for each gate. 63

3.3 Simulation cost for algorithms discussed in the introduction and developed in this chapter. Here, m is the total number of gates, x is the number of entangling gates across the patches, d is the circuit depth, t is the number of non-Clifford gates in the circuit, 2^k is the stabilizer projector rank for each layer of gates, where $k \sim n$, and d_{nc} is the non-Clifford gate-depth. $tw(G)$ is the treewidth of the undirected graph corresponding to the circuit [MS08]. cw is the contraction width corresponding to a certain contraction order of the tensor, which is defined as the size of the biggest clique formed along the contraction. Big-O notation is implicit in the table. 68

List of Figures

1.1 Circuit diagram for the three generators of the Clifford group. (a). Hadamard gate. (b) Phase gate. (c) Control-NOT (CNOT) gate. (d) Control-Z (CZ) gate. 6

1.2 Single-qubit and two-qubit diagonal and non-diagonal gates turned into tensor diagrams. (a) Single-qubit diagonal gate diagram. (b) Two-qubit diagonal gate diagram. (c) Single-qubit non-diagonal gate diagram. (d) Two-qubit non-diagonal gate diagram. 17

1.3 Gadget to implement a T-gate using an ancilla magic state $|A\rangle$ as defined in [ZLC00]. Using this gadget, universal quantum computation (UQC) can be achieved using a Clifford circuit with injected magic states. 19

2.1 Gadget for qudit U_v gate. 39

3.1 Alternating layers of Clifford and non-Clifford gates for implementing the stabilizer projector contraction (SPC) simulation. Each layer doesn't have to have constant depth. In fact for Clifford layers it doesn't matter because they can all be efficiently processed by the Gottesman-Knill theorem. If one run into a non-Clifford layer, one projects the stabilizer states obtained from the previous round onto the stabilizer projectors by taking inner products between stabilizer states [GMC12]. If a non-Clifford layer has depth bigger than 1, then one can always chop them further into thinner slices such that the gates in each slice can be efficiently written down as a sum of stabilizer projectors. 66

3.2 In this plot we consider the ensemble of random circuits with alternating rounds of single qubit and two qubit layers. We plot the threshold density (probability) of non-Clifford gates for the SPIR method to have a better scaling than the strong Bravyi-Gosset algorithm for such circuits of different non-Clifford depth as in eq. (3.17) and eq. (3.19). The two-qubit gates in comparison here are CZ and CS gates. Here the curve for CS is going negative for $d_{nc} > 16$, which means for any density of single-qubit non-Clifford gates, the scaling of the SPIR method is better than the strong Bravyi-Gosset algorithm. For $p = 1/3$ and CZ gates as two-qubit gates, the SPIR method starts to have a better scaling when the non-Clifford depth is bigger than 31. For $p = 1/3$ and CS gates, the SPIR method starts to have a better scaling when the non-Clifford depth is bigger than 11. 71

4.1 Turning a general two-qubit gate and a diagonal two-qubit gate sideways. One can see that turing a diagonal two-qubit gate sideways is equivalent to a single-qubit matrix with one input index and one output index. 75

4.2 Turning gates sideways at the input and output of the original circuit where indices are fixed. (a) The original circuit where input and output indices of the first qubit are all fixed to be 0. Using standard matrix multiplication rules, the matrix elements in the two single-qubit gates that are relevant to this amplitude are circled. (b) Sideways circuit with i being the input index and j being the output index. 76

4.3 Two overlapping sideways T gates. Here we use the same symmetric gate symbol as the CZ gate but colored in green to represent the sideways T gate. 83

4.4 For the original circuit in (a), one can follow the the arrow, which goes through the top qubit first, therefore indices i, m are the input indices of the sideways circuit. Then it touches indices j, s, which should be the outputs of the sideways U. Following the arrow, s goes on to be the input of sideway V, so does n. Following this analysis, we obtain the sideways circuit with indices as shown in (b). 87

4.5 (a) A 5-qubit supremacy-type circuit that has 5 cycles (depth 11). The arrows of different dark colors represent the qubit-flow of its sideways circuit. Here the two-qubit gates are all Control-Z gates. (b) The sideways circuit of (a) which has depth 23. Here the two-qubit gates of bright blue, red, green colors corresponds to $\sqrt{X}$, $\sqrt{Y}$ and T gates respectively. Notice that we omit the normalization factor for the Hadamard gate (control-Z turned sideways). A layer in this circuit is sequential in a way that the two-qubit gates are commingled with single-qubit gates, while the original circuit has cycles that is composed of parallel gates. 89

4.6 (a) The same 5-qubit supremacy-type circuit that has 5 cycles (depth 11). The arrows of different dark colors represents a different sideways qubit-flow from figure 4.6. The sideways qubits don't just run across space (qubits in the original circuit) but also time. (b) The sideways circuit of (a). Compared to 4.6, the sideways circuit here still has the parallel feature of the original circuit. The depth is reduced to 16. . 90

Chapter 1

Introduction

Quantum computing is an interdisciplinary field spanning physics, computer science and mathematics. Quantum computing explores how controlling and scaling up the microscopic physical apparatus in the lab, which are governed by the laws of quantum mechanics, can improve our understanding of the theory of computation and therefore increase our computing capabilities [Fey82, Sho99, Gro96, HHL09, AA11]. The field has undergone rapid development in the past twenty years, with remarkable progresses both theoretically and experimentally. Many experimental platforms, from superconducting qubits [WSB$^+$04, KTG$^+$07, AAB$^+$19] to ion traps [MMK$^+$95, LBMW03], from Rydberg atoms [SWM10, JCZ$^+$00] to photonic devices [KLM01, OFV09], have been steadily improving in terms of number of qubits and gates fidelity. At the moment, the field is in what is called the Noisy Intermediate Scale Quantum (NISQ) [Pre18] era. During this era, we have $10 \sim 100$ qubits and noisy gate operations that allow us to perform a sequence of constant number of operations before we lose the signal due to noise. During this era, classical simulation is of particular interest because it offers insight into when quantum computation is more powerful than classical computation as well as providing tools for benchmarking and certifying NISQ devices. In this introduction chapter, we will introduce the basics concepts of quantum computation and quantum circuits, then we will review several important classical simulation algorithms, especially the Gottesman-Knill theorem and the stabilizer rank algorithm. These algorithms will

be used later in the main chapters.

1.1 Qubits, gates and quantum circuits

A qubit, or a quantum bit, is the quantum version of a classical bit. A classical bit has states 0 or 1. A qubit can be a superposition of the 0 state and 1 state with complex coefficients. From now on, we will use the Dirac braket notation to represent quantum states. The computational basis states for a qubit are defined as follows:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{1.1}$$

A qubit $|\psi\rangle$ is in general $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. As we mentioned before, the coefficients α and β are complex values that are normalized to 1:

$$|\alpha|^2 + |\beta|^2 = 1. \tag{1.2}$$

For multi-qubit separable states, we write them as tensor products of single qubit states, e.g., $|0\rangle \otimes |0\rangle$. We often omit the tensor product symbol and write $|00\rangle$ for simplicity. For entangled states, we write them as a superposition of separable states, e.g., the Bell state is written as $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ [Bel64].

A quantum circuit starts from some fiducial state (usually $|0\rangle^{\otimes n}$), followed by a sequence of quantum gates, each of which is a unitary operation that satisfies $UU^\dagger = I$ and ends with measurements defined by some observable. In order to implement a quantum algorithm, one compiles a circuit into elementary gates, namely single-qubit and two-qubit gates. In practice, one has a fixed set of elementary gates to choose from instead of arbitrary single-qubit and two-qubit gates.

We usually denote the number of qubits in a quantum circuit as n. A circuit has depth d when the circuit can be divided into d layers where each layer has gates acting on disjoint sets of qubits.

1.2 Gottesman-Knill theorem

There are certain subsets of operations in the universal gate set that can be classically simulated efficiently. One prominent example are the Clifford operations, which includes initial fiducial states $|0\rangle^{\otimes n}$ and final measurements in the computational basis and unitaries generated by Hadamard, Phase and $CNOT$ gates. The Gottesman-Knill theorem states that Clifford operations can be efficiently simulated classically [Got98b].

1.2.1 Pauli matrices and stabilizer states

Pauli matrices are defined by (we add the identity matrix to complete the Pauli basis of two-dimensional Hermitian operators)

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{1.3}$$

The three non-identity Pauli matrices pairwise anti-commute while the product of two of the three non-identiy matrices commutes with the third. For multi-qubit Pauli operators, i.e., a tensor product of single-qubit Pauli matrices (including the identity), if two multi-qubit operators anti-commute on odd number of the qubits, they anti-commute, otherwise they commute.

We have $Z|0\rangle = |0\rangle$, $X|+\rangle = |+\rangle$ and $Y|+i\rangle = |+i\rangle$ for single-qubit states. Here we denote $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ and $|\pm i\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm i|1\rangle)$. Because these operators acting on the states yield eigenvalue $+1$, we call Z the stabilizer of $|0\rangle$, X the stabilizer of $|+\rangle$ and Y the stabilizer of $|+i\rangle$. In this sense any Hermitian operator with one of the eigenvalues being 1 can be called a stabilizer, but the stabilizers in literature specifically refer to Pauli operators that yield eigenvalue $+1$ for some states. These states are hence called the stabilizer states. Note the identity matrix stabilizes every state.

Single-qubit Pauli operators stabilize a single state, while multi-qubit Pauli operators stabilize a subspace of states. For example, $Z \otimes Z$ doesn't just stabilize

$|00\rangle$, it also stabilizer $|11\rangle$ and therefore all linear superposition of the two $\alpha|00\rangle + \beta|11\rangle$. In fact, there are 2^n Pauli operators that stabilize a single n-qubit stabilizer state and they form a group, which we call the stabilizer group. However, this group can be generated by n Pauli operators, which we call the stabilizer generating set. Therefore, we only need to write down n Pauli operators to uniquely specify an n-qubit stabilizer state. For example, the only two-qubit stabilizer state that is stabilized by $Z \otimes I$ and $I \otimes Z$ is $|00\rangle$. Note the generating set of this group is not unique, i.e., two different stabilizer generating set could generate the same stabilizer group. For example, the stabilizer group that stabilizes $|00\rangle$ has four elements, $I \otimes I$, $Z \otimes I$, $I \otimes Z$ and $Z \otimes Z$. Not only $Z \otimes I$ and $I \otimes Z$ can generate this group but also $Z \otimes I$ and $Z \otimes Z$, or $I \otimes Z$ and $Z \otimes Z$. The specific choice of the generating set doesn't matter. For simplicity, we will denote $Z \otimes I$ as Z_1 and $I \otimes Z$ as Z_2 where the subscripts represents the qubit that Z acts non-trivially on, same for X and Y operators. This generalizes to multi-qubits. There are $2^{O(n^2)}$ n-qubit stabilizer states [AG04]. This can be enumerated by counting the total number of stabilizer generating sets and being divided by the number of ways one can form different generating sets out of the same stabilizer group. Calculating the inner product of two stabilizer states takes time $O(n^3)$ [AG04, GMC12].

An n-qubit stabilizer states has a canonical form [DDM03, Nes08]:

$$|\phi\rangle = \frac{1}{2^{k/2}} \sum_{u \in Z_2^k} i^{l(v)} (-1)^{q(v)} |v = Ru + t\rangle \tag{1.4}$$

where $u \in Z_2^k$, $v, t \in Z_2^n$ and R is a full column rank $n \times k$ matrix with its entries being 0 or 1. Function $q(v) = \sum q_{ij} v_i v_j + q_i v_i$ is a quadratic function and $l(v) = \sum l_i v_i$ is a linear function of the bits of v .

1.2.2 Clifford gates and the Clifford Hierarchy

The Clifford hierarchy was defined by Gottesman and Chuang in [GC99]:

$$\mathcal{C}(k+1) = \left\{ U | UPU^\dagger \in \mathcal{C}(k), P \in \mathcal{P}(k \geq 0) \right\} \tag{1.5}$$

The first level of the Clifford hierarchy is the Pauli group $\mathcal{C}(1) = \mathcal{P}$. The Clifford group is the second level of the hierarchy, i.e., the Clifford gates are unitaries that map the Pauli group to itself. Note that the Clifford group contains the Pauli group. The three generators of the Clifford group are the Hadamard gate, the Phase gate and the CNOT gate

$$\begin{aligned} H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \\ CNOT = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \end{aligned} \tag{1.6}$$

One other gate that is commonly referred to is the Control-Z gate:

$$CZ = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z = H_2 \cdot CNOT \cdot H_2, \tag{1.7}$$

where H_2 refers to Hadamard gate acting on the second qubit. CZ gate will be discuss extensively in Chapter IV. The circuit diagrams for these gates are shown in Figure 1.1.

These gates map Pauli operators to Pauli operators when they act by conjugation. The stabilizer group after being acted on by these gates is simply another stabilizer group. Therefore we can simulate the action of these gates by updating the n stabilizer generators, which at most takes linear operations in terms of n.

Assume P stabilizes the state $|\psi\rangle$ before the update of Clifford gate C, the reason it is important to use stabilizers to build the equivalence between the state updates and stabilizers updates is

$$CPC^\dagger C|\psi\rangle = CP|\psi\rangle = C|\psi\rangle. \tag{1.8}$$

Therefore the updated stabilizer $CPC^\dagger$ stabilizes the updated state $C|\psi\rangle$. Hence up-

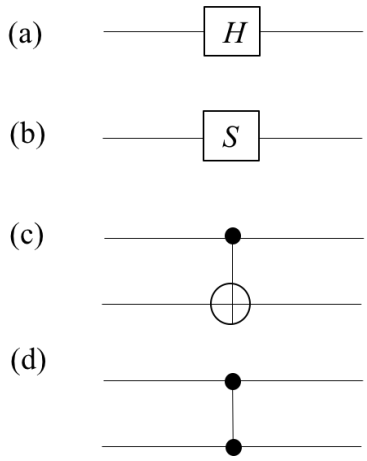

Figure 1.1: Circuit diagram for the three generators of the Clifford group. (a). Hadamard gate. (b) Phase gate. (c) Control-NOT (CNOT) gate. (d) Control-Z (CZ) gate.

Gate type	Action
Hadamard gate	$X \to Z$
	$Z \to X$
Phase gate	$X \to Y$
	$Z \to Z$
CNOT	$X \otimes I \to X \otimes X$
	$I \otimes X \to I \otimes X$
	$Z \otimes I \to Z \otimes I$
	$I \otimes Z \to Z \otimes Z$

Table 1.1: Actions of the generators of Clifford gates on Pauli operators.

dating the stabilizer generators is equivalent to updating the state in the Schrodinger picture. The actions of the Clifford generators on Pauli matrices are listed in Table 1.1.

1.2.3 Updates to the stabilizer generators regarding computational-basis measurements

Measurements in the computational basis measure observables Z_i s, where Z_i only acts non-trivially on qubit i. There are two cases determining the measurement outcome of measuring a stabilizer state. The two cases depend upon the commutative property of the stabilizer generators with the measurement operator Z_i. In order to check whether the stabilizer generators commute with the Z_i, it is enough

to check the Pauli operator on the ith qubit for all n generators because Z_i only acts non-trivially on the ith qubit. This procedure take $O(n)$ operations.

The first case is when the measurement observable Z_i anti-commmutes with some of the Pauli generators. If measurement Z_i anti-commutes with one of the generators, then a measurement will give a random outcome because the stabilizer state before the measurement is neither stabilized by Z_i or $-Z_i$. After the measurement the state will be stabilized by either of these two operators, we simply replace the generator that anti-commutes with Z_i or $-Z_i$ with 50% probability each. The new set of generators represent the state after the measurement and the sampled outcome of the measurement will be ± 1 with equal probability. If the measurement anti-commutes with more than one of the generators, one can reduce to the previous situation. There are 2^n Pauli operators in the stabilizer group overall and the choice of the n generators is not unique. By the properties of group, operations that multiply one of the generators by another does not change the group the whole generating set represents. If two generators both anti-commute with Z_i, they have either X or Y on the ith qubit and their product is either Z or identity on the qubit and thus their product commutes with Z_i. Therefore one can choose one of those generators and multiply it to all others that anti-commute with Z_i. This only leaves the one generator we choose that anti-commutes with Z_i and we are back to the previous situation. This procedure takes $O(n^2)$ time because in the worst case scenario, $n-1$ multiplications are needed to eliminate all other generators that anti-commute with Z_i and each multiplication takes $O(n)$ time.

The second case is when all stabilizer generators commute with Z_i. In this case there is no need to update the generators, but one has to compute the measurement outcome by finding the sign of Z_i in the stabilizer group. Z_i is not necessarily in the stabilizer generators. Therefore, one have to use multiplications of the generators to obtain Z_i or $-Z_i$. This procedure is essentially the same as the procedure of Gaussian elimination, which takes $O(n^3)$ time in practice. One can improve this scaling further to $O(n^2)$ [AG04]. Now we will present the improved algorithm in Section 1.2.4.

1.2.4 Aaronson-Gottesman Tableau algorithm

Destabilizers [AG04] are a set of Pauli operators that have the following properties:

- There are exactly n destabilizers.
- They all commute with each other.
- They generate the whole Pauli group together with the stabilizer generators.
- The i-th destabilizer anti-commutes with the i-th stabilizer but commutes with all other stabilizer generators.

One can see $\{X_1, ..., X_n\}$ is the set of destabilizers for initial state $|0^n\rangle$. Therefore one can update the destabilizer together with the stabilizers throughout the Clifford gates of the circuit and doing this only adds a factor of 2 to the cost.

With the destabilizers, one can compute the outcome of the measurement Z_i which commutes with all stabilizers in $O(n^2)$ time [AG04]. One only need to check through the destabilizers and multiply together the stabilizer generators whose corresponding destabililizers anti-commute with Z_i. The time scaling for the check is $O(n)$ and $O(n^2)$ for the multiplication. Now we will show why this gives the element Z_i (or $-Z_i$) in the stabilizer group. We now assume the stabilizer generators being $g_1, g_2, ..., g_n$ and destabilizer generators being $g'_1, g'_2, ..., g'_n$, and measurement $+Z_i$ or $-Z_i$ is obtained by

$$\pm Z_i = \prod_{j=1}^{n} (g_j)^{c_j} \tag{1.9}$$

where c_k is 0 or 1.

Theorem 1.2.1. *For k such that $\{g'_k, Z_i\} = 0$, $c_k = 1$; otherwise $c_k = 0$.*

Proof:

$$\{g'_k, Z_i\} = \{g'_k, \prod_{j=1}^{n} (g_j)^{c_j}\} = \prod_{j=1, j\neq k}^{n} (g_j)^{c_j} \{g'_k, g_k^{c_k}\} = 0 \tag{1.10}$$

The last equality only holds if $\{g'_k, g_k^{c_k}\} = 0$, i.e., $c_k = 1$.

1.3 Strong simulation and weak simulation

Born's rule gives the probability distribution for the measurement outcomes of a quantum system. When trying to simulate a quantum circuit on a classical computer, there are two types of simulation one can consider: One is to output the probability distribution according to Born's rule:

$$p(x) = \left|\langle x| U \left|0^{\otimes n}\right\rangle\right|^2 \tag{1.11}$$

where the U is the unitary operator that implements the circuit and $x \in F_2^k$ is the string that represents the measurement outcome on k of the n qubits; the other gives samples instead of a probability distribution. The former notion of simulation is called strong simulation and the latter is called weak simulation. In fact, what a quantum computer does is the latter. One can only learn about a quantum system by making measurements and obtaining samples. The complexity class that represents the set of decision problems solvable by quantum computers in polynomial time with bounded error is called BQP (Bounded-error Quantum Polynomial time). This is the complexity class weak simulation tackles. Strong simulation is at least as hard as the weak simulation because one can simply use inversion sampling to output samples accordingly if they have full knowledge about the output probability distribution, or one can use rejection sampling if the probability distribution is anti-concentrated [Dev]. In fact, we will discuss shortly, strong simulation is $\#P$-hard [Nes08, AA11], while the class BQP, is believed to be strictly smaller than $\#P$-hard. Here $\#P$ is the complexity class the count the number of solutions to a decision problem, where each solution to the problem takes at most polynomial time to verify. $\#P$-hard stands for the problems that is as least as hard as the hardest problems in $\#P$.

1.3.1 Strong Simulation

According to the results of Boson Sampling [AA11], strong simulation is known to be $\#P-$hard. Boson Sampling is a restricted subset of linear optics operations. To calculate the probability distribution of the outputs for Boson sampling, i.e., simulating it in a strong sense on a classical computer, is equivalent to calculating the permanent of a matrix. This is known to be $\#P-$hard [AA11]. While full linear optics is know to be universal for quantum computation [KLM01], the strong simulation of Universal Quantum Computation (UQC) is at least as hard as the strong simulation of Boson sampling. Therefore, strong simulation of UQC is $\#P-$hard. In fact, it has been proven that the strong simulation of constant depth quantum circuit and another restricted subset of UQC, called Instantaneous Quantum Polynmial (IQP) circuits, are hard unless some very implausible complexity collapse happens [TD02, BJS11].

ϵ-strong simulation is an approximation of strong simulation with an multiplicative error ϵ. Namely,

$$(1-\epsilon)p(x) \leq p^*(x) \leq (1+\epsilon)p(x) \tag{1.12}$$

where $p^*(x)$ is the output of the $\epsilon-$strong simulator that gives the estimation of the probability of measuring a given string x.

But even approximating the probability distribution of the Boson sampling problem within multiplicative error is known to be $\#P-$hard [AA11], as well as for IQP circuits [BJS11]. Therefore, doing ϵ-strong Simulation is still $\#P-$hard.

1.3.2 Weak Simulation

The task for a weak simulator is to output a set of samples from the probability distribution generated by the quantum circuit. The distinction between strong simulation and weak simulation is not completely understood. Therefore it is not known to what extent one need approximate the probability distribution such that it is just enough to for the weak simulator to generate samples that are indistinguishable

from the samples of the quantum device. In fact, sampling is shown to be hard if the exact distribution $|\langle x| U |0^n\rangle|^2$ or a distribution with multiplicative error is $\#P$-hard to compute [Sto83, Tod91, TD02, BJS11, AG04]. However, there are several ways to weaken the condition in eq. (1.12) for the purpose of weak simulation, which we discuss in this section.

First, our approximation $\hat{p}(x)$ has to be 0 if the actual $p(x) = 0$, according to the multiplicative approximation in eq. (1.12). This is not necessary for weak simulation. We assume the number of samples required for the weak simulator is polynomial in the number of qubits. In this case it maybe enough to spoof a verification test by sampling from a probability distribution that approximates the real probabilities better on the high probability (heavy) outputs but more poorly on the low probability outputs. Hence, our estimation $\hat{p}(x)$ for $p(x) = 0$ does not need to be exactly 0 but only needs to be small enough. Therefore we have the following weaker condition compared to eq. (1.12):

$$|p(x) - \hat{p}(x)| < \epsilon_1 \sim O\left(\frac{1}{\exp(n)}\right). \tag{1.13}$$

What's more, one may not even need to approximate all high probability outputs well, but to approximate a majority of them (more than half) better. For example, a total variational distance approximation suffices in most cases:

$$\sum_x |p(x) - \hat{p}(x)| < \epsilon'. \tag{1.14}$$

If there are only polynomial number of samples output by the quantum device and the output probability distribution only has non-negligible support on polynomial number of strings, then by the Chernoff–Hoeffding bound, one can only obtain information about the probabilities with polynomial precision [Nes09]. Therefore it is enough to have a weak simulator that approximates the probabilities to the following degree:

$$|p(x) - \hat{p}(x)| < \epsilon_2 \sim O\left(\frac{1}{\text{poly}(n)}\right). \tag{1.15}$$

This condition is much weaker than that of eq. (1.14) and (1.13). Anti-concentrated output probability distribution satisfies

$$\Pr(|\langle x| U |0^n\rangle|^2 \geq \frac{\alpha}{2^n}) > \beta, \tag{1.16}$$

where α, β are some constant not related to n. This distribution has non-negligible supports on exponential number of outputs. For quantum circuits that have this property, constraint (1.15) alone is not enough for a faithful weak simulator. For example, we can imagine a distribution where almost all probabilities are exponentially small. The approximation in eq. (1.15) may give us a distribution that only has supports on a polynomial number of outputs, say $q(n)$. Assume we sample a set of outputs that has size $10q(n)$ according to this distribution, which is still polynomial, we will see many repetitions in the sampled outputs. When we sample from the actual quantum circuit, we will almost see no repetition because the actual probability distribution is anti-concentrated.

As we mentioned before, we care more about the high probability outputs for the purpose of weak simulation. Therefore if we know the shape of the probability distribution, one could determine l threshold probability values $0 \leq y_1 \leq y_2 \leq ... \leq y_l \leq 1$ and it may be enough in most cases to specify what range $p(x)$ is in among these y_is for most probability distributions. For example, for an anti-concentrated probability distribution, if may suffice to know whether the probability $p(x)$ is bigger or smaller than the median probability of $p(x)$s for all xs [AC17].

One can see there are several different notions and different conditions for the approximation of $p(x)$ for weak simulation. For some weak conditions like eq. (1.15), the accuracy of the weak simulator depends on the shape of the actual distribution itself.

1.4 Schrodinger simulation vs. Feynman simulation and simulation algorithms in the monotone class

Schrodinger simulation, also known as direct simulation, of n-qubit quantum evolution stores the whole wave function and evolves it by applying unitary operators. Direct simulation requires memory $O(2^n)$ and time $O(m2^n)$, where m is the number of gates in the circuit. We refer to this as the Schrodinger approach in accordance with [AAB+19]. In this section, we will also briefly review the recursive Feynman path-integral simulation algorithm and the Bravyi-Gosset algorithm for simulating quantum circuits, as they will be relevant for our later discussion.

The width w_i of each layer will be the number of qubits that are acted on by gates in each layer. If the width for all layers is $O(w)$, then the number of gates m will be $O(dw)$. Even for shallow NISQ devices, we know that the depth of a quantum circuit at least needs to be more than 2 to be hard for classical computers to simulate [TD02]. Assuming the width for all layers is $O(n)$, then the depth-3 circuit that is hard for classical simulation has $m \approx O(3n)$ gates. Therefore, we have $m > n$ in general. In this thesis, we also define the non-Clifford depth d_{nc}, which we will use in Chapter III. We divide the circuit into Clifford layers and non-Clifford layers and the number of non-Clifford layers is the non-Clifford depth d_{nc} in that context.

1.4.1 Calculating single amplitude vs. the whole wave function

Storing the whole wave function is beneficial if one wants to know amplitudes $\langle x| U |0^n\rangle$ for all x. However, for many case of the application of classical simulation, this is not necessary.

The linear cross entropy benchmarking (XEB) scheme [BIS+18, AAB+19] calculates the following benchmark for their device:

$$\mathcal{F}_{XEB} = 2^n \sum_{x_i \in O} p(x_i)/|O| \tag{1.17}$$

where O denotes the set of output strings for actual measurements in experiment. One can see here the task required for classical simulation is to obtain amplitudes for a certain set of outputs instead of all output strings. The number of outputs in this set doesn't scale with 2^n, but depends on the operation time and depth of the circuit in actual experimental platforms. Again we take the supremacy experiment as an example: the number of samples taken is 1 million, which is much smaller than 2^{53}. Moreover, for many quantum algorithms, one doesn't measurement the whole joint probability distribution at the end of the circuits, instead, only marginal distribution is needed. Sometimes only one qubit is measured.

For the tasks mentioned above, it is not worth it to keep track of the whole wave function, instead, algorithms that calculate single amplitudes per run are sufficient. Here in this thesis, more specifically the stabilizer path-integral recursive (SPIR) method we will present in Chapter III and the main algorithm in Chapter IV, both belong to this class of simulation.

1.4.2 Recursive Feynman path-integral algorithm

The Feynman approach calculates the amplitude of obtaining outcome of string x using the path-integral representation of a circuit:

$$\begin{aligned} &\langle x| U_m U_{m-1} ... U_1 |0^n\rangle \\ &= \sum_{j_1,\ldots,j_{m-1}} \langle x| U_m |j_{m-1}\rangle \langle j_{m-1}| U_{m-1} |j_{m-2}\rangle ... \langle j_1| U_1 |0^n\rangle \end{aligned} \tag{1.18}$$

This method requires memory $O(m+n)$, because after calculating each amplitude, one only needs to add the final result to the sum. However, $O(4^m)$ operations are required to process all possible paths. Because m is almost always bigger than n, the number of operations $O(4^m)$ here is larger than that of the Schrodinger approach.

To reduce the time cost for the Feynman approach, one can notice that there are a large number of repeated calculations in eq. (1.18) if we calculate the terms in the sum path by path. For example, if two of the paths are the same for the first $m-2$ steps, but only differ at the last step. In eq. (1.18), these two paths will give

two terms that only differ on the first factor of the term, $\langle x| U_m |j_{m-1}\rangle$. However, naive evaluation of eq. (1.18) results in redundant computation of the first $m-1$ factors.

As discussed in [AC17], one can avoid this repeated calculation as follows. By first slicing a circuit into two sub-circuits first, C_1 and C_2:

$$\langle x| C |0^n\rangle = \sum_{y\in\{0,1\}^n} \langle x| C_2 |y\rangle \langle y| C_1 |0^n\rangle . \tag{1.19}$$

one can obtain a recursion relation for the time cost for calculating the whole sum from the results of the two sub-circuits:

$$T(d) = 2^{n+1}T(d/2) \tag{1.20}$$

assuming the depth of the whole circuit is d and the depth of both C_1 and C_2 is $d/2$. Following this relation, one can recursively divide the two sub-circuits further until getting down to single-layer circuits. In this way, one can calculate the sum more efficiently than the Feynman approach, without sacrificing much of the space cost advantages of the Feynman approach [AC17]. In fact, we only need $O(\log d)$ steps to reach the leaf level (single layer) with recursion, and at each step we need a n-bit string y to label the term we are trying to compute. Therefore, we need $O(n \log d)$ space to recursively return a single term to the whole summation. Meanwhile, one can see the total time cost is brought down to $O(n2^{n \log d})$ by solving the recursion relation above. We will use the idea of this algorithm later when we apply this recursion to our algorithm in Chapter III.

1.4.3 Feynman-Schrodinger Hybrid simulation

The Schrodinger approach requires exponential space due to the need to store the entire wave function. One can take advantage of limited entanglement to reduce the space required [CZX$^+$18, MFIB18]. This method is called the Feynman-Schrodinger hybrid algorithm.

If one divides the circuit into two sub-circuits and ignores the entangling gates connecting the sub-circuits, one can reduce the memory cost to $O(2^{n/2})$. How can we "ignore" the entangling gates between the patches? For every such gate, one performs a Schmidt decomposition of the gate, e.g., decomposing the control-Z gate as follows [CZX+18, MFIB18]:

$$CZ = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z. \tag{1.21}$$

If there are x entangling gates between the two patches, there will be 2^x patched circuit configurations. Now the space cost of running Schrodinger's approach on every patch becomes $O(2^{n/2})$. Afterwards, one could establish a Feynman path summation with 2^x terms from the results of the Schrodinger simulation on the patched circuits, where the number of operations required is now 2^x. Therefore, the total number of operations becomes $2^{(x+n/2)}$, which is better than the direct simulation time scaling 2^n if the connectivity between the two patches is low enough such that $x < n/2$. While the number of operations required by the hybrid algorithm is greater than direct simulation if $x > n/2$, the memory cost is brought down to $2^{n/2}$.

1.4.4 Tensor network diagrams

A quantum circuit can be regarded as a tensor network [MS08]. By mapping all of the gates and therefore the free indices of eq. (1.18), one obtains an undirected graph whose vertices correspond to the free indices and the edges correspond to the gate structure relating these indices [BISN17, CZH+18, HZN+20]. The mappings for general single-qubit and two-qubit gates are illustrated in Figure 1.2. In eq. (1.18), if some gate U_r is diagonal, then the only chance $\langle j_r| U_r |j_{r-1}\rangle$ is non-zero is having $j_r = j_{r-1}$. Therefore, the effective number of free indices is decreased by 1. This is the idea that is visualized in Figure 1.2 and what tensor network simulation takes advantage of. Later in Chapter IV, we will incorporate the advantage of tensor network simulation for general diagonal gates into our stabilizer-based algorithm.

For the tensor network approach, one maps a quantum circuit into a graph and contracting the vertices in a certain order to compute the amplitude in eq. (1.18). The optimal contraction has a running time that is exponential in terms of a parameter called the treewidth of a graph, which in itself is a number that is NP-hard to compute [ACP87]. Therefore one optimize the contraction order empirically [GD12, BISN17, CZH+18, HZN+20, PZ21].

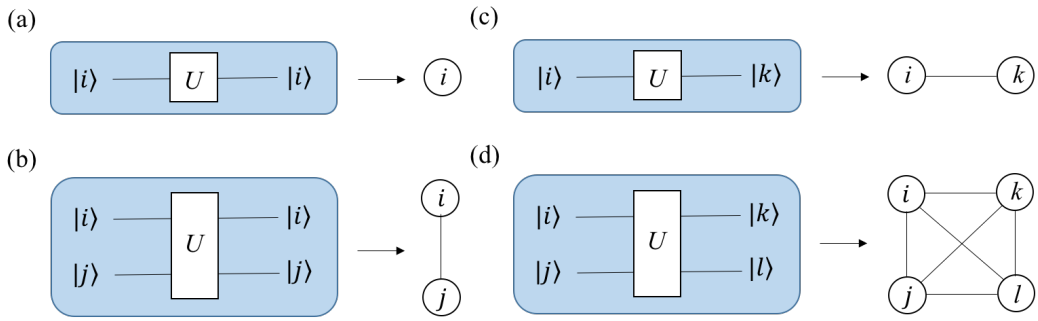

Figure 1.2: Single-qubit and two-qubit diagonal and non-diagonal gates turned into tensor diagrams. (a) Single-qubit diagonal gate diagram. (b) Two-qubit diagonal gate diagram. (c) Single-qubit non-diagonal gate diagram. (d) Two-qubit non-diagonal gate diagram.

Schrodinger simulation, (recursive) Feynman path-integral simulation, the hybrid simulation and the tensor network simulation all belong to a bigger class of simulation methods called the monotone methods [HNS20]. Algorithms in this class only care about the number and locations of non-zero elements in the matrices that describe the gates in the circuit, not what values these elements take. This means they will not be sensitive to gate and circuit structures, e.g., Clifford gates. An unconditional lower bound on the time scaling for this class of method is obtained as $(n-2)(2^{n-3}-1)$ [HNS20]. In the following section we review another class of algorithms outside of monotone methods which works better for circuits dominated by Clifford circuits.

1.5 Classical Simulation algorithm based on stabilizer rank

Clifford operations plus one non-Clifford gate are universal for quantum computation. One such non-Clifford gate is the T gate, which is defined by:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}. \tag{1.22}$$

Therefore, Clifford+T circuits cannot be efficiently simulated unless $P = BQP$, where P stands for decision problems solvable in polynomial time on a classical computer. An algorithm for simulating Clifford+T circuits with cost scaling exponentially with the number of T gates but polynomially with the number of qubits is the Bravyi-Gosset algorithm [BG16]. The algorithm is based on the stabilizer rank [BSS16] of resource states called the magic state [BK05]:

$$|A\rangle^{\otimes t} = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle) = (T|+\rangle)^{\otimes t}. \tag{1.23}$$

The T gates are implemented by magic state injection of these magic states. Bravyi and Gosset gave algorithms for both weak and strong simulation. The strong simulation version of the Bravyi-Gosset algorithm has time cost that scales linearly with the stabilizer rank of $|A\rangle^{\otimes t}$. This rank is upper bounded by $2^{0.47t}$.

In this section, we give a summary of a type of classical simulation method outside of the monotone methods. This methods aims at Clifford-dominated circuits with a few non-Clifford gates, which we will refer to as the Bravyi-Gosset algorithm, is based on a measure called *stabilizer rank*.

1.5.1 Strong Simulation for qubits

Let t be the number of T gates in the n-qubit quantum circuit we wish to classically simulate. The first step is to replace every T gate in the circuit by Clifford gates and an ancilla input of a magic state $|A\rangle$. This is accomplished using the gadget

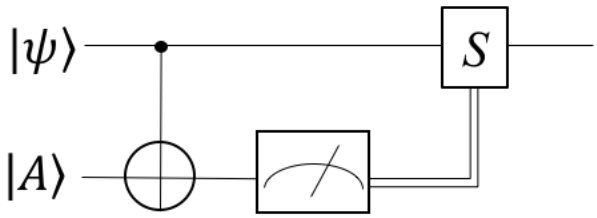

Figure 1.3: Gadget to implement a T-gate using an ancilla magic state $|A\rangle$ as defined in [ZLC00]. Using this gadget, universal quantum computation (UQC) can be achieved using a Clifford circuit with injected magic states.

shown in Figure 1.3 [ZLC00]. The number of ancilla qubits is t. We consider an initial state $|0^{\otimes n}\rangle$ for the Clifford+T circuit and $|0^{\otimes n}\rangle \otimes \left|A^{\otimes t}\right\rangle$ for the gadgetized circuit.

At the end of the computation we will measure a subset of w of the n qubits in the logical basis. This measurement with outcome x (where x is a bitstring of length w), postselected to the case where all ancilla measurements have result 0, is represented by a projector $\Pi = |x\rangle\langle x| \otimes I \otimes \left|0^t\right\rangle\left\langle 0^t\right|$. The strong simulation algorithm classically computes the probability of this measurement outcome after acting with a Clifford circuit V, which is our original (non-Clifford) circuit with all T-gates replaced by the gadget of Figure 1.3. Therefore we can express the probability of obtaining output x as:

$$P(x) = 2^t \left\langle 0^n A^t \right| V^\dagger \Pi V \left| 0^n A^t \right\rangle . \tag{1.24}$$

The factor of 2^t here compensates for the fact that we postselected on the measurement outcomes of the t ancilla qubits.

We define a t-qubit projection operator $\Pi_G = \langle 0^n| V^\dagger \Pi V |0^n\rangle$. This projector

maps states onto a stabilizer subspace. Then eq.(1.24) becomes

$$P(x) = 2^t \left\langle 0^n A^t \right| V^\dagger \Pi V \left| 0^n A^t \right\rangle = 2^{-u} \left\langle A^t \right| \Pi_G \left| A^t \right\rangle . \tag{1.25}$$

where u is an integer that depends on the number of qubits we are measuring out of n and the dimension of the stabilizer subspace Π_G is mapping onto.

If we can expand $\left|A^t\right\rangle$ into a sum of stabilizer states, then we can express $P(x)$ as a sum of inner products of t-qubit stabilizer states, which can be computed in $O(t^3)$ time ([AG04, GMC12, BSS16, BG16]). The fewer stabilizer states in the expansion of $\left|A^t\right\rangle$, the more efficient the algorithm is.

Stabilizer rank is defined as the minimal number of stabilizer states needed to write a pure state as a linear combination of stabilizer states:

$$\chi(|\psi\rangle) = \min\{k : \psi = \sum_{j=1}^{k} c_j \, |\phi_j\rangle\}. \tag{1.26}$$

where $|\phi_j\rangle$s are stabilizer states. The value of $\chi(t)$ of a t-qubit state is trivially upper bounded by 2^t because logical basis states are stabilizer states, and is also believed to be lower bounded by an exponential in t. For practical purposes we can achieve progress through a series of constructive upper bounds.

In [BSS16], Bravyi *et al.* found a stabilizer rank upper bound by obtaining $\chi_A(6) \leq 7$ for $\left|A^6\right\rangle$ and dividing the t-qubit state into a product of 6-qubit states. Therefore, $\chi_A(t)$ has a upper bound $7^{t/6} \simeq 2^{0.47t}$.

If we denote the stabilizer rank for the tensor product of t single-qubit magic states $\left|A^t\right\rangle$ as $\chi_A(t)$, the cost of classically computing $P(x)$ by taking inner products as described above is $O(t^3 \chi_A(t)^2)$.

The quadratic dependence on stabilizer rank can be improved by a Monte Carlo method, developed by Bravyi and Gosset, to approximate the norm of a tensor product of magic states projected on a stabilizer subspace:

$$\left| \left\langle A^t \right| \Pi_G \left| A^t \right\rangle \right| = \left\| \Pi_G \left| A^t \right\rangle \right\|^2 = \|\psi\|^2 \tag{1.27}$$

therefore enabling one to calculate $P(x)$ with cost $O(t^3\chi_A(t))$, linear in stabilizer rank. This concludes our summary of the strong simulation algorithm of Bravyi and Gosset.

1.5.2 Sampling from a resource initial state and Clifford circuit

The key advantage of weak simulation is that one can sample from a $\tilde{P}_{out}(x)$ that is close enough to the actual $P_{out}(x)$. Bravyi and Gosset devised a method to approximate the t-qubit magic state $\left|A^{\otimes t}\right\rangle$, where $|A\rangle = 2^{-1/2}(|0\rangle + e^{i\pi/4}|1\rangle$, with a reduced number of stabilizer states.

The *approximate stabilizer rank* χ' is defined as the minimal stabilizer rank (defined in [BSS16] and in Appendix 1.5.1) of a state $|\psi\rangle$ that satisfies $\left|\langle\psi|A^{\otimes t}\rangle\right| \geq 1-\delta$. A close approximation to the tensor product of magic states means a close approximation to the action of a Clifford+T circuit realized by magic state injection [BG16]. Therefore, $\tilde{P}_{out}(x)$ will be close enough to $P_{out}(x)$ if δ is small enough.

The sampling procedure given by Bravyi and Gosset relies on standard computations of stabilizers. We will therefore refer the reader to [BG16] for details of these procedures and focus on the approximate stabilizer rank.

Beginning with the magic state $|A\rangle$ defined above one can construct the equivalent magic state:

$$|H\rangle = e^{-\pi i/8} PH\,|A\rangle = \cos(\pi/8)\,|0\rangle + \sin(\pi/8)\,|1\rangle\,. \tag{1.28}$$

The state $|H\rangle$ can be decomposed into a sum of non-orthogonal stabilizer states as follows:

$$|H\rangle = \frac{1}{2\cos\pi/8}(\left|\tilde{0}\right\rangle + \left|\tilde{1}\right\rangle) \tag{1.29}$$

where $\left|\tilde{0}\right\rangle = |0\rangle$ and $\left|\tilde{1}\right\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Then $\left|H^{\otimes t}\right\rangle$ can be rewritten as

$$\left|H^{\otimes t}\right\rangle = \frac{1}{(2\cos(\pi/8))^t} \sum_{x\in\mathcal{F}_2^t} |\tilde{x}\rangle \tag{1.30}$$

The weak simulation algorithm the reduces the number of stabilizer states

required by approximating $\left|H^{\otimes t}\right\rangle$. This approximation $\left|H^{\otimes t^*}\right\rangle$ is constructed by taking a subspace $\mathcal{L}$ of $\mathcal{F}_2^t$:

$$\left|H^{\otimes t^*}\right\rangle = \frac{1}{(2\cos(\pi/8))^t} \sum_{x\in\mathcal{L}} |\tilde{x}\rangle \tag{1.31}$$

The stabilizer rank of this approximation state is the number of elements in $\mathcal{L}$, which is 2^k. The random subspace $\mathcal{L}$ is chosen so that $\left|H^{\otimes t^*}\right\rangle$ satisfies:

$$\left\langle H^{\otimes t^*}\middle|H^{\otimes t}\right\rangle \leq 1-\delta. \tag{1.32}$$

$\mathcal{L}$ is a k-dimensional binary linear code which can be specified by k generators of length t. These generators can be written in a standard form as a $k \times t$ matrix $\{1_k|G\}$ where 1_k is the $k \times k$ identity matrix and G is a $k \times (t-k)$ matrix. Sampling random subspaces of $\mathcal{F}_2^t$ is therefore equivalent to sampling matrices G.

The algorithm of Bravyi and Gosset achieves a scaling of $\cos(\pi/8)^{-2t} = 2^{0.23t}$ for weak simulation over $2^{0.47t}$ for strong simulation, with an overhead of $1/\delta^2$. In Section 2.5 and 2.6 of Chapter II, we will see more details of how to bound the scaling while we extend this approximate rank and weak simulation scheme to qudits.

1.6 Stabilizer Projector decomposition of density matrices and Robustness of Magic

The stabilizer rank is defined for pure states. In order to describe mixed states, Howard and Campbell used the density matrix formulation to introduce the notion of the stabilizer pseudomixture, where they proposed an l_1 measure called *robustness of magic* (RoM) [HC17], the minimal 1-norm of the decomposition:

$$\mathcal{R}(\rho) = \min\{\sum_i |x_i|, \rho = \sum_i x_i \left|\phi_i\right\rangle \left\langle\phi_i\right|\}, \tag{1.33}$$

where the $|\phi_i\rangle\langle\phi_i|$s are projectors onto stabilizer states, and the coefficients x_i's are real numbers. One can obtain the x_js by solving the convex optimization problem:

$$\min\|x\|_1, \text{s.t.} Ax = b. \tag{1.34}$$

where

$$b_i = \text{Tr}(\rho P_i) \tag{1.35}$$

are proportional to the coefficients of the Pauli decomposition of the density matrix and:

$$A_{ij} = \text{Tr}\{P_i |\phi_j\rangle\langle\phi_j|\}. \tag{1.36}$$

Here the P_i are 4^n Pauli operators that form a complete basis for the density matrix. Using the gadgetized circuits we described in Section 1.5.1, Howard and Campbell proposed a Monte-Carlo algorithm for approximating expectation value $\text{Tr}\big(Z \cdot C\rho C^\dagger\big)$ within δ error, where ρ is the initial resource state, C is the Clifford gadgetized circuit and Z is the computational-basis measurement. The method they use is to randomly draw stabilizer state samples $|\phi_i\rangle\langle\phi_i|$ with probability $\frac{|x_i|}{\sum_i |x_i|}$ in eq. (1.33) and then simulate its evoluation by the Gottesman-Knill theorem. The estimation converges to the true value with more samples calculated. The number of samples needed scales as $\mathcal{R}(\rho)^2\delta^2$ to achieve the required precision. This is the motivation of minimizing the l_1 measure of the stabilizer pseudomixture decomposition.

The stabilizer states form an over-complete basis for density matrices. The representable states for given rank k are given by the span of all k-tuples over the basis. The larger the basis, the more distinct k-tuples there are, and hence a sparse representation is more likely to exist. The *stabilizer pseudomixture (SPM) rank*, previously considered in [BSS16, BK19], is defined as follows:

$$\kappa(\rho) = \min\{k : \rho = \sum_{i=1}^{k} x_i |\phi_i\rangle\langle\phi_i|\}. \tag{1.37}$$

The SPM rank is the minimal l_0 norm of the coefficients of the stabilizer pseudomixture, as compared with the RoM which is the minimal l_1 norm. One can implement an exact strong simulation by running Gottesman-Knill on every single term in the l_0 minimized decomposition.

In Chapter III we will give some numerical examples of stabilizer pseudomixture decompositions for several commonly-discussed resource states and therefore upper-bound their SPM ranks non-trivially. By these low-rank decompositions, we compare the scaling of the resulting exact strong simulation to the scaling of the Howard-Campbell algorithm, which essentially is an l_0 vs. l_1 norm comparison. We then use the idea the stabilizer projector decomposition for unitary operators and propose an improved version of the recursive Feynman path-integral method.

1.7 Objectives and outline of the dissertation

This thesis will focus on exploring simulation based on the Gottesman-Knill theorem, and how they can be viewed from a broader prospective and be combined with other known simulation techniques. The thesis is structured as follows.

In Chapter II, we will discuss how the notion approximate stabilizer rank, and therefore weak Bravyi-Gosset simulation, can be applied to qudits. We also rederived the canonical form of qudit stabilizer states with Gauss sums.

In Chapter III, we will look at how the simulation of circuits with relatively large fraction of non-Clifford gates, e.g. constant fraction, can benefit from stabilizer-based methods. We introduce the stabilizer projector decomposition for density matrices and give several numerical results for upper bounds of stabilizer projector rank of various types of magic states. Extending stabilizer projector decomposition for unitaries, we improved the time scaling of Aaronson and Chen's recursive Feynman path-integral method, while still only requiring polynomial space. We also analyse further advantages and disadvantages of our method compared to the Bravyi-Gosset algorithm and recursive Feynman path-integral algorithms. We construct a parameterized circuit ensemble and identify the parameter regime in this

ensemble where our method offers superior performance. We also estimate the time cost for simulating quantum supremacy experiments with our method and motivate potential improvements of the method.

In Chapter IV, we combine teleportation inspired methods for classical simulation with the algorithm we proposed in Chapter III. Teleportation-inspired simulation corresponds to a different way of contracting the tensor network of the circuit, which we will called turning circuit (gates) sideways. The time scaling to obtain a single amplitude of a quantum circuit with the combined method has exponential factor $O(2^{O(\tilde{n}-\tilde{n}_t/2)})$, where $\tilde{n}$ is the same order as the depth of the original circuit and $\tilde{n}_t$ is the minimal number of sideways T gates (or any sideways diagonal non-Clifford gates) in each non-Clifford layer of the sideways circuit. The method takes advantage of the fact that unitary gates turned sideways may not be unitary, therefore, the stabilizer projector rank will not be full rank, thus improving upon the $O(2^{O(\tilde{n})})$ scaling of the original teleportation-based method. Meanwhile Clifford gates become easy in our method compared to general tensor network methods. We discuss different ways the sideways qubits could flow and how they may change the structure of the sideways circuit.

Finally in Chapter V, we summarize our main results and proposed some future directions. We will also discuss one unfinished work of ours which investigate how l_1 minimization could help finding better stabilizer decompositions using results in the compressed sensing literature.

Chapter 2

Weak simulation of Clifford-dominated circuits and approximate stabilizer rank for qudits

What is supplied by the addition of T-gates to a Clifford circuit? The fault tolerant implementation of Clifford+T circuits substitutes magic states for each T gate [BK05, ZLC00]. Colloquially, T gates add "magic" to a Clifford circuit. Magic is supplied by contextuality, a longstanding source of puzzles and paradoxes in the foundations of quantum mechanics [HWVE14].

The relationship of magic to contextuality also provides a connection to quasiprobability representations of quantum mechanics [Spe08, FE08, FE08]. Specifically, positivity of a quasiprobability representation is equivalent to the absence of contextuality, and such positive states, operations and measurements admit efficient classical simulation in some cases [VWFE13, ME12]. Classical statistical theories with an imposed uncertainty principle can reproduce these positive quasiprobabilistic theories for Gaussian states and qudits with $d > 2$ [Spe16, BRS12].

Pashayan et al gave an algorithm allowing a positive quasiprobability de-

scription to include some negativity [PWB15]. Comparing the algorithms of Bravyi and Gosset and Pashayan should shed more light on the relationship between magic, contextuality and negativity [BG16, PWB15]. However quasiprobability representations for qubits are distinct from their d-dimensional cousins [KL17, KL18, KL19]. The desire to understand the relationship between magic, contextuality and negativity therefore motivates extension of the algorithm of Bravyi and Gosset to qudits for odd prime dimension. In the chapter we obtain the upper bounds of approximate stabilizer rank for odd prime dimension, and therefore extend the weak simulation algorithm of Bravyi and Gosset to qudits. This work is published in [HL19].

2.1 Qudit Pauli group and Clifford gates

In a d dimensional system, where d is odd prime, the Pauli operators X and Z are defined as:

$$X = \sum_{j \in F_d} |j \oplus 1\rangle \langle j| \qquad Z = \sum_{j \in F_d} \omega^j |j\rangle \langle j|, \tag{2.1}$$

where $\omega = \exp(2\pi i/d)$. These operators obey the Heisenberg-Weyl commutation relation:

$$\omega XZ = ZX. \tag{2.2}$$

In d dimensions the Weyl-Heisenberg displacement operators are defined by:

$$D_{\vec{x}} = \tau^{xz} X^x Z^z, \tag{2.3}$$

where $\vec{x} = (x, z)$,$\tau = e^{(d+1)\pi i/d} = \omega^{2^{-1}}$. The qubit Pauli operators are recovered from this expression for $d = 2$, with $D_{(1,0)} = X$, $D_{(0,1)} = Z$ and $D_{(1,1)} = -Y$. The Heisenberg-Weyl operators form a group with multiplication rule:

$$D_{\vec{x}_1} D_{\vec{x}_2} = \tau^{\langle \vec{x}_1 \cdot \vec{x}_2 \rangle} D_{\vec{x}_1 + \vec{x}_2} \tag{2.4}$$

where $\langle \vec{x}_1 \cdot \vec{x}_2 \rangle$ is the symplectic inner product:

$$\langle \vec{x}_1 \cdot \vec{x}_2 \rangle = z_1 x_2 - x_1 z_2 \tag{2.5}$$

For $d > 2$ the Weyl-Heisenberg operators are unitary but not generally Hermitian.

In the qubit case, the Clifford gates map Pauli operators to Pauli operators. In the qudit case Clifford gates map Weyl-Heisenberg operators to one another. The generators of the Clifford group are defined so that the Hadamard gate maps $X \to Z$ and the phase gate maps $X \to XZ$. The generators of the single-qubit Clifford group are:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{2.6}$$

The d-dimensional Clifford operators are generated by:

$$P = \sum_{j \in F_d} \omega^{j(j-1)/2} |j\rangle \langle j| \qquad H = \sum_{j,k} \omega^{jk} |j\rangle \langle k| / \sqrt{d}, \tag{2.7}$$

and:

$$CNOT = \sum_j |j\rangle \langle j| \otimes X^j. \tag{2.8}$$

The single-qudit Clifford group is isomorphic to the semidirect product group of $SL(2, Z_d)$ [1] and $(Z_d)^2$ [App09, Zhu10].

We can represent the Clifford group using a 2×2 matrix F and a 2 vector $\vec{\chi}$, both with entries in Z_d:

$$\mathcal{C} = \left\{ C_{(F|\vec{\chi})} | F \in SL(2, Z_d), \vec{\chi} \in {Z_d}^2 \right\} \tag{2.9}$$

Specifically, a Clifford unitary is given as follows:

$$C_{(F|\vec{\chi})} = D_{\vec{\chi}} U_F, \tag{2.10}$$

[1] $SL(2, Z_d)$ is the group of 2×2 matrices with entries from Z_d and determinant 1.

Where if:

$$F = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}, \qquad \vec{\chi} = \begin{bmatrix} x \\ z \end{bmatrix}, \tag{2.11}$$

then:

$$U_F = \frac{1}{\sqrt{d}} \sum_{j,k=0}^{d-1} \tau^{\beta^{-1}(\alpha k^2 - 2jk + \delta j^2)} \left| j \right\rangle \left\langle k \right|, \tag{2.12}$$

if $\beta \neq 0$ and

$$U_F = \sum_{k=0}^{d-1} \tau^{\alpha\gamma k^2} \left| \alpha k \right\rangle \left\langle k \right|. \tag{2.13}$$

if $\beta = 0$ [Zhu10].

The multiplication rule is:

$$C_{(F_1|\vec{\chi}_1)} C_{(F_2|\vec{\chi}_2)} = \tau^{\langle \vec{\chi}_1 \cdot F \vec{\chi}_2 \rangle} C_{(F_1 F_2 | \vec{\chi}_1 + F_1 \vec{\chi}_2)}. \tag{2.14}$$

The action of the Clifford operators on the Heisenberg-Weyl operators in this representation can be given as follows:

$$C_{(F|\vec{\chi})} D_{\vec{x}} C^{\dagger}_{(F|\vec{\chi})} = \omega^{\vec{\chi} \cdot \vec{x}} D_{F\vec{x}} \tag{2.15}$$

In particular we are interested in Clifford operations defined by matrices of the form:

$$F_\gamma = \begin{bmatrix} 1 & 0 \\ \gamma & 1 \end{bmatrix} \tag{2.16}$$

and we introduce the notation:

$$C_{\gamma,\vec{\chi}} = C_{\begin{bmatrix} 1 & 0 \\ \gamma & 1 \end{bmatrix}, \begin{bmatrix} x \\ z \end{bmatrix}} \tag{2.17}$$

for $\vec{\chi} = (x, z)^T$. From Table I in Zhu [Zhu10] the order of any element $C_{\gamma,\vec{\chi}}$ is d. Clearly X, P and Z are order d. For $d = 2$ H is order 2 and for $d > 2$ H is order 4.

The generators H and P are given by:

$$F_H = \begin{pmatrix} 0 & d-1 \\ 1 & 0 \end{pmatrix}, \quad \vec{\chi}_H = (0,0)^T \tag{2.18}$$

which follows from $HXH^\dagger = Z$ and $HZH^\dagger = X^{-1}$ and:

$$F_P = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad \vec{\chi}_P = (0,(d-1)/2,)^T. \tag{2.19}$$

These expressions for H and P allow us to construct the F and $\vec{\chi}$ for any single qudit Clifford operation expressed as a word on the generators H and P.

The Pauli and Clifford groups were first generalized beyond qubits by Gottesman [Got98a]. Assuming henceforth that d is an odd prime, we define the Heisenberg-Weyl operators:

$$D_{\vec{x}} = \tau^{xz} X^x Z^z, \tag{2.20}$$

where $X|j\rangle = |j \oplus 1\rangle$, where $\oplus$ denotes addition modulo d, $Z|j\rangle = \omega^j |j\rangle$, $\vec{x} = (x,z)$, where x and z are integers modulo d, $\omega = \exp(2\pi i/d)$ and $\tau = e^{(d+1)\pi i/d} = \omega^{2^{-1}}$. The Heisenberg-Weyl operators form a group whose product rule follows from the Heisenberg-Weyl commutation relation $\omega XZ = ZX$:

$$D_{\vec{x}_1} D_{\vec{x}_2} = \tau^{\langle \vec{x}_1 \cdot \vec{x}_2 \rangle} D_{\vec{x}_1 + \vec{x}_2} \tag{2.21}$$

where $\langle \vec{x}_1 \cdot \vec{x}_2 \rangle$ is the symplectic inner product: $\langle \vec{x}_1 \cdot \vec{x}_2 \rangle = z_1 x_2 - x_1 z_2$.

The generators of the Clifford group on qudits are P, H and $CNOT$, where $P|j\rangle = \omega^{j(j-1)/2}|j\rangle$, $H|j\rangle = d^{-1/2}\sum_k \omega^{jk}|k\rangle$ and $CNOT|j,k\rangle = |j, k \oplus j\rangle$. We can also write any single qudit Clifford unitary as $C_{F,\vec{\chi}} = D_{\vec{\chi}} U_F$, where $\vec{\chi} = (x,z)$ and F is a 2×2 matrix with entries modulo d. We will make particular use of matrices $C_{\gamma,\vec{\chi}} = D_{\vec{\chi}} U_\gamma$ for $U_\gamma |k\rangle = \tau^{\gamma k^2} |k\rangle$. The order of $C_{\gamma,\vec{\chi}}$ is d.

Qudit stabilizer states can be prepared from a logical basis state by a qudit Clifford circuit. The Gottesmann-Knill theorem generalizes to qudits and qudit

stabilizer computations allow efficient classical simulation. Qudit stabilizer states possess canonical forms in the logical basis just as in the qubit case [Nes08, DDM03, HDDM05].

The remaining generalization we require is an efficient classical algorithm for obtaining the inner product of two stabilizer states. This is required by the algorithm of Bravyi and Gosset and the qubit case was given in [BG16]. We give an $O(n^3)$ algorithm for the inner product of two n-qudit stabilizer states based on Gauss sums in Section 2.4.1.

2.2 Qudit Magic states and T gates

The Clifford hierarchy, as we defined in eq. (1.5), classifies unitary operators by their action on the Pauli group. The first level of the Clifford hierarchy is the Pauli group $\mathcal{C}(1) = \mathcal{P}$. The Clifford group is the second level of the hierarchy, unitary operators that map the Pauli group to itself. The third level of the Clifford hierarchy are operators that map Pauli operators to Clifford operators. The qubit T gate is such an operator because $TXT^\dagger = PHP^2H$, a non-Pauli element of the second level of the Clifford hierarchy.

Bravyi and Kitaev first proposed qubit magic states in [BK05]. They define magic states as the image of $|H\rangle$ and $|T\rangle$ under single-qubit Clifford gates, where $|H\rangle$ is defined by eq. (1.28) and $|T\rangle$ by

$$|T\rangle = \cos\beta\,|0\rangle + \sin\beta e^{i\pi/4}\,|1\rangle\,, \tag{2.22}$$

for $\cos(2\beta) = \frac{1}{\sqrt{3}}$. $|H\rangle$ is the eigenstate of the Hadamard gate H and $|T\rangle$ is the eigenstate of the product of Hadamard and Phase gate PH.

Any magic state is equivalent as a resource to any other state obtainable from it by a Clifford operation. We can define magic states more generally as the eigenstates of Clifford operations and obtain them as follows. Taking any H-type

magic state $U|H\rangle$, we have

$$UHU^\dagger U|H\rangle = UH|H\rangle = \lambda U|H\rangle \tag{2.23}$$

where λ is the eigenvalue of H and U is a Clifford gate. This means that $U|H\rangle$ is the eigenstate of a new Clifford operator $UHU^\dagger$. The same is true for T-type magic states.

Campbell *et al.* [CAB12] used this relationship between magic states and eigenvectors of Clifford operators to extend the definition of magic state to qudits [CAB12]. Concurrently, equivalent extensions were obtained by Howard and Vala [HV12].

2.2.1 Qudit T gates

Campbell *et al.* [CAB12] define sets of gates $\mathcal{M}_d^m$ containing all gates M with the following properties:

1. M is diagonal
2. $M^{d^m} = 1$
3. $\det M = 1$ so that $M \in SU(d)$
4. M is in the third but not the second level of the Clifford hierarchy.

Amongst this set of gates is the canonical $\mathcal{M}_d$ gate

$$M_d = \sum_j \exp(i2\lambda_j \pi/d^m)|j\rangle\langle j| \tag{2.24}$$

Which is defined so that it maps the X operator to a Clifford operator proportional to XP:

$$C_d = M_d X M_d^\dagger = \begin{cases} e^{2\pi i/9} XP. & d = 3. \\ \omega^{-\bar{3}} XP. & d > 3. \end{cases} \tag{2.25}$$

where $\bar{3}$ indicates the multiplicative inverse of 3 modulo d. This Clifford operator has order d.

This condition, and the condition $\det M = 1$, gives the following form for the λ_j (See Appendix A of [CAB12]):

$$\lambda_j = d^{m-2}\left[d\binom{j}{3} - j\binom{d}{3} + \binom{d+1}{4}\right]. \tag{2.26}$$

The parameter m determines the order d^m of the operator M. For $d = 3$ the form above is valid when $m \geq 2$. For $d > 3$ it is valid when $m \geq 1$.

By definition M maps X, a generalized Pauli operator, to a non-Pauli Clifford operator and so is in the third, but not the second, level of the Clifford hierarchy. We can therefore think of M as a generalized T gate.

From the definition of the matrix M in (2.24), we have for $d = 3$ and $m = 2$:

$$M_3 = \mathrm{diag}\left(e^{i2\pi/9}, \quad 1, \quad e^{-i2\pi/9}\right), \tag{2.27}$$

and

$$M_5 = \mathrm{diag}\left(e^{-\frac{4\pi i}{5}}, \quad e^{\frac{2\pi i}{5}}, \quad e^{-\frac{2\pi i}{5}}, \quad e^{-\frac{4\pi i}{5}}, \quad e^{-\frac{2\pi i}{5}}\right), \tag{2.28}$$

for $d = 5$ and $m = 1$ where $\omega = e^{2\pi i/5}$. The qudit version of the T gate M, is further generalized in [HV12], which we will discuss below.

The T gate is also sometimes called the $\pi/8$ gate because

$$T = e^{-i\pi/8}\begin{pmatrix} e^{i\pi/8} & 0 \\ 0 & e^{-i\pi/8} \end{pmatrix}. \tag{2.29}$$

Vala and Howard developed the qudit versions of this gate concurrently with Campbell et al's development of qudit magic states [CAB12, HV12]. The results are equivalent and we give the details of the relationship between them here.

Vala and Howard parameterize the set of diagonal gates on a single qudit as follows:

$$U_v = U(v_0, v_1, ..., v_{p-1}) = \sum_{j=0}^{d-1} \omega^{v_k} |k\rangle \langle k| \, (v_k \in Z_d). \tag{2.30}$$

All diagonal gates fix $D_{(0,1)}$ and so their action is completely determined by $U_v D_{(1,0)} U_v^\dagger =$

$U_v X U_v^\dagger$. This parallels the development of Campbell et al. who considered the action of their canonical gate M on the operator X and insisted that the result of that action was $\propto XP$.

Vala and Howard proceed more generally, computing the action of these diagonal matrices:

$$U_v D_{(x|z)} U_v^\dagger = D_{(x|z)} \sum_k \omega^{(v_{k+1}-v_k)} |k\rangle \langle k| . \tag{2.31}$$

Given U_v is diagonal, only $U_v D_{(1|0)} {U_v}^\dagger$ is nontrivial.

Vala and Howard then consider the case that U_v is in the third level of the Clifford hierarchy so that the image of X can be written (c.f. eq (18) in [HV12]):

$$U_v X {U_v}^\dagger = \omega^{\epsilon'} C_{\gamma',(1,z')^T}. \tag{2.32}$$

where $\epsilon', \gamma', z' \in Z_d$. The right hand side here is the most general form allowed because eqn. (2.31) implies that the image of X must be X times a diagonal Clifford operator, and the most general form of a diagonal Clifford operator has $\vec{\chi} = (0, 1)$ and $\beta = 0$, $\alpha = 1$. Combining equation (2.31) and (2.32), one obtains (c.f. eq. (19) in [HV12]):

$$X \sum_k \omega^{(v_{k+x}-v_k)} |k\rangle \langle k| = \omega^{\epsilon'} C_{\gamma,(1,z')^T}. \tag{2.33}$$

Vala and Howard then solve for U_v with these 3 parameters.

$$v_k = \bar{12}k\{\gamma' + k[6z' + (2k-3)\gamma']\} + k\epsilon', \tag{2.34}$$

This analysis is equivalent to that performed in Campbell et al. [CAB12], Appendix A.

The $d = 3$ case as usual presents some special difficulties. In the Campbell analysis one must choose $m = 2$ for λ as there are no Clifford operators with $m = 1$, $d = 3$ [CAB12].

The set of operators U_v for $d = 3$ is given by:

$$U_v = \sum_{k=0}^{2} \xi^{v_k} |k\rangle \langle k| . \tag{2.35}$$

where $\xi = e^{2\pi i/9}$. The v_k are given by:

$$v = (v_0, v_1, v_2) = (0, 6z' + 2\gamma' + 3\epsilon', 6z' + \gamma' + 6\epsilon'), \tag{2.36}$$

where all operations can be taken modulo 9. The determinant of U_v for $d = 3$ can be computed from this definition:

$$\det U_v = e^{\frac{2\pi i}{9} \sum_k = 0^2 v_k} = e^{\frac{2\pi i}{3}(z' + \gamma')}$$

showing that U_v is not in $SU(3)$ for $d = 3$.

We can relate the diagonal operators U_v defined by Vala and Howard and the operators M defined by Campbell et. al as follows. Writing:

$$M = \sum_{k=0}^{d-1} \exp\left(\frac{2\pi i}{d^m} \lambda_k\right) |k\rangle \langle k| = \sum_{k=0}^{d-1} \omega^{\lambda_k / d^{m-1}} |k\rangle \langle k| \tag{2.37}$$

and:

$$U_v = \sum_{k=0}^{d-1} \omega^{v_k} |k\rangle \langle k| \tag{2.38}$$

we wish to compare:

$$\frac{\lambda_k}{d^{m-1}} = \frac{1}{d} \left[d \binom{k}{3} - k \binom{d}{3} + \binom{d+1}{4} \right] \tag{2.39}$$

and

$$v_k = \bar{12} k \{\gamma' + k[6z' + (2k-3)\gamma']\} + k\epsilon'. \tag{2.40}$$

These are both cubic in k and so we can find the particular U_v that corresponds to M by equating the coefficients. We begin by setting $k = 0$ to find the

constant term. We immediately obtain:

$$v_0 = 0, \qquad \frac{\lambda_0}{d} = \frac{1}{d}\binom{d+1}{4} \tag{2.41}$$

We conclude that U_v and M will only be equivalent up to a global phase determined by this convention.

Equating the cubic terms yields $\gamma' = 1$. Equating the quadratic terms gives

$$z' - \frac{\gamma'}{2} = d - 1 \tag{2.42}$$

so that $z' = (d-1)/2$. Finally, equating the linear terms gives:

$$\epsilon' = \bar{12}(6d - 2d^2 - 1). \tag{2.43}$$

We may therefore relate $U_v(z', \gamma', \epsilon')$ and M for arbitrary $d > 3$ as follows:

$$M_d = \omega^{\frac{1}{d}\binom{d+1}{4}} U_v((d-1)/2, 1, \bar{12}(6d - 2d^2 - 1)) \tag{2.44}$$

The first two cases of this equivalence are for $d = 5$ amd $d = 7$ and, up to a global phase, are as given in equations (70) and (71) of [HV12].

The case of $d = 3$ is distinct ($\bar{12}$ does not exist modulo 3.) but from the definition of U_v for $d = 3$ given in eqn. 2.35 and eqn. 2.36 we have:

$$M_3 = e^{\frac{2\pi i}{9}} U_v(1, 1, 0) \tag{2.45}$$

This is, up to a global phase, as given in eqn. (69) of [HV12].

2.2.2 Qudit Magic states

The gates M also allow us to find eigenstates of C_M as follows. Define the state $|M_k\rangle = M|+_k\rangle$, where $|+_k\rangle$ is the eigenstate of X with eigenvalue ω^k. We can

calculate:

$$\begin{aligned} C_M |M_k\rangle &\propto MXM^\dagger |M_k\rangle \\ &= MXM^\dagger M |+_k\rangle \\ &= \omega^k M |+_k\rangle \\ &= \omega^k |M_k\rangle \end{aligned} \tag{2.46}$$

Given eq.(2.32), Vala and Howard recovered the definition of the magic states of Campbell and showed that these magic states $U_v |+\rangle$ are eigenstates of $C_{\gamma',(1,z')^T}$ with eigenvalue $\omega^{-\epsilon'}$:

$$\begin{aligned} C_{\gamma',(1,z')^T} U_v |+\rangle &= \omega^{-\epsilon'} U_v D_{(1|0)} U_v{}^\dagger U_v |+\rangle \\ &= \omega^{-\epsilon'} U_v D_{(1|0)} |+\rangle = \omega^{-\epsilon'} U_v |+\rangle \end{aligned} \tag{2.47}$$

The qudit T-gate was defined in [HV12, CAB12] as a diagonal gate U_T that maps Pauli operators to Clifford operators. Its action is specified by the image of $X = D_{(1,0)}$ under U_T. Magic states are then eigenvectors of this image. Let the eigenstate of X with eigenvalue ω^k be $|+_k\rangle$, then the magic states are $U_T |+_k\rangle$. This approach is that taken by Howard in [HV12].

The image of X under U_T can be written (up to a phase) as $C = XP^\gamma Z^\xi$ for γ, ξ integers modulo d. The effect of nonzero ξ is simply to reorder the eigenvectors and hence we can choose $\xi = 0$. Similarly, the eigenvectors for $\gamma > 1$ and $\gamma = 1$ are related by application of $P^{\gamma-1}$, a Clifford operator. We can therefore specialize to the case $\gamma = 1$ and $\xi = 0$, and the gate with action:

$$C_d = M_d X M_d^\dagger = \begin{cases} e^{2\pi i/9} XP. & d = 3. \\ \omega^{-\bar{3}} XP. & d > 3. \end{cases} \tag{2.48}$$

where $\bar{3}$ indicates the multiplicative inverse of 3 modulo d. This is the gate defined by Campbell et al. [CAB12].

The definition of magic states allows one to replace a Clifford+T circuit with

a Clifford circuit with injected magic states [ZLC00, BK05]. This construction was extended to qudits in [HV12] and we will discuss it in the next section.

2.3 Qudit T gate Gadget

We also require a gadget that substitutes a qudit T-gate by an injected qudit magic state and Clifford gates. The qudit gadget was introduced by Howard and Vala and is shown in Figure 2.1.

Howard and Vala also generalized the qubit T-gate gadget to qudits for their magic state construction [HV12]. We reproduce their gadget here in the interest of making the paper self contained.

In order to project a qudit state onto the eigenstate of operator P with eigenvalue ω^k, the projection operator can be written as:

$$\Pi_{(P|k)} = \frac{1}{d}(I + \omega^{-k}P + \omega^{-2k}P^2 + ... + \omega^{-(d-1)k}P^{d-1}) \tag{2.49}$$

By analogy with the qubit case, we need a gadget that allows us to implement qudit U_v gate by injecting magic states. It's straight forward to check that the following does the job:

$$\text{CSUM}^{-1} \cdot \Pi_{(0,0|1,d-1)[0]}(|\psi\rangle |\psi_{U_v}\rangle) = U_v |\psi\rangle |0\rangle \tag{2.50}$$

for a given arbitrary state $|\psi\rangle$, where $|\psi_{U_v}\rangle = U_v |+\rangle$ is the magic state and Π is a rank-p projector defined by

$$\Pi_{(0,0|1,d-1)[0]} = \frac{1}{d}[I + Z \otimes Z^{-1} + ... + (Z \otimes Z^{-1})^{d-1}] \tag{2.51}$$

This projection is equivalent to measuring the $Z \otimes Z^{-1}$ observable to get eigenvalue 1. If we get eigenvalue ω^k, we perform a X^{-k} on the first qudit state to recover it back to the 1-eigenspace. In fact, this gadget works for implementing any diagonal gate U by injecting the state $U |+\rangle$.

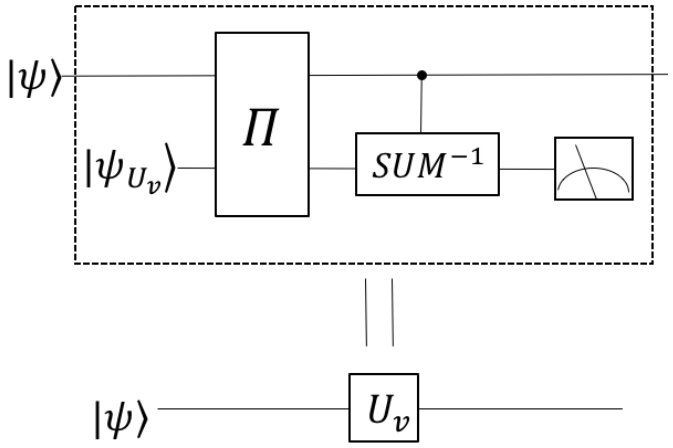

Figure 2.1: Gadget for qudit U_v gate.

2.4 Canonical forms for Qudit stabilizer states and inner product algorithm

To prove the canonical form in eq. (1.4) holds true for all qubit stabilizer states, one only need to make sure that every state in this form is the eigenstate of a stabilizer operator, as shown in [DDM03]. It also suffices to verify that any of the $\{H, P, CNOT\}$ gates preserves the form, only changing the coefficients of $q(v)$, $l(v)$ and affine space $Ru + t$. This proof is given in [Nes08].

The normal form was generalized to arbitrary dimensions in [HDDM05]. The stabilizer canonical form for qudits is:

$$|\psi\rangle \propto \sum_{u \in Z_d^k} \omega^{q_d(u)+q_n(u)} |Gu + h\rangle \,. \tag{2.52}$$

where $q_n(u) = \sum_{i \neq j} q_{ij} u_i u_j$ is a quadratic form with only non-diagonal terms, and the coefficients are modulo d. The term $q_d(u) = \sum_{i=1}^{k} q_i \frac{u_i(u_i-1)}{2} + l_i u_i, q_i, l_i \in Z_d$ only has diagonal terms and is coupled to the linear term. The state has support in an k-dimensional affine space

$$\vec{x} = Gu + h = \text{span}(g^1, ...g^k) \oplus h = u_1 g^1 \oplus u_2 g^2 ... \oplus u_k g^k \oplus h. \tag{2.53}$$

G is an $n \times k$ matrix and has each of its columns being $g_1, ..., g_k$ with binary entries, while h is a $n \times 1$ vector that has entries in Z_d. The phase is divided into two quadratic terms that reflect the action of the phase and Hadamard gates, respectively. States of this form were shown to be the $+1$ eigenstate of some Pauli (Weyl-Heisenberg) operator in [HDDM05].

This quadratic form on the exponent can also be represented in matrix form:

$$q_d(u) + q_n(u) = 2^{-1} u^T Q u + L u. \tag{2.54}$$

where 2^{-1} is taken modulo d. Here Q is a $k \times k$ matrix with its diagonal terms being q_i and off-diagonal terms being q_{ij} and L is $1 \times n$ and each term corresponds to $l_i - q_i$.

We will give a new proof of this below by showing that this form is preserved under Clifford operations, using the properties of quadratic Gauss sums. We give this proof in order to develop the techniques we will use in the inner product algorithm for qudit stabilizer states.

We consider the single qudit case first. We will prove that the form:

$$\frac{1}{\sqrt{d}} \sum_{j \in Z_d^m} \omega^{f\frac{(j-1)j}{2} + gj} \left| j + y \right\rangle, \tag{2.55}$$

is preserved under the action of the single-qudit Clifford generators where f and g belong to Z_d, y is a shift vector that also belongs to Z_d. We are studying single-qudit case here so m is either 0 or 1. When $m = 0$, this is simply a computational basis state.

Acting with diagonal Clifford gates on (2.55) such as P or Z will only change the coefficients f and g in this expression. Similarly, acting with powers of the X gate will only shift y, again preserving the quadratic form of the exponents.

It only remains to check the Hadamard gate:

$$\begin{aligned} & H\frac{1}{\sqrt{d}}\sum_{j\in Z_d^m}\omega^{\frac{fj(j-1)}{2}+gj}\,|j+y\rangle \\ & =\frac{1}{d}\sum_k \omega^{yk}\left(\sum_{j\in Z_d^m}\omega^{\frac{fj(j-1)}{2}+(k+g)j}\right)|k\rangle \end{aligned} \tag{2.56}$$

If $m=0$, the quantity in the parentheses is simply a phase factor without the sum. Then this form reverts to (2.55) with $f=0 \mod d$. If $m=1$, we recognize the quantity in the parentheses as a Gauss sum. There are again two cases. If $f=0 \mod d$, then we have

$$\sum_{j\in Z_d}\omega^{(k+g)j}=d\delta_{k+g,0} \tag{2.57}$$

Then (2.56) reverts to (2.55) as $m=0$ case, i.e., a computational basis state.

If $f\neq 0 \mod d$, to compute this Gauss sum, we first complete the square:

$$\begin{aligned} \sum_j \omega^{\frac{fj(j-1)}{2}+(k+g)j} &= \sum_j \omega^{\frac{f}{2}(j^2-j+2(k+g)j\bar{f})} \\ &= \omega^{-\bar{2}f(\bar{f}(k+g)-\bar{2})^2}\sum_j \omega^{\bar{2}f(j-\bar{2}+(k+g)\bar{f})^2} \\ &= \omega^{-\bar{2}f(\bar{f}(k+g)-\bar{2})^2}\sum_n e^{2\pi i\bar{2}fn^2/d} \end{aligned} \tag{2.58}$$

where $\bar{2}$, $\bar{f}$ meaning that $2\bar{2}\equiv 1 \mod d$ and $f\bar{f}\equiv 1 \mod d$.

The value of this Gauss sum is well known:

$$\sum_n e^{2\pi i\bar{2}fn^2/d}=\begin{cases}(\frac{\bar{2}f}{d})\sqrt{d}, & d\equiv 1(\mod 4)\\ i(\frac{\bar{2}f}{d})\sqrt{d}, & d\equiv 3(\mod 4),\end{cases} \tag{2.59}$$

where $(\frac{\bar{2}f}{d})$ is the Legendre symbol.

Hence:

$$\sum_j \omega^{\frac{fj(j-1)}{2}+(k+g)j}\propto\omega^{\frac{-\bar{f}k(k-1)}{2}-\bar{2}((2g+1)\bar{f}-1)k} \tag{2.60}$$

The new coefficients $-\bar{f}$ and $-\bar{2}((2g+1)\bar{f}-1)$ here are still in Z_d. This means that the general form $\sum_k \omega^{f\frac{(k-1)k}{2}+gk}\,|k\rangle$ of single-qudit stabilizer states is preserved

under the action of any Clifford operations.

For multi-qudit states, we have the same affine space property as the qubit case except that the additions are modulo d. Before we give the proof, we need to show that quadratic form given in terms of the basis vectors of the affine space $\vec{u}$ and the qudit vectors itself $\vec{x}$ are equivalent. Changing the arguments only changes the coefficients of the quadratic form. Given eq.(2.54), we further assume the quadratic and linear matrices in terms of x being $\tilde{Q}$ and $\tilde{L}$:

$$\begin{aligned} \omega^{x^T\tilde{Q}x+\tilde{L}x} &= \omega^{(u^TG^T+h^T)\tilde{Q}(Gu+h)+\tilde{L}(Gu+h)} \\ &\propto \omega^{u^TG^T\tilde{Q}Gu+(2h^T\tilde{Q}G+\tilde{L}G)u} \end{aligned} \tag{2.61}$$

From this equation, we can see the relationship between Q: L and $\tilde{Q}$, $\tilde{L}$: $Q = G^T\tilde{Q}G$, and $L = 2h^T\tilde{Q}G + \tilde{L}G$.

Now we are going to reproduce Van den Nest's method [Nes08] here to prove that the canonical form (2.52) preserves under the action of $CSUM$, the Phase and the Hadamard gates. The $CSUM_{i\to j}$ gate shifts the affine space by mapping $|a\rangle|b\rangle$ to $|a\rangle|a\oplus b\rangle$, without changing the phases. As in the qubit case, we only need to add the ith column of the matrix G to the jth column:

$$\begin{aligned} CSUM &\sum_{u\in Z_d^m} \omega^{q_d(u)+q_n(u)} |Gu+h\rangle \\ &= \sum_{u\in Z_d^m} \omega^{q_d(u)+q_n(u)} \left|G'u+h\right\rangle . \end{aligned} \tag{2.62}$$

G' differs from G by $g_j \to g_i \oplus g_j$.

Action with a phase gate on qudit i results in the state:

$$P_i|\psi\rangle \propto \sum_{x\in A} \omega^{q_n(x)+q_d(x)}\omega^{\frac{x_i(x_i-1)}{2}}|x\rangle \tag{2.63}$$

Which again leaves the normal form unchanged.

The Hadamard gate requires some work. Without loss of generality, we

assume that the Hadamard gate acts on the first qudit:

$$H_1 \left|\psi\right\rangle \propto \sum_{v=0}^{d-1} \sum_{u} \omega^{q_n(u)+q_d(u)+v(\bar{g_1}u+h_1)} \left|v, \bar{G}u+\bar{h}\right\rangle \tag{2.64}$$

where $\tilde{g_1}^T$ is the first row of G and $\bar{G}$ is the rest of it. If $\bar{G}$ is still full rank after taking out $\tilde{g_1}^T$, we obtain the new G' to be:

$$\begin{pmatrix} 1 & \vec{0}^T \\ \vec{0} & \bar{G} \end{pmatrix} \tag{2.65}$$

Therefore we have $m+1$ basis vectors now, and v becomes the new u_1. The term $v(\bar{g_1}u+t_1)$ in the phase can be absorbed in the quadratic form $q_n(u)$. So this is of the canonical form (2.52).

If $\bar{G}$ is rank $m-1$ after taking out $\tilde{g_1}^T$, then we have each column of $\bar{G}$ linearly dependent of each other. In this case one of the u_is is redundant and we want it to be summed out on order to get back to the canonical form. Without loss of generality, let's assume that $u_1 = \sum_{i=2}^{m} r_i \bar{g_i}$, therefore $\bar{G}u+\bar{h} = \sum_{i=2}^{m}(u_i+r_i)\bar{g_i}+\bar{h}$. If we denote $u'_i \equiv u_i + r_i$ for $i=2$ to m ($\bar{u}$) and $u'_1 \equiv v$, $q_n(\bar{u})$ and $q_d(\bar{u})$ can be written in terms of $\bar{u'}$ with different coefficients from q_n and q_d, say $q'_n(\bar{u'})$ and $q'_d(\bar{u'})$, together with some constant factor which can be neglected. Then eq.(2.64) becomes:

$$\begin{aligned}
H_1 \left|\psi\right\rangle &\propto \sum_{v=0}^{d-1} \sum_{u} \omega^{q_n(u)+q_d(u)+v(\tilde{g_1}^T u+h_1)} \left|v, \bar{G}u+\bar{h}\right\rangle \\
&\propto \sum_{v=0}^{d-1} \sum_{u'_2,\ldots u'_m} \sum_{u_1} \omega^{q_n(u)+q_d(u)+v(\tilde{g_1}^T u+h_1)} \left| v, \sum_{i=2}^{m} u'_i \bar{g_i} + \bar{h} \right\rangle \\
&= \sum_{u'_1,u'_2,\ldots u'_m} \omega^{q'_n(\bar{u'})+q'_d(\bar{u'})+u'_1(\sum_{i=2}^{m} g_{1i}(u'_i-r_i)+\bar{h_1})} \\
&\left(\sum_{u_1} \omega^{q_n(u_1)+q_d(u_1)+v g_{11} u_1} \right) \left| v, \sum_{i=2}^{m} u'_i \bar{g_i} + \bar{h} \right\rangle
\end{aligned} \tag{2.66}$$

Here what's in the parenthesis is the Gauss sum we computed earlier. Then we can drop the prime for the us and absorbed the result of the Gauss sum and

$u'_1(\sum_{i=2}^m g_{1i}(u'_i - r_i) + \bar{h_1})$ into the q'_n and q'_d functions. Finally we arrive at the same form but with different coefficients. Hence, the canonical form is preserved for Hadamard gate and all Clifford gates.

We now use this canonical form and the Gauss sum techniques to provide an $O(n^3)$ algorithm for the computation of the inner products of two stabilizer states.

2.4.1 The inner product of two qudit stabilizer states

The inner product between two qubit stabilizer states can be computed efficiently in $O(n^3)$ ([AG04, GMC12, BSS16, BG16]). However, a corresponding algorithm for qudits has not been obtained yet, although most aspects of the theory of stabilizer states have been generalized [Got98a, HDDM05]. We will now describe a $O(n^3)$ algorithm that computes the inner product of two qudit stabilizer states based on the Gauss sum techniques we discussed in the previous section.

As discussed above, the quadratic form in terms of the basis vector of the affine space $\vec{u}$ and the qudit vector itself $\vec{x}$ are equivalent. Therefore eq. (2.52) is equivalent to the following:

$$|\psi\rangle \propto \sum_{x\in A} \omega^{\tilde{q_n}(x)+\tilde{q_d}(x)} |x\rangle \tag{2.67}$$

where A is the affine space defined by $Gu + h$ in eq.(2.53).

Assume we have two qudit stabilizer states $|\psi_1\rangle$ and $|\psi_2\rangle$, which take the above form (2.67) with subindices 1 and 2:

$$\begin{aligned}
\langle\psi_2|\psi_1\rangle &= d^{-(k_1+k_2)/2} \sum_{x_1\in A_1} \sum_{x_2\in A_2} \omega^{\tilde{q_1}(x_1)-\tilde{q_2}(x_2)} \langle x_2|x_1\rangle \\
&= d^{-(k_1+k_2)/2} \sum_{x\in A_1\cap A_2} \omega^{\tilde{q_1}(x)-\tilde{q_2}(x)} \\
&= d^{-(k_1+k_2)/2} \sum_{x\in A_1\cap A_2} \omega^{\tilde{q}(x)} \\
&= d^{-(k_1+k_2)/2} \sum_{u\in F_d^k} \omega^{q(u)}
\end{aligned} \tag{2.68}$$

where $\tilde{q_1} = \tilde{q_{1d}} + \tilde{q_{1n}}$, $\tilde{q_2} = \tilde{q_{2d}} + \tilde{q_{2n}}$, $\tilde{q} = \tilde{q_1} - \tilde{q_2}$, k is the dimension of $A_1 \cap A_2$ and q is the quadratic form in the new basis of $A_1 \cap A_2$.

As discussed by Bravyi and Gosset in Appendix B, C for qubits [BG16], The new basis of affine space $A_1 \cap A_2$, as well as the new quadratic form associated with it, can be calculated with the same method for qudit in $O(n^3)$.

What remains in eq.(2.68) is a Gauss sum, which we again rewrite in the following form:

$$\sum_{u \in F_d^k} \omega^{u^T Q u + L u} \tag{2.69}$$

according to eq. (2.54). We can diagonalize Q and factor this sum into a product of k Gauss sums over F_d. Here we use elementary-operation method to obtain a transformation matrix P that gives

$$P^T Q P = \Lambda. \tag{2.70}$$

where Λ is the diagonal matrix with entries $(\lambda_1, ..., \lambda_k)$.

Then if we further define $u = Pu'$, we obtain

$$\begin{aligned} \sum_{u \in F_d^k} \omega^{u^T Q u + L u} &= \sum_{u' \in F_d^k} \omega^{u'^T P^T Q P u' + L P u'} \\ &= \sum_{u' \in F_d^k} \omega^{u'^T \Lambda u' + L P u'} \\ &= \prod_{i=1}^{k} \sum_{u_i \in F_d} \omega^{\lambda_i u_i'^2 + l_i' u_i'}, \end{aligned} \tag{2.71}$$

where $l_i' = \sum_j p_{ji} l_j$. This is a product of k Gauss sums, as given in eq.(2.58,2.59,2.60).

Each Gauss sum only takes $O(1)$ time, so the product of k of them takes time $O(k)$. The scaling of this algorithm is determined by the complexity of Gaussian elimination, $O(k^3)$ because Q has rank k. Therefore, together with the first step to obtain $A_1 \cap A_2$, the algorithm takes $O(n^3)$ time overall in the worst case.

2.5 Nonorthogonal decompositions of qudit Magic states

The qudit magic state we want to decompose is an eigenvalue one eigenstate of the Clifford operator C_d as defined by eq. (2.48). We choose a stabilizer state $\left|\tilde{0}\right\rangle$ with non-zero inner product with the magic state and act on it with powers of C_d to obtain d stabilizer states $\{\left|\tilde{j}\right\rangle = C_d^j \left|\tilde{0}\right\rangle, j = 0, ..., d-1\}$. We know these stabilizer states are distinct because if any pair were equal then the original state $\left|\tilde{0}\right\rangle$ would be an eigenstate of the Clifford operator and hence a magic state. The sum of these d states form a decomposition of the magic state (up to a possible global phase). Because C_d has order d this state is by construction an eigenvalue one eigenstate of C_d.

The d stabilizer states in the decomposition form an *orbit* around the magic state. This is also noticed and discussed by Mark Howard in [How15], which he calls MUB(mutual unbiased basis)-cyclers. There are $d(d+1)$ single-qudit stabilizer states [Woo87], partitioned into $d+1$ orbits, each orbit giving a decomposition of the magic state. Every state in each orbit has the same overlap with the magic state:

$$\left\langle\tilde{j}\middle|M_d\right\rangle = \left\langle\tilde{j}\right| C_d^\dagger C_d \left|M_d\right\rangle = \left\langle\widetilde{j+1}\middle|M_d\right\rangle \tag{2.72}$$

where the qudit magic state is $|M_d\rangle = M_d|+\rangle$. This property is a generalization of $\left\langle\tilde{0}\middle|H\right\rangle = \left\langle\tilde{1}\middle|H\right\rangle = \cos\pi/8$ for the qubit case. The overlaps of the elements of the nonorthogonal basis are given by: $\left|\left\langle\tilde{0}\middle|\tilde{j}\right\rangle\right| = \frac{1}{\sqrt{d}}$ for all js, i.e.:

$$\left|\left\langle\tilde{j}\middle|\tilde{k}\right\rangle\right|^2 = \frac{1+(d-1)\delta_{j,k}}{d}. \tag{2.73}$$

This expression is that for states in a SIC-POVM, and the construction here is similar to the generation of such states from a fiducial state [FHS17, Zhu10]. Here we only obtain d states, however. See Section 2.7.2 for the evaluation of the phase of $\left\langle\tilde{j}\middle|\tilde{k}\right\rangle$.

The states $|+_p\rangle = Z^p|+\rangle$ are representatives of the d orbits, each of which generated by C_d. This is because $C_d^a|+_p\rangle \neq |+_q\rangle$ for any a, p, q, which follows simply

from the action of C_d in the logical basis. C_d applies phases quadratic in j to $|j\rangle$ followed by a shift. This cannot be equal to a state generated from $|+\rangle$ by any power of Z, which can only apply phases linear in j to $|j\rangle$. This is essentially a re-partition of the $d(d+1)$ single-qudit stabilizer states than the *Mutual Unbiased Basis* partition [Woo87]. In the MUB partition, all d states within each set are orthogonal to each other and are aligned the same direction in the discrete Wigner function representation, all $d+1$ different sets have different alignments. Here we have $d+1$ non-orthogonal states in each set, which forms an orbit around the magic state and have the same overlap with the magic state. There are d orbits overall.

From the orbit representatives we can determine the inner product of the states in the orbit with the magic state. This is given by:

$$\alpha = \langle +| Z^{-p} |M_d\rangle = \langle +| Z^{-p} M_d |+\rangle = \frac{1}{d}\mathrm{Tr}(Z^{-p} M_d). \tag{2.74}$$

This is a cubic Gauss sum which can be written:

$$\alpha = \frac{\omega^{\frac{1}{d}\binom{d}{4}-p}}{d} \sum_{l=0}^{d-1} \omega^{\bar{6}l(l^2+\psi(p,d))} \quad d > 3. \tag{2.75}$$

The $d = 3$ case, the magnitude and phase of this cubic Gauss sum, and $\phi(p, d)$, are computed in Section 2.7.1. The sum is real, although not necessarily positive. Although we do not obtain a closed form for this sum, we can compute the integer value of p which maximizes its absolute value for a given d. These values are tabulated for small d in Table 2.1.

The complete form of the nonorthogonal decomposition is:

$$|M_d\rangle = \pm \frac{\omega^{\frac{1}{d}\binom{d}{4}-p}}{d|\alpha|} \sum_j C_d^j |\tilde{0}\rangle . \tag{2.76}$$

which is the generalization of eq. (1.29) to arbitrary d.

2.6 Weak Simulation and Approximate Stabilizer Rank

In order to get an approximation for $\left|M^{\otimes t}\right\rangle$, we can follow the method of Bravyi and Gosset for the qubit case, taking a k-dimensional subspace of $\mathcal{F}_d^t$:

$$\left|M^{\otimes t*}\right\rangle = |\mathcal{L}\rangle = \frac{1}{\sqrt{d^k Z(\mathcal{L})}} \sum_{x\in\mathcal{L}} \left|\tilde{\vec{x}}\right\rangle \tag{2.77}$$

Here we label the state by $\mathcal{L} \subset \mathcal{F}_d^t$, a k dimensional code subspace of $\mathcal{F}_d^t$ and $Z(\mathcal{L})$ is a normalization factor. Comparison with eq. (2.76) shows that $Z(F_d) = d|\alpha|^2$. We require:

$$\left|\left\langle\mathcal{L}\middle|M^{\otimes t}\right\rangle\right|^2 = \frac{d^k|\alpha|^{2t}}{Z(\mathcal{L})} \geq 1-\delta \tag{2.78}$$

for a given δ, where the first equality follows from eq. (2.72) and where:

$$Z(\mathcal{L}) = \sum_{\vec{x}\in\mathcal{L}} \left\langle\tilde{0}^{\otimes t}\right| C_{\vec{x}} \left|\tilde{0}^{\otimes t}\right\rangle \tag{2.79}$$

for $C_{\vec{x}} = C^{x_1} \otimes C^{x_2} ... \otimes C^{x_t}$.

Selection of the subspace $\mathcal{L}$ depends on two factors. First, we choose the dimension of $\mathcal{L}$ by setting k:

$$k = \lceil 1 - 2t \log_d |\alpha| - \log_d \delta \rceil. \tag{2.80}$$

Note that the maximum precision that can be required from the method for given t is obtained by setting $k = t$, so that $\delta_{\max} = 2^{-t(1+2\log_d|\alpha|)+1}$.

Next we find an $\mathcal{L}$ for which $Z(\mathcal{L})$ is not too large. The probability of obtaining a small enough $Z(\mathcal{L})$ can be analyzed as in [BG16] by evaluating the

expectation value of $Z(\mathcal{L})$ over all possible $\mathcal{L} \in \mathcal{F}_d^t$:

$$\begin{aligned} E(Z(\mathcal{L})) &= 1 + \sum_{\vec{x} \in \mathcal{F}_d^t / \{0\}} \left\langle \vec{\tilde{0}}^t \right| C_{\vec{x}} \left| \vec{\tilde{0}}^t \right\rangle E(I_{\mathcal{L}}(\vec{x})) \\ &= 1 + \frac{(d^k - 1)}{(d^t - 1)} (Z(F_d)^t - 1) \\ &= (1 + \frac{d^k - 1}{d^t - 1} (d^t |\alpha|^{2t} - 1)) \\ &\leq (1 + d^k |\alpha|^{2t}) \end{aligned} \tag{2.81}$$

Here $I_{\mathcal{L}}(\vec{x})$ is a indicator function, i.e., it is equal to 1 when $x \in \mathcal{L}$ and 0 otherwise. The second equal sign stands because the expectation value of $I_{\mathcal{L}}(x)$ for a fixed x is $\frac{d^k-1}{d^t-1}$ and

$$\begin{aligned} \sum_{x \in \mathcal{F}_d^t / \{0\}} \left\langle \vec{\tilde{0}}^t \right| C_{\vec{x}} \left| \vec{\tilde{0}}^t \right\rangle &= \sum_{x \in \mathcal{F}_d^t} \left\langle \vec{\tilde{0}} \right| C_{\vec{x}} \left| \vec{\tilde{0}} \right\rangle - 1 \\ &= \left(\left\langle \tilde{0} \right| \sum_{x=0}^{d-1} C^x \left| \tilde{0} \right\rangle \right)^t - 1. \end{aligned} \tag{2.82}$$

From eq. (2.80) we have $d^k \alpha^{2t} = O(1)$ so $E(Z(\mathcal{L})) = O(1)$. Therefore from Markov's inequality we obtain

$$\begin{aligned} &\text{Prob}(Z(\mathcal{L}) \leq (1 + d^k |\alpha|^{2t})(1 + \delta)) \\ &> 1 - \frac{E(Z(\mathcal{L}))}{(1 + d^k |\alpha|^{2t})(1 + \delta)} \geq 1 - \frac{1}{1 + \delta} > \delta. \end{aligned} \tag{2.83}$$

Randomly choosing δ^{-1} subspaces gives an $\mathcal{L}$ such that:

$$Z(\mathcal{L}) \leq (1 + d^k |\alpha|^{2t})(1 + \delta) \tag{2.84}$$

and hence satisfying eq.(2.78), with high probability.

The upper bound for the approximate stabilizer rank of a t-qudit magic state given by the above method is:

$$\chi'(t) = d^k = O(\delta^{-1} |\alpha|^{-2t}). \tag{2.85}$$

It is worth-mentioning that Bravyi etc. made a similar generalization for constructing stabilizer decompositions of magic states and finding upper bound of approximate stabilizer rank concurrently [BBC$^+$19]. They also directly upper-bounded the approximate stabilizer rank with *stabilizer fidelity*, which is the overlap we discussed here. The difference is that their focus is for multi-qubit magic states and instead we focus on single-qudit magic states. In the paper they defined a class of Clifford magic state and worked out the stabilizer decomposition for $CCZ\,|+\rangle^{\otimes 3}$ explicitly. Their work, together with ours, highlights the essential role of the maximum stabilizer overlap with the magic state in bounding the approximate stabilizer rank.

In the qubit case an explicit sum formula was given for $Z(\mathcal{L})$ with 2^k terms, and hence the cost of evaluating $Z(\mathcal{L})$ is $O(2^k)$. What is the cost of evaluating $Z(\mathcal{L})$ for arbitrary d. In the following section 2.7.2, we give an explicit formula for $Z(\mathcal{L})$ as a sum of products, and hence the cost of evaluating $Z(\mathcal{L})$ for arbitrary d is $O(d^{k+1})$.

d	M_d	p	$\lvert\alpha(d)\rvert$	$\lvert\alpha(d)\rvert$	$d^{\kappa t}$
2	$\mathrm{diag}(1, e^{i\pi/4})$	0	$\cos\pi/8$	0.92388	$2^{0.23t}$
3	$\mathrm{diag}(e^{2\pi i/9}, 1, e^{-2\pi i/9})$	0	$\frac{1+2\cos(2\pi/9)}{3}$	0.84403	$3^{0.31t}$
5	$\mathrm{diag}(\omega^{-2}, \omega, \omega^{-1}, \omega^{-2}, \omega^{-1})$	4	$\frac{3+2\cos(2\pi/5)}{5}$	0.723607	$5^{0.40t}$
7	$\mathrm{diag}(\omega^{3}, \omega^{-2}, 1, \omega^{3}, \omega^{1}, \omega^{2}, 1)$	3	$\frac{1+6\cos(2\pi/7)}{7}$	0.677277	$7^{0.40t}$

Table 2.1: The matrices M_d, optimal value of p and approximate stabilizer rank scaling comparison for $d = 2, 3, 5, 7$. Here $\kappa = -2\log_d \alpha$ so that $d^{\kappa t} = \alpha^{-2t}$. Here the ω for $d = 5$ and $d = 7$ rows are $e^{2\pi i/5}$ and $e^{2\pi i/7}$ respectively.

2.7 Detailed calculations of $\phi(p,d)$ and $Z(\mathcal{L})$

2.7.1 Magnitude and phase of magic state inner product with orbit representatives of nonorthogonal decompositions

Here we compute equation 2.75. We begin with $d = 3$. In this case we need only tabulate the inner product for three values of p:

$$\begin{aligned}
\langle +| Z^{-p} |M_3\rangle &= \langle +| Z^{-p} M_3 |+\rangle \\
&= \frac{1}{d}\mathrm{Tr}(Z^{-p}M_3) \\
&= \frac{1}{d}\left(e^{2\pi i/9} + e^{-2\pi ip/3} + e^{2\pi i/3(2p-\frac{1}{3})}\right) \\
&= \frac{1}{3}\left(1 + 2\cos\left(\frac{2\pi}{9}\right)\right) \quad p = 0 \\
&= \frac{1}{3}e^{\frac{i\pi}{3}}\left(2\cos\left(\frac{\pi}{9}\right) - 1\right) \quad p = 1 \\
&= \frac{1}{3}e^{\frac{2i\pi}{3}}\left(1 + 2\cos\left(\frac{4\pi}{9}\right)\right) \quad p = 2.
\end{aligned} \tag{2.86}$$

The largest magnitude overlap is obtained for $p = 0$.

Now we consider general prime $d > 3$. Given the expression for M_d we can write:

$$\langle +| Z^{-p} |M_d\rangle = \frac{\omega^{\frac{1}{d}\binom{d+1}{4}}}{d}\sum_{j=0}^{d-1}\omega^{\phi(j)} \tag{2.87}$$

where $\phi(j)$ is a cubic in j given by:

$$\phi(j) = \binom{j}{3} - \frac{j}{d}\binom{d}{3} - pj. \tag{2.88}$$

The evaluation of cubic gauss sums is not as straightforward as for quadratic gauss sums. However, we can obtain a closed form for the phase of the sum by depressing

the cubic to remove the quadratic term. In this case this is particularly simple:

$$\begin{aligned}\phi'(j) &= \phi(j+1)\\ &= \binom{j+1}{3} - \frac{j+1}{d}\binom{d}{3} - pj - p\\ &= \bar{6}j\left(j^2 - 1 - (d-1)(d-2) - 6p\right) - \frac{1}{d}\binom{d}{3} - p\\ &= \bar{6}j\left(j^2 - \psi(d,p)\right) - \frac{1}{d}\binom{d}{3} - p,\end{aligned} \tag{2.89}$$

where:

$$\psi(d,p) = d^2 - 3d + 3 + 6p. \tag{2.90}$$

Then:

$$\langle +|\, Z^{-p}\, |M_d\rangle = \frac{\omega^{\frac{1}{d}\binom{d+1}{4} - \frac{1}{d}\binom{d}{3} - p}}{d} \sum_{j=0}^{d-1} \omega^{\bar{6}j\left(j^2 - \psi(d,p)\right)} \tag{2.91}$$

The magnitude of this expression can be determined from the sum, which is real:

$$\begin{aligned}S &= \frac{1}{d}\sum_{j=0}^{d-1} \omega^{\bar{6}j\left(j^2-\psi(d,p)\right)}\\ &= \frac{1}{d} + \frac{1}{d}\sum_{j=1}^{(d-1)/2} \omega^{\bar{6}j\left(j^2-\psi(d,p)\right)} + \frac{1}{d}\sum_{j=(d+1)/2}^{d-1} \omega^{\bar{6}j\left(j^2-\psi(d,p)\right)}\\ &= \frac{1}{d} + \frac{1}{d}\sum_{j=1}^{(d-1)/2} \left(\omega^{\bar{6}j\left(j^2-\psi(d,p)\right)} + \omega^{\bar{6}(d-j)\left((d-j)^2-\psi(d,p)\right)}\right)\\ &= \frac{1}{d} + \frac{1}{d}\sum_{j=1}^{(d-1)/2} \left(\omega^{\bar{6}j\left(j^2-\psi(d,p)\right)} + \omega^{-\bar{6}j\left(j^2-\psi(d,p)\right)}\right)\\ &= \frac{1}{d} + \frac{2}{d}\sum_{j=1}^{(d-1)/2} \cos\frac{2\pi}{d}\bar{6}j\left(j^2 - \psi(d,p)\right)\end{aligned} \tag{2.92}$$

While this shows that the sum is real, it does not guarantee that it is positive, and hence the phase of the inner product, up to a sign, is given by:

$$\omega^{\frac{1}{d}\binom{d+1}{4} - \frac{1}{d}\binom{d}{3} - p} = \omega^{\frac{1}{d}\binom{d}{4} - p}. \tag{2.93}$$

2.7.2 Evaluation of $Z(\mathcal{L})$

As we've obtained in eq.(2.79):

$$\begin{aligned} Z(\mathcal{L}) &= \sum_{x\in\mathcal{L}} \left\langle \tilde{0}^t \right| C_{\vec{x}} \left| \tilde{0}^t \right\rangle \\ &= \sum_{x\in\mathcal{L}} \prod_{l=1}^{t} \left\langle \tilde{0} \right| C^{x_l} \left| \tilde{0} \right\rangle \\ &= \sum_{x\in\mathcal{L}} \prod_{l=1}^{t} \left\langle \tilde{0} | \tilde{x}_l \right\rangle \end{aligned} \tag{2.94}$$

We can see that this quantity is a function of the values $\left\langle \tilde{0}|\tilde{1}\right\rangle$,...,$\left\langle \tilde{0}\middle| d\tilde{-}1\right\rangle$. We label the phase of $\left\langle \tilde{0}|\tilde{j}\right\rangle$ by β_j for all j, where $\beta_0 = 1$. Using eqn. (2.73) $Z(\mathcal{L})$ can be rewritten in the following form:

$$\begin{aligned} Z(\mathcal{L}) &= \sum_{x\in\mathcal{L}} \prod_{l=1}^{t} \left\langle \tilde{0} | \tilde{x}_l \right\rangle \\ &= \sum_{x\in\mathcal{L}} \prod_{l=1}^{t} \beta_{x_l} \sqrt{\frac{1+(d-1)\delta_{0,x_l}}{d}} \\ &= \sum_{x\in\mathcal{L}} \frac{\Pi_{l=1}^{t} \beta_{x_l}}{d^{(t-|x|)/2}} \\ &= \sum_{x\in\mathcal{L}} \frac{\Pi_{j=1}^{d-1} \beta_j^{|x|_j}}{d^{(t-|x|)/2}} \end{aligned} \tag{2.95}$$

where $|x|$ is the Hamming weight of codeword x in code $\mathcal{L}$, i.e. the number of nonzero elements in the codeword. $|x|_j$ means the number of digits in string x that equals to j. If we regard $\mathcal{L}$ as a linear code, then the qubit case $Z(\mathcal{L})$ is exactly the weight enumerator of the code. For qudit case, it depends on the Hamming weight as well as the β_js. Now let's calculate an explicit expression for the β_js.

For $d = 3$ case, we specifically obtain $\beta_1 = e^{\pi i/18}$ and $\beta_2 = e^{-\pi i/18}$.

For $d > 3$ case, we assume our initial stabilizer state $\left|\tilde{0}\right\rangle = Z^a \left|+\right\rangle$. And

$$\beta_j = \sqrt{d} \left\langle \tilde{0}|\tilde{j}\right\rangle = \sqrt{d} \left\langle \tilde{0}\right| C^j \left|\tilde{0}\right\rangle \tag{2.96}$$

where the C for Campbell's choice of $|M_d\rangle$ is simply $\omega^{-\bar{3}}XP$ according to eq.(2.48) and subsection C of section 2.5. We can calculate $(XP)^j$ as

$$\begin{aligned}(XP)^j &= \sum_k \omega^{\sum_{l=0}^{j-1}\binom{k+l}{2}} |k+j\rangle\langle k| \\ &= \omega^{\bar{6}(j^3-3j^2+2j)} \sum_k \omega^{\bar{2}(jk^2+(j^2-2j)k)} |k+j\rangle\langle k| .\end{aligned} \tag{2.97}$$

Therefore we can rewrite C^j as

$$\begin{aligned}C^j &= \omega^{-\bar{3}j}(XP)^j \\ &= \omega^{\bar{6}(j^3-3j^2)} \sum_k \omega^{\bar{2}(jk^2+(j^2-2j)k)} |k+j\rangle\langle k| .\end{aligned} \tag{2.98}$$

Then we can calculate β_j as

$$\begin{aligned}\beta_j &= \sqrt{d}\,\langle +| Z^{-a} C^j Z^a |+\rangle \\ &= \frac{\omega^{\bar{6}(j^3-3j^2)}}{\sqrt{d}} \langle k''| \sum_{k''} \omega^{-ak''} \sum_k \omega^{\bar{2}(jk^2+(j^2-2j)k)} |k+j\rangle \\ &\langle k| \sum_{k'} \omega^{ak'} |k'\rangle \\ &= \frac{\omega^{\bar{6}(j^3-3j^2-6aj)}}{\sqrt{d}} \sum_k \omega^{\bar{2}(jk^2+(j^2-2j)k)}\end{aligned} \tag{2.99}$$

This is a quadratic Gauss sum times a phase. Using eq.(2.58) and (2.59) for $f = j$ and $k + g = \bar{2}(j^2 - j)$, we obtain:

$$\sum_k \omega^{\bar{2}(jk^2+(j^2-2j)k)} = \begin{cases} \omega^{-\bar{2}^3 j(j-2)^2}\left(\frac{2j}{d}\right) \\ \omega^{-\bar{2}^3 j(j-2)^2} i\left(\frac{2j}{d}\right). \end{cases} \tag{2.100}$$

The final expression of β_j in terms of a is:

$$\begin{aligned}\beta_j &= \begin{cases} \omega^{\bar{6}j^3-\bar{2}j^2-aj}\omega^{-\bar{2}^3j(j-2)^2}(\frac{2j}{d}) \\ \omega^{\bar{6}j^3-\bar{2}j^2-aj}\omega^{-\bar{2}^3j(j-2)^2}i(\frac{2j}{d}) \end{cases} \\ &= \begin{cases} \omega^{(\bar{6}-\bar{2}^3)j^3-(a+\bar{2})j}(\frac{2j}{d}), & d \equiv 1(\mod 4) \\ \omega^{(\bar{6}-\bar{2}^3)j^3-(a+\bar{2})j}i(\frac{2j}{d}), & d \equiv 3(\mod 4) \end{cases}\end{aligned} \tag{2.101}$$

where again $(\frac{\bar{2}j}{d})$ is the Legendre symbol.

Chapter 3

Feynman path-integral Simulation with stabilizer projector decomposition of unitaries

The work in this chapter is published in [HL21].

The original Bravyi-Gosset algorithm can only tackle non-Clifford gates that are in the third level of the Clifford hierarchy [ZLC00], because only these gates can be implemented by injecting magic states. Alternatively, one can use the method based on Sum-over-Cliffords for unitaries [BBC$^+$19]:

$$U = \sum_j a_j K_j. \tag{3.1}$$

where K_j's are all Clifford gates and a_j's are some complex coefficients. This decomposition of unitaries is able to deal with unitaries that are beyond the third level of the Clifford hierarchy.

The Bravyi-Gosset algorithm has the property that adding one non-Clifford gate increases the number of stabilizer states in the expansion of the wavefunction.

On average, one T gate multiplies the number of stabilizer states one needs to keep track of by a factor of $2^{0.47}$.

The way to avoid branching is to insert stabilizer projectors. No matter how many stabilizer states there are in the decomposition from the previous step, a sum of κ stabilizer projectors will always turn them into a sum of κ stabilizer states. In Section 3.3 we introduce an algorithm that combines the stabilizer projector insertion with the recursive Feynman path-integral approach. By replacing the non-Clifford gates with stabilizer projectors, we incorporate the advantage of stabilizer-based simulation into Feynman path-integral simulation.

Unentangled quantum states have a polynomial space description, as do stabilizer states. The Gottesman-Knill theorem allows classical computers to simulate quantum systems with unlimited entanglement provided all states are stabilizer states. However, in a sense, Clifford operations don't unleash the true power of entanglement and can be interpreted in a locally causal description [Bel64, Cuf17]. In fact, any sets of operations that can be classically simulated have a locally causal description [Cuf17]. What's more, the T gate is hard for stabilizer-based simulation but it doesn't cause trouble for monotone simulation methods, $CNOT$ is hard for monotone methods while being easy for stabilizer-based algorithms. Due to the existence of different of the descriptive formalisms of quantum mechanics, specific operations can be easy in one formalism but hard in another. Might it be advantageous to have a formalism that combines the locally causal properties of these different classically simulatable descriptions? In this chapter, we step into the realm of combining monotone methods and stabilizer-based descriptions. We obtained the polynomial-space SPIR algorithm, which improves the time scaling of the previous polynomial-space recursive Feynman path-integral simulation [AC17].

The Google "quantum supremacy" experiment [AAB$^+$19] implemented 20 cycles of random gates chosen from a universal gate set. The single-qubit Clifford gates in the gate set are:

$$\sqrt{X} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \tag{3.2}$$

and

$$\sqrt{Y} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}. \tag{3.3}$$

The non-Clifford single-qubit gate is:

$$\sqrt{W} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -\sqrt{i} \\ \sqrt{-i} & 1 \end{pmatrix}. \tag{3.4}$$

and the two-qubit gate is

$$\mathrm{fSim}(\pi/2, \pi/6) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 0 & e^{-i\pi/6} \end{pmatrix}. \tag{3.5}$$

which is non-Clifford.

Each cycle contains one round of single-qubit gates on every qubit and one round of two-qubit gates in fixed repeating configurations [BIS$^+$18, AAB$^+$19]. A linear cross-entropy fidelity of ~ 0.002 is achieved after measurement. The set of single-qubit gates used contains two Clifford gates and one non-Clifford gate. The two-qubit gates are all non-Clifford gates. One restriction on the random selection of single-qubit gates is that the same gate does not act on the same qubit in two consecutive cycles. The non-Clifford depth of the circuit is 40.

For this random circuit, there are 1113 single-qubit gates and 430 two-qubit gates. Out of the 1113 single-qubit gates, one third of them are non-Clifford. This means there are roughly 800 non-Clifford gates in the circuit. This is beyond the reach of the Bravyi-Gosset or the Sum-over-Clifford methods.

Motivated by this fact, the algorithms we introduce in this paper can serve as an extension to the Bravyi-Gosset or the Sum-over-Clifford method for circuits with a larger fraction of non-Clifford gates. However, as we shall see, the large fraction of non-Clifford gates of the circuits in [AAB$^+$19] remains an obstacle for

stabilizer-based methods.

Before presenting our algorithm, we first introduce the method we use for obtaining stabilizer projector decomposition of unitaries by discussing several numerical results on the upper bound of SPM rank for magic states in the density matrix form. This motivates the use of stabilizer projectors as over-complete basis.

3.1 l_0 measure for stabilizer pseudomixture

We now consider some examples of SPM rank. We adapted the simulated annealing method introduced in [BSS16] for finding stabilizer rank. Our implementation is adapted slightly from [BSS16] to use 4^n dimensional real vectors instead of 2^n dimensional complex vectors and optimizes the SPM rank of a matrix rather than the stabilizer rank of a state. We computed upper bounds on the SPM rank of the resource states $|H\rangle$, $|T\rangle$, $|CS\rangle = CS\,|+\rangle^{\otimes 2}$, $|CCZ\rangle = CCZ\,|+\rangle^{\otimes 3}$. We also computed upper bounds on the SPM rank of states of the form $\left|\tilde{M}_k\right\rangle\left\langle\tilde{M}_k\right|$ where $\left|\tilde{M}_k\right\rangle = \cos\left(\frac{\pi}{2^k}\right)|0\rangle + \sin\frac{\pi}{2^k}|1\rangle$. The state is equivalent to the H-type magic state when $k = 3$. The rank upper bounds for these states are shown in Table 3.1. Here we give decompositions for a few typical non-Clifford gates in the third level of the Clifford hierarchy:

The SPM rank of $|H\rangle\langle H|$ is at most 3 [BSS16]:

$$|H\rangle\langle H| = \frac{\sqrt{2}}{2}|+\rangle\langle +| + \frac{1}{2}|0\rangle\langle 0| + \frac{1-\sqrt{2}}{2}|1\rangle\langle 1| \tag{3.6}$$

The SPM rank-5 decomposition of $|H\rangle\langle H|^{\otimes 2}$ is:

$$\begin{aligned}|H\rangle\langle H|^{\otimes 2} = &\left(\frac{1}{2} - \frac{\sqrt{2}}{4}\right)\left|\Psi^+\right\rangle\left\langle\Psi^+\right| - \frac{\sqrt{2}}{4}\left|\Psi^-\right\rangle\left\langle\Psi^-\right| \\ &+ \frac{1-\sqrt{2}}{2}\left|\Phi^{+i}\right\rangle\left\langle\Phi^{+i}\right| + \frac{\sqrt{2}}{2}|+i,+\rangle\langle +i,+| \\ &+ \frac{\sqrt{2}}{2}|+,+i\rangle\langle +,+i|\end{aligned} \tag{3.7}$$

where where $|+i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle)$, $|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$, $|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$

n	$\|H\rangle$	$\|T\rangle$	$\|CS\rangle$	$\|CCZ\rangle$	$\left\|\tilde{M}_4\right\rangle$	$\left\|\tilde{M}_5\right\rangle$
1	3/2	4/3	NA	NA	4	4
2	5/3.05	7/4.98	6/4.84	NA	6	6
3	13/4.92	24/9.60	NA	7/6.53	14	15

Table 3.1: Upper bounds of SPM rank/RoM squared for various resource states. States $\left|\tilde{M}_k\right\rangle = \cos\left(\frac{\pi}{2^k}\right)|0\rangle + \sin\frac{\pi}{2^k}|1\rangle$ and therefore $|H\rangle = \left|\tilde{M}_3\right\rangle$.

and $\left|\Phi^{+i}\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + i\,|11\rangle)$.

The upper bound of SPM rank for $|CS\rangle$ state is 6:

$$\begin{aligned}|CS\rangle\langle CS| =& \frac{1}{2}|+0\rangle\langle+0| + \frac{1}{2}|-i,1\rangle\langle-i,1| + \frac{1}{2}|{+}{+}\rangle\langle{+}{+}| \\ &- \frac{1}{2}|+,-i\rangle\langle+,-i| \\ &- \frac{1}{2}(\frac{1}{\sqrt{2}}(|+0\rangle - |-1\rangle))(\frac{1}{\sqrt{2}}(\langle+0| - \langle-1|)) \\ &+ \frac{1}{2}(\frac{1}{\sqrt{2}}(|+i,+\rangle + i\,|-i,-\rangle)) \\ &(\frac{1}{\sqrt{2}}(\langle+i,+| - i\,\langle-i,-|)).\end{aligned} \tag{3.8}$$

and the upper bound of SPM rank for $|CCZ\rangle$ state is 7:

$$\begin{aligned}&|CCZ\rangle\langle CCZ| = |{+}{+}{+}\rangle\langle{+}{+}{+}| \\ &\frac{1}{2}(\frac{1}{\sqrt{2}}(|+11\rangle - |-00\rangle))(\frac{1}{\sqrt{2}}(\langle+11| - \langle-00|)) \\ &- \frac{1}{2}|1{+}{+}\rangle\langle1{+}{+}| - \frac{1}{2}|+\rangle\left|\Phi^+\right\rangle\langle+|\left\langle\Phi^+\right| \\ &+ \frac{1}{2}|1\rangle(\frac{1}{\sqrt{2}}(|+0\rangle + |-1\rangle))\langle1|(\frac{1}{\sqrt{2}}(\langle+0| + \langle-1|) \\ &+ \frac{1}{2}(\frac{1}{\sqrt{2}}(|+--\rangle - |-++\rangle))(\frac{1}{\sqrt{2}}(\langle+--| - \langle-++|)) \\ &- \frac{1}{2}(\frac{1}{\sqrt{2}}(|+i-i-i\rangle - |-i+i+i\rangle)) \\ &(\frac{1}{\sqrt{2}}(\langle+i-i-i| - \langle-i+i+i|))\end{aligned} \tag{3.9}$$

Other decompositions of these upper bounds are given in Appendix A.

The new stabilizer decomposition of $|H\rangle\langle H|^{\otimes 2}$ improves the efficiency of using classical processing and a small quantum computer to simulate a bigger quantum

computer under the Pauli-Based computation (PBC) model ([BSS16]).

If one were to decompose $|H\rangle\langle H|^{\otimes n}$ into $\left\lfloor \frac{n}{2} \right\rfloor$ $|H\rangle\langle H|^{\otimes 2}$s, one would get a upper bound on the SPM rank of $|H\rangle\langle H|^{\otimes n}$ of $5^{n/2} = 2.236^n$. This is worse than the scaling given by the Howard-Campbell result, which is upper bounded by the RoM squared for state $\left|H^{\otimes 5}\right\rangle$: $\mathcal{R}(\left|H^{\otimes 5}\right\rangle)^{2n/5} = 1.685^n$, but our scaling provides an exact strong simulation. And as we discussed in Section 1.6 of the introduction, because $\frac{1}{\delta^2}$ provides an exponential overhead if one wants the total variational distance bounded, the strong simulation based on our exact l_0 decomposition will scale better for some application.

Assume there are z CCZ gates in the circuit we are trying to simulate. Note [HC17] $\mathcal{R}(|CCZ\rangle) = 2.555$, this yields the overhead scaling $\mathcal{R}(|CCZ\rangle)^{2z} = 6.531^z$, which is only slightly smaller than our strong simulation scaling 7^z. However, our strong simulation scaling doesn't have any overhead in terms of precision, this makes our l_0 norm decomposition preferable in terms of simulating CCZ circuits with small precision or small z. It is also known [Jon13, HC17] that the RoM of these states obeys:

$$\mathcal{R}(|H\rangle^{\otimes 3}) \leq \mathcal{R}(|CCZ\rangle) \leq \mathcal{R}(|H\rangle^{\otimes 4}), \tag{3.10}$$

whereas the upper bound for the SPM rank of $|CCZ\rangle$ (7) is smaller than the upper bound on the SPM rank of $|H\rangle^{\otimes 3}$ (12). If these bounds are tight, it means the exact stabilizer pseudomixture decomposition is more efficient to simulate a Clifford+CCZ circuit by injecting $|CCZ\rangle$ states.

Next, we will use the overcomplete basis of stabilizer projectors for decomposing unitaries, out of which we propose a polynomial space algorithm for simulating quantum circuits.

3.2 Stabilizer Projector decomposition for unitaries

We denote the full set of n-qubit stabilizer states as STAB, which gives an overcomplete basis for n-qubit states. The stabilizer projector basis, $\{|\phi_i\rangle\langle\phi_i|\}_{\phi_i \in \text{STAB}}$,

provides an overcomplete basis for n-qubit operators. This basis is used to decompose density matrices in [HC17] and forms stabilizer pseudomixtures, where the coefficients are all real. They can also be used for decomposing a unitary matrix as follows:

$$U = \sum_i c_i \left|\phi_i\right\rangle \left\langle\phi_i\right| . \tag{3.11}$$

where the c_i's are complex coefficients and the $|\phi_i\rangle$s are stabilizer states.

We define the stabilizer projector rank of a unitary operator as follows:

$$\kappa(U) = \min\{k : U = \sum_{i=1}^{k} c_i \left|\phi_i\right\rangle \left\langle\phi_i\right|\}. \tag{3.12}$$

Because the matrix rank of a single stabilizer projector is one while the rank of a unitary matrix is 2^n, we at least need 2^n terms to decompose a unitary operator. This implies that the stabilizer projector rank of an n-qubit unitary is at least 2^n. Therefore it is clear that this quantity is not a measure of non-Cliffordness of the gate, because $\kappa \neq 1$ for Clifford gates. The stabilizer projector decomposition provides a representation of an arbitrary unitary that enables the action of the operator on the stabilizer states to be computed with cost $O(n^3\kappa)$.

We give upper bounds for the stabilizer projector rank for the non-Clifford gates used in the Google quantum supremacy experiment [AAB+19] in Table 3.2. These non-Clifford gates are not diagonal, therefore the stabilizer projector decomposition is non-trivial. For the operators that have a non-trivial decomposition, the rank for a decomposition over an over-complete set is hard to compute exactly. However, the advantage of having many more states available in the over-complete basis may still provide a sparser decomposition than the decomposition over standard orthogonal basis. One can see this by realizing that operator $\sqrt{W} \otimes \sqrt{W}$ has support on 9 of the 16 two-qubit Pauli operators, but we have already found a stabilizer projector decomposition that only has 6 non-zero terms.

Gate	Upper bounds	Probability
$\mathrm{fSim}(\pi/2, \pi/6)$	4	4/9
$\mathrm{fSim}(\pi/2, \pi/6) \cdot \sqrt{W_1}\sqrt{W_2}$	10	1/9
$\mathrm{fSim}(\pi/2, \pi/6) \cdot \sqrt{W_1}$	12	4/9
$\sqrt{W_1}\sqrt{W_2}$	6	NA

Table 3.2: Upper bounds on stabilizer projector rank and probability of occurrence among all two-qubit gates for the non-Clifford gates in the Google supremacy experiment. See Appendix D for one of the decompositions achieving these upper bounds for each gate.

3.3 Algorithms

In this Section we discuss two simulation algorithms based on the stabilizer projector decomposition for unitaries introduced in Section 3.2. We call the first algorithm the stabilizer path integral recursion (SPIR) method and the second algorithm the stabilizer projector contraction (SPC) method. Our polynomial-space SPIR algorithm improves the upper bound of the Aaronson-Chen polynomial-space Feynman path-integral simulation from $O((2d)^n)$ to $O((2d_{nc})^n)$. Here we omitted the non-exponential factors in the scaling and the exponents are exactly n for diagonal non-Clifford gates but αn for non-diagonal cases where $\alpha > 1$ is a constant the depends on the stabilizer projector decomposition, see Table 3.3 for the exact number.

3.3.1 Polynomial-space SPIR method

To evaluate an amplitude $\langle x| U |0^n\rangle$, one can replace all the non-Clifford gates in the circuit for U with the projector decomposition of eq. (3.11) such that they form layers of stabilizer projectors. The amplitude then is given as a sum of products of Clifford operators contracted by stabilizer states. Calculating the inner product between two n-qubit stabilizer states takes $O(n^3)$ time [AG04, GMC12]. Therefore the exponential part of the scaling of this simulation comes from the number of terms in the stabilizer projector decomposition of each gate.

This procedure is analogous to a path integral formulation in eq. (1.18), except here the unitaries are all Clifford gates and the computational basis states are replaced by general stabilizer states. Therefore the same structure of repeated calcu-

lations is also present in this procedure. We can use the recursion implementation of [AC17] as follows. In general, if we divide the circuit U we want to simulate into 2 depth $\lceil \frac{d_{nc}}{2} \rceil$ sub-circuits U_1 and U_2 with a depth-1 layer of non-Clifford gates $U_{nc} = \sum_i c_i |\phi_i\rangle \langle\phi_i|$ in the middle, we can now calculate an amplitude. The Clifford gates in the circuit will be ignored because they can be absorbed into the stabilizer states in $O(n)$ time.

$$\begin{aligned} \langle x| U |0^n\rangle &= \langle x| U_2 U_{nc} U_1 |0^n\rangle \\ &= \sum_i c_i \langle x| U_2 |\phi_i\rangle \langle\phi_i| U_1 |0^n\rangle . \end{aligned} \tag{3.13}$$

The stabilizer projector rank of U_{nc}, which acts on n qubits, is $\kappa_n = 2^k$, which we assume is the maximum among all non-Clifford layers without loss of generality. We keep splitting the circuit at the non-Clifford layers, obtaining the following recursion relation

$$T(d_{nc}) = 2\kappa_n T(d_{nc}/2), \tag{3.14}$$

The base case, which only contains one layer of non-Clifford gates, takes $T(1) = O(n^3 2^k)$ operations because one needs to calculate all 2^k inner-products for one layer of stabilizer projectors. Therefore the total number of operations to calculate this amplitude will be

$$T(d_{nc}) = O((2d_{nc})^{k+1} n^3) = O(\kappa_n d_{nc}^{k+1} n^3). \tag{3.15}$$

We summarize this algorithm as follows:

1. Divide the circuit into layers of Clifford gates and non-Clifford gates.

2. Find the middle ($[d_{nc}/2]$th) layer of non-Clifford gates, divide the whole circuit into two sub-circuits, and rewrite this non-Clifford layer into a sum of stabilizer projectors.

3. Recursively implement step 2 for each sub-circuit and for each stabilizer pro-

jector until one gets down to the base case. For the base case, the U_1 and U_2 in eq. (3.13) become Clifford layers C_1 and C_2. Calculate the inner-products for all 2^k projectors and sum them up to obtain the value for $\langle\phi_j| C_2 U_{nc} C_1 |\phi_k\rangle$.

4. Return the value obtained in the base cases from step 3 and recursively return the values to the level above, until one gets back to the root level and obtains the value for the final amplitude.

This recursive method, is in spirit the same as the original recursive Feynman path-integral simulation, therefore the space requirement is $O(n \log d_{nc})$, as we already discussed in Section 1.4.2. However, we manage to improve the time cost from $O((2d)^n)$ to $O((2d_{nc})^n)$, which makes it the state-of-the-art polynomial-space simulation of general random quantum circuits as far as we know. Previous work in [Shi17] is also a recursive method with polynomial space where the expensive gates are the non-diagonal gates. Their methods are more preferable for quantum Fourier transform circuits [Cop02, NC11] or Instantaneous Quantum Polynomial (IQP) circuits [SB09, BJS11] because the number of non-diagonal gates is $O(n)$. Meanwhile, in our method and the original recursive Feynman path-integral method, one can further reduce the time cost by compressing adjacent layers of diagonal gates into one layer, in which case the effective non-Clifford depth of the IQP circuit is only 1. Therefore, our algorithm can simulate IQP circuits in $O(2^n)$ time with polynomial space requirement.

In Subsection 3.3.2 we discuss an algorithm requiring exponential space, which may be used to combine with the Feynman-Schrodinger hybrid algorithm or the Sum-over-Clifford algorithm.

3.3.2 Exponential-space SPC method

The SPIR algorithm in the previous section is a Feynman-path type algorithm that calculates a single amplitude at a time. One can also evolve the whole state altogether as in the Schrodinger method using the stabilizer projector decomposition.

One first partitions the circuit into alternating rounds of one layer of Clifford

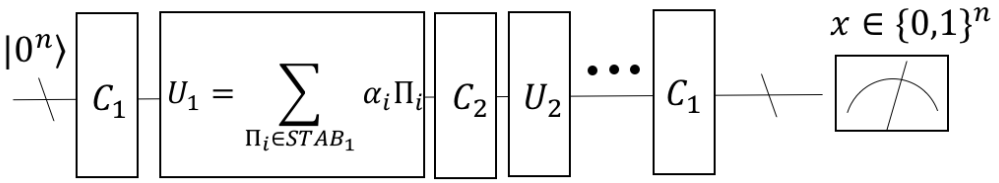

Figure 3.1: Alternating layers of Clifford and non-Clifford gates for implementing the stabilizer projector contraction (SPC) simulation. Each layer doesn't have to have constant depth. In fact for Clifford layers it doesn't matter because they can all be efficiently processed by the Gottesman-Knill theorem. If one run into a non-Clifford layer, one projects the stabilizer states obtained from the previous round onto the stabilizer projectors by taking inner products between stabilizer states [GMC12]. If a non-Clifford layer has depth bigger than 1, then one can always chop them further into thinner slices such that the gates in each slice can be efficiently written down as a sum of stabilizer projectors.

gates followed by one layer of non-Clifford gates, as shown in 3.1. Then one runs the following exponential-space SPC algorithm:

1. Start with computational basis state $|0\rangle^{\otimes n}$ and use the Gottesman-Knill theorem to evolve through the first Clifford layer at the beginning of the circuit, obtaining a stabilizer state $|\psi_1\rangle = |\phi\rangle$. This step takes a $O(m_1 n^2)$ operations where m_1 is the number of Clifford gates in the first layer.

2. Decompose the next layer of non-Clifford gates into a sum of stabilizer projectors as in eq. (3.12), and project the stabilizer state we obtained in step 1 onto these projectors. The new state after this layer of non-Clifford gates is described as a sum of stabilizer states $|\psi_2\rangle = \sum_i c_i \langle\phi_i|\phi\rangle |\phi_i\rangle$. This step takes $O(\kappa_n n^3)$ operations.

3. Evolve each stabilizer state in the sum for $|\psi_2\rangle$ above through the next round of Clifford layer separately. Then project each one onto the next non-Clifford layer with stabilizer projector decomposition. This step take $O(\kappa_n^2 n^3)$ operations.

4. Iterate step 3 until the end of the circuit.

The overall number of operations is therefore $O(m_1 n^2 + \kappa n^3 + (d_{nc} - 1)\kappa^2 n^3) =$

$O(d_{nc}\kappa^2 n^3)$ because we iterate step 3 $O(d_{nc})$ times. The space complexity is $O(\kappa)$.

The limitation of the improvement of our algorithm is, as we discussed before, the stabilizer projector rank of each layer of gates is at least 2^n. Therefore, one wants to put as many non-Clifford gates as possible in one layer for stabilizer projector decomposition, and to ensure the gates act non-trivially on as many qubits as possible. In this way, one can avoid devoting resources to representing the identity. Even though, we remark that our SPC algorithm doesn't provide speedup compared to direct simulation or the improved stabilizer frame simulation [GM14]. We present this algorithm to shed light on the method we will develop in Chapter IV that improves the scaling with unitary gates turned sideways. As we will discuss, the key idea of the Feynman-Schrodinger hybrid algorithm applies naturally to our SPC method and can be therefore taken advantage of to reduce the space and time requirements.

3.3.3 Combination of SPC method with Feynman-Schrodinger hybrid algorithm algorithm

As introduced in Section 1.4.3, we can partition the initial circuit we need to simulate into two patches with an approximately equal number of qubits. The criterion for the partition is to make the number of entangling gates, x, that connects the two patches as small as possible. Then we decompose these entangling gates across the two patches into a sum of separable operations as in eq. (1.21). Now we have 2^x pairs of circuits, each of which contains two separate sub-circuits of $\sim n/2$ qubits. Then for each of these sub-circuits, we use our SPC method as a subroutine. Then we sum the amplitudes computed for all 2^x pairs. The space requirement of this algorithm is $\kappa_{n/2}$. The time requirement will be $O(2^x d_{nc}\kappa_{n/2}^2 n^3)$. As we mentioned before in Section 1.4.3, the space complexity is greatly reduced from $\sim 2^n$ to $\sim 2^{n/2}$, while the time complexity is reduced if $x < 2\log_2 \kappa_{n/2}$. This is why it is important that we choose two patches of relatively equal size with x as small as possible.

Algorithm \ cost	Time cost	Memory cost
Direct	$m2^n$	2^n
Feynman path	4^m	$m+n$
Hybrid [CZX$^+$18, MFIB18]	$2^{n/2+x}$	$2^{n/2+1}$
Recursive path integral [AC17]	d^n	$n \log d$
Tensor contraction [BISN17, CZH$^+$18]	$2^{tw(G)}$	2^{cw}
Stabilizer rank [BSS16, BG16]	$n^3 2^{0.47t}$	$2^{0.47t}$
SPIR method	$n^3(2d_{nc})^k$	$n \log d_{nc}$
SPC method	$d_{nc}2^{2k}n^3$	2^k

Table 3.3: Simulation cost for algorithms discussed in the introduction and developed in this chapter. Here, m is the total number of gates, x is the number of entangling gates across the patches, d is the circuit depth, t is the number of non-Clifford gates in the circuit, 2^k is the stabilizer projector rank for each layer of gates, where $k \sim n$, and d_{nc} is the non-Clifford gate-depth. $tw(G)$ is the treewidth of the undirected graph corresponding to the circuit [MS08]. cw is the contraction width corresponding to a certain contraction order of the tensor, which is defined as the size of the biggest clique formed along the contraction. Big-O notation is implicit in the table.

3.3.4 Combination of SPC method with Sum-over-Cliffords algorithm

The SPC algorithm is complementary to the original Sum-over-Cliffords algorithm when the number of non-Clifford gates in the circuit exceeds the number of qubits. From another point of view, we can use the Sum-over-Clifford method to improve the time scaling of the SPC algorithm using its advantages for circuits with a small number of non-Clifford gates. In fact, in step 1 and 2 of the SPC procedure we described above, the number of terms in the stabilizer decomposition of ψ_1 and ψ_2 suddenly goes from 1 to κ, which is of the order of 2^n. Therefore, we will replace replace step 1 and 2 in the SPC method section by evolving the Sum-over-Cliffords algorithm through the circuit until the number of Cliffords in the decomposition exceeds the stabilizer projector rank κ of the next non-Clifford layer. In the case of T-gates, this allows us to process $2.13n$ gates because one T-gate only adds a factor of $2^{0.47}$ to the rank on average. After that, we use steps 3 and 4 of the SPC procedure to prevent the rank from growing rapidly.

3.4 Comparison to Bravyi-Gosset algorithm for a family of circuits

In this Section we will compare the time cost of our SPIR method with the strong version of the Bravyi-Gosset algorithm for a specific family of circuits.

Both of the algorithms have an n^3 factor in the time cost, therefore it is sufficient to only compare the exponential part. The Bravyi-Gosset algorithm has exponential scaling $2^{0.47t}$, while our method has scaling $(2d_{nc})^k$.

We will compare our SPIR method to the Bravyi-Gosset algorithm for simulation of an ensemble of random circuits with alternating single-qubit and two-qubit rounds. Specifically we consider the family of circuits where the single qubit gate-set contains both Clifford gates and non-Clifford gates and the non-Clifford gates are chosen with probability p.

We analyse the performance for an example in the family where two-qubit gates are Control-Z (CZ) gates and single qubit gates are T gates with probability p and Clifford gates with probability $1-p$. The only non-Clifford gates in the circuit are the T gates. On average, for a circuit with d_{nc} cycles, there are $d_{nc}np$ T gates. We decompose the product of T gates and CZ gates in one cycle into a single unitary and perform a stabilizer projector decomposition. Therefore the effective depth in the SPIR method is $d_{nc}/2$. We also have $k = n$ because the product of T gates and CZ gates are all diagonal and can be represented exactly by the 2^n projectors of the computational basis states. Therefore the time complexities of our SPIR method and the Bravyi-Gosset algorithm are $(d_{nc})^n$ and $2^{0.47npd_{nc}}$ respectively. Hence the SPIR method will have a better scaling than the Bravyi-Gosset algorithm when

$$\log_2 d_{nc} \leq 0.47pd_{nc}, \tag{3.16}$$

giving

$$p \geq \frac{\log_2 d_{nc}}{0.47d_{nc}}. \tag{3.17}$$

This inequality is shown in Figure 3.2. Therefore if this inequality is satisfied, i.e.,

the density of non-Clifford gates is larger than this threshold with respect to the non-Clifford depth, the SPIR method has a better scaling, otherwise the Sum-over-Clifford method has the advantage.

Next we consider another example of the ensemble in which the two-qubit gates are Control-S (CS) gates, which is a non-Clifford gate. The number of non-Clifford gates in the circuits will be greatly increased and the non-Clifford depth is twice the number of cycles. In this case, for a $d_{nc}/2$ cycle circuit, there are on average $d_{nc}np/2$ T gates and $nd_{nc}/4$ CS gates. Meanwhile, due to the fact that the product of T gates and CS gates are all diagonal, we have $k = n$ for our algorithm. Therefore the time complexity of our algorithm and the Bravyi-Gosset algorithm are $(d_{nc})^n$ and $2^{0.47npd_{nc}/2+nd_{nc}/4}$ respectively. Hence our algorithm will have a better scaling when

$$\log d_{nc} \leq 0.47pd_{nc}/2 + d_{nc}/4. \tag{3.18}$$

Giving

$$p \geq \frac{2(\log d_{nc} - d_{nc}/4)}{0.47d_{nc}}. \tag{3.19}$$

This inequality is also shown in Figure 3.2.

The comparison between the SPC method and Bravyi-Gosset algorithm is straightforward: it is the comparison between κ_n^2 and the stabilizer rank. Now we assume T gates as the only non-Clifford resource. In this case, $\kappa_n = 2^n$ for one layer of T gates no matter how many qubits it acts on non-trivially. Meanwhile the stabilizer projectors in this decomposition are computational basis projectors $|x\rangle\langle x|$. In the Bravyi-Gosset algorithm, one T gate contributes a factor of $2^{0.47}$ to the time scaling. Hence, our algorithm performs better when $2^{2n} > 2^{0.47t}$, i.e, when the number of T gates is bigger than $4.26n$. If this is the case, we introduce the SPC method to complements the Sum-over-Clifford method, as we discussed in Section 3.3.4. In fact, if a stabilizer projector decomposition has all of its non-zero projectors orthogonal to each other, one can implement the SPC algorithm in $O(2^n)$ instead of $O(2^{2n})$ using fast fourier transform, as we will see in Section 4.1.2.

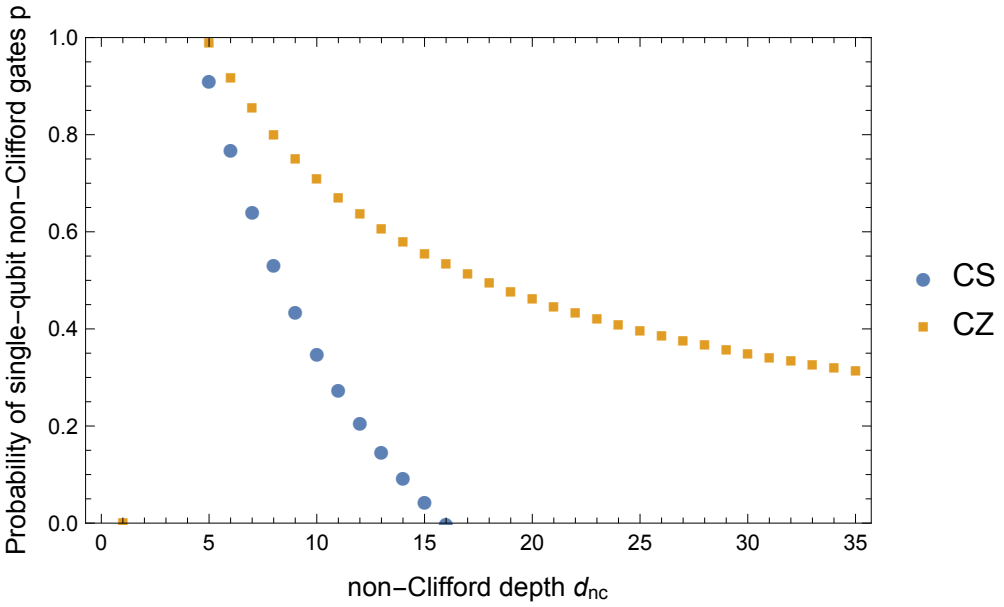

Figure 3.2: In this plot we consider the ensemble of random circuits with alternating rounds of single qubit and two qubit layers. We plot the threshold density (probability) of non-Clifford gates for the SPIR method to have a better scaling than the strong Bravyi-Gosset algorithm for such circuits of different non-Clifford depth as in eq. (3.17) and eq. (3.19). The two-qubit gates in comparison here are CZ and CS gates. Here the curve for CS is going negative for $d_{nc} > 16$, which means for any density of single-qubit non-Clifford gates, the scaling of the SPIR method is better than the strong Bravyi-Gosset algorithm. For $p = 1/3$ and CZ gates as two-qubit gates, the SPIR method starts to have a better scaling when the non-Clifford depth is bigger than 31. For $p = 1/3$ and CS gates, the SPIR method starts to have a better scaling when the non-Clifford depth is bigger than 11.

3.5 Quantum supremacy experiment

The random circuit in the Google supremacy experiment is one example in the ensemble of circuits we defined in Section 3.4. The two qubit gates are all non-Clifford gates. The one qubit gates are randomly selected from two Clifford gates and one non-Clifford gate. If we choose from these three gates with equal probability, we have $p = 1/3$. The non-Clifford depth of the supremacy circuit is 20 if we combine the single-qubit gates and two-qubit gates in one cycle and decompose them into stabilizer projectors altogether as in Table I.

Now we explicitly analyse the performance of our SPIR method for the supremacy circuits. In the average-case scenario, there will be roughly 10 $\mathrm{fSim}(\pi/2, \pi/6)\cdot$

$\sqrt{W_1} \cdot C_1$ gates (Or $\text{fSim}(\pi/2, \pi/6) \cdot C_2 \cdot \sqrt{W_2}$ gates), 2 $\text{fSim}(\pi/2, \pi/6) \cdot \sqrt{W_1}\sqrt{W_2}$ gates and 10 $\text{fSim}(\pi/2, \pi/6) \cdot C$ and 3 individual $\sqrt{W}$ gates in one cycle, where C, C_1, C_2 represent some Cliffords such as tensor product of I, $\sqrt{X}$ and $\sqrt{Y}$. Therefore, average-case stabilizer projector rank for a cycle is

$$\kappa_n = 2^k = 12^{10}4^{10}10^2 \cdot 18 = 2^{66.67} \approx 2^{1.25n}. \tag{3.20}$$

Therefore, the total cost for our SPIR method is roughly $O(2^{1.25n \log d_{nc}} n^3)$ and $O(40^{1.25n} n^3) = O(2^{6.7n} n^3)$ given $d_{nc} = 20$. This cost makes the SPIR algorithm impractical for $n = 53$.

If we try to tackle the supremacy circuit by the Sum-over-Clifford method, we have a rank-2 Clifford decomposition for the $\text{fSim}(\pi/2, \pi/6)$ gate and a rank-4 decomposition for the $\sqrt{W}$ gate, which are shown in Appendix C. As we discussed in the introduction, there are 430 $\text{fSim}(\pi/2, \pi/6)$ gates. This alone gives an overhead of 2^{430}, which is bigger than the exponential cost of our SPIR method. This indicates that the supremacy circuits are in a regime where the SPIR method scales better than the exact Sum-over-Clifford method. However, it is still impractical to simulate the supremacy circuits for both methods.

The original recursive Feynman path-integral approach in [AC17] gave a coefficient on the exponent for the supremacy circuit of $\log_2 2d = \log_2(80) = 6.32$, where d is the total depth instead of the non-Clifford-gate depth. One can see our modified method is slightly worse than that of [AC17]. This is because the non-Clifford-gate depth to total gate depth ratio is one half for the supremacy circuit, which is relatively large.

For our SPC method, the time cost is $O(d_{nc} n^3 \kappa_n^2) = O(d_{nc} n^3 2^{2.5n})$, from the discussion in the previous section. By combining the SPC method with the Feynman-Schrodinger algorithm, we obtain two equal-size sub-circuits with 35 cross-patch entangling gates (see Figure S21 of the supplemental information of [AAB+19]). Therefore the overall time scaling is $O(d_{nc} n^3 2^{35+1.25\times 53}) \sim O(d_{nc} n^3 2^{100})$.

By combing the SPC method with Sum-over-Clifford, we can save one cycle

of computation time. Because one cycle gives the exact Sum-over-Clifford method an overhead of $2^2 2 \cdot 4^{18} = 2^{58}$, two such cycles gives a factor of $2^{114} \sim 2^{2.5\times 53}$, which is the exponential overhead for one cycle in our SPC method. Again, as we remarked before, because the non-Clifford gates are relatively dense in the supremacy circuit, the benefit of combining the Sum-over-Clifford method with the SPC is limited. These two methods are prohibitive in the supremacy regime because the space requirement also scales as $O(2^{2.5n})$. This motivates for further research of approximate simulation with the SPC method which suits weak simulation.

Chapter 4

Improving teleportation-based simulation by stabilizer-based method on sideways gates

Tensor network simulation [MS08, BISN17, CZH+18, HZN+20] enables one to calculate the amplitude $\langle x| U |0^n\rangle$ for a quantum circuit U by mapping the circuit to a tensor network and contracting the network, as we discussed in Section 1.4.4 of the introduction. The direct simulation of quantum circuits can be regarded as a contraction along the gate sequence direction. However, one could also choose to contract in the perpendicular direction across qubits, i.e., turn the gates sideways as we will elaborate later. By doing so, one obtains a storage cost of $O(2^d)$ and time complexity of $O(n2^d)$ for 1D circuits, which is known to be efficiently simulable [FNW92, Vid04, Has07, ECP10]. For shallow 1D circuits where $d \ll n$, the cost is greatly reduced by turning circuit sideways. Even for 2D dynamics, this idea could still be taken advantage of to reduce the simulation cost, as already explored in [CLG+20] from teleportation point of view, and [NLPD+19] from unitary-projection point of view.

What we mean by "turning a gate sideways" is mixing the input and output labels of a unitary matrix, which is similar to a partial transpose operation. One can

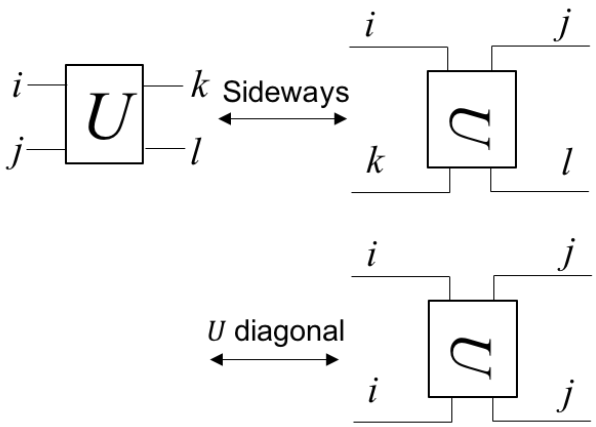

Figure 4.1: Turning a general two-qubit gate and a diagonal two-qubit gate sideways. One can see that turing a diagonal two-qubit gate sideways is equivalent to a single-qubit matrix with one input index and one output index.

see from the diagram in Fig. 4.1 how turning a two-qubit diagonal gate sideways is equivalent to a general single-qubit matrix. The transition amplitudes that define the original U are:

$$\langle kl| U |ij\rangle . \tag{4.1}$$

The sideways gate $\widetilde{U}$ has input indices i and k and outputs j and l. Therefore $\tilde{U}$ should satisfy

$$\langle jl| \widetilde{U} |ik\rangle = \langle kl| U |ij\rangle . \tag{4.2}$$

When U is diagonal, we have $k = i$ and $l = j$ in order of the matrix elements to be non-zero. In this case, the equation above becomes

$$\langle jj| \widetilde{U} |ii\rangle = \langle ij| U |ij\rangle . \tag{4.3}$$

As we can see $\widetilde{U}$ only has one effective input index and one effective output index now and is reduced to a single-qubit gate. By matching the indices for corresponding elements in U and $\widetilde{U}$, we obtain the following: a two-qubit diagonal gate

$$U = \mathrm{diag}\{u_{11}, u_{12}, u_{21}, u_{22}\}. \tag{4.4}$$

is turned into a single-qubit gate [CLG+20]

$$\widetilde{U} = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}, \tag{4.5}$$

and vice versa. In particular, if we let $u_{11} = u_{12} = u_{21} = 1$ and $u_{22} = -1$, one can see if the two-qubit gates are CZ gates, which are Clifford gates, they will become proportional to a Hadamard gate after the transformation.

For the input and output and the circuits, the indices are fixed, see the first qubit in Figure 4.2(a). In this case, only one column (for the input) or one row (for the output) of the single-qubit gates are involved because b, d, g, h will give zero contribution to this amplitude whatever values they take.

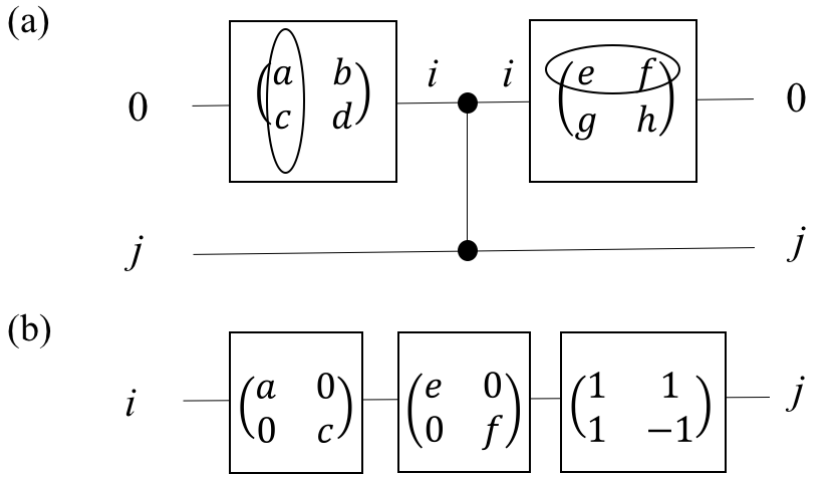

Figure 4.2: Turning gates sideways at the input and output of the original circuit where indices are fixed. (a) The original circuit where input and output indices of the first qubit are all fixed to be 0. Using standard matrix multiplication rules, the matrix elements in the two single-qubit gates that are relevant to this amplitude are circled. (b) Sideways circuit with i being the input index and j being the output index.

This method works naturally for the stabilizer projector algorithms we introduced in Chapter III. The stabilizer projector rank is at least 2^n for an n-qubit unitary. Turning the circuit sideways reduces the stabilizer projector rank from $O(2^n)$ to $O(2^d)$ for 1D shallow circuits. Then one may be able to store the whole "wave function" for the sideways circuit. Therefore one will be able to simulate the cir-

cuit by using the SPC algorithm. For 2D architecture, the sideways circuit will have more than constant number of qubits even when d is constant [CLG+20, NLPD+19], but the reduction of cost by stabilizer projector decomposition of non-Clifford gates is still valid. Therefore, the discussion in the rest of the chapter will not be limited to a specific architecture.

In this chapter we will mainly focus on the circuit ensemble in [BIS+18], which contains $\sqrt{X}$ and $\sqrt{Y}$ gates in eq. (3.2) and eq. (3.3), as well as the T gate:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}, \tag{4.6}$$

and the control-Z gate:

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \tag{4.7}$$

as non-Clifford gates, instead of the gates $\sqrt{W}$ and fSim$(\pi/2, \pi/6)$ in the Google supremacy experiment [AAB+19].

4.1 Stabilizer projector algorithm for sideways circuits

The sideways gates are not generally unitary anymore but our stabilizer projector decomposition is not limited to unitary matrices. In fact, one advantage this provides is that the stabilizer projector rank may be fewer than 2^n. We would like to take advantage of this fact to keep the stabilizer rank throughout the circuit as small as possible. Therefore, by finding gate sets in the original circuits such that after being turned sideways they are not full rank anymore, one can potentially reduce the cost of simulation.

In order to take advantage of the Gottesman-Knill theorem as we apply the SPC algorithm [HL21] to the sideways circuit, we need to analyse the effect of turning sideways on Clifford gates. In fact, some of the sideways Clifford gates

won't be Clifford anymore. We give examples in Section 4.1.1 for the sideways transformation and discuss the circumstances where the sideways gate could be Clifford. It would be great if there are some non-Clifford gates becoming Cliffords after turning sideways, which will potentially reduce the number of Clifford gates or even Clifford depth for some circuits, but we haven't found any such cases yet.

4.1.1 Clifford properties of sideways gates

If the single-qubit gates in Figure 4.2 are $\sqrt{X}$ in eq. (3.2) or $\sqrt{Y}$ (or Hadamard) in eq. (3.3), the matrix after transformation for the two columns of these gates are proportional to phase gate, the inverse of phase gate, identity and Z gate respectively, similar for the rows. If the $\sqrt{X}$ and $\sqrt{Y}$ are between two layers of entangling gates, they will become two-qubit Clifford gates, as we can see by reversing the order in Figure 4.1. According to the equivalence transformation between eq. (4.4) and eq. (4.5), sideways $\sqrt{X}$ gate will be turned into

$$\widetilde{\sqrt{X}} \propto \mathrm{diag}\{1, -i, -i, 1\}, \tag{4.8}$$

which is a Clifford gate, and sideways $\sqrt{X}$ gate will be turned into

$$\widetilde{\sqrt{Y}} \propto \mathrm{diag}\{1, -1, 1, 1\}, \tag{4.9}$$

which is also a Clifford gate. One can check that $\widetilde{\sqrt{X}} = S_1^\dagger \cdot S_2^\dagger \cdot CZ$ and $\widetilde{\sqrt{Y}}$ is equivalent to a CZ gate. In general, we have the following lemma:

Lemma 4.1.1. *Any single-qubit Clifford gate, whose all four matrix elements are non-zero, being turned sideways will still be proportional to a Clifford gate.*

Proof. Assuming the Clifford gate being

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}, \tag{4.10}$$

where $a, b, c, d \neq 0$. The relative phase between a, c or b, d can only be multiples

of i, otherwise this gate acting on $|0\rangle$ or $|1\rangle$ will output non-stabilizer states, which means it isn't a Clifford gate. Using the same argument, the absolute values of a and c have to be the same, so do b and d. Further, with normalization condition, all four elements have to have the same absolute value, i.e., $\frac{1}{\sqrt{2}}$. Meanwhile, when three of the four elements are real or imaginary, then the fourth one also has to be real (or imaginary).

Now we discuss two cases. The first one is when all four elements are real or imaginary. The only possible case is that three elements are 1 s and one is -1 up to an overall phase, e.g., the Hadamard gate and the $\sqrt{Y}$ gate. This being turned sideways will be variations of the CZ gate, which are all Cliffords.

The second case is when the elements in one column are real and those of the other column are imaginary. The only possible case here is

$$\begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix} \tag{4.11}$$

up to an overall phase or switch of the columns. This gate turned sideways will be $S_1 \cdot CZ$.

The last case is when $a = 1$ and $c = \pm i$. The gate then becomes

$$\begin{pmatrix} 1 & b \\ \pm i & \mp ib \end{pmatrix} \tag{4.12}$$

where $b = \pm 1$ or $b = \pm i$. These turned sideways is $CZ \cdot B_1 \cdot S_2$, where $B = I$, Z, S or $S^\dagger$ depending on the value of b. These gates are all Cliffords.

□

However, any single-qubit gate, which is diagonal or anti-diagonal, on turning sideways is not full rank anymore and therefore not unitary. In fact, matrix

$$\begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix} \tag{4.13}$$

turned sideways will become $\mathrm{diag}\{a, 0, 0, d\}$ and

$$\begin{pmatrix} 0 & b \\ c & 0 \end{pmatrix} \tag{4.14}$$

turned sideways will become $\mathrm{diag}\{0, b, c, 0\}$.

4.1.2 Time scaling based on stabilizer projector rank for non-unitary sideways gates

Now we consider certain gate sets which compose random circuits and examine the amount of non-Clifford gates after we turn them sideways. For gate set $\{\sqrt{X}, \sqrt{Y}, T, CZ\}$, which is the circuit ensemble in [BIS+18], if we turn this circuit ensemble sideways, $\sqrt{X}, \sqrt{Y}, CZ$ will still be proportional to Clifford gates, as we showed in the last section. The only non-Clifford (and non-unitary) operations will be the sideways T gates:

$$\widetilde{T} = \mathrm{diag}\{1, 0, 0, e^{i\pi/4}\}, \tag{4.15}$$

Because this is not full rank (only rank 2 instead of rank 4), our algorithm can take advantage of this to reduce the stabilizer projector rank of its decomposition. In fact, as we will see in Section 4.3 where we give examples of exactly what the sideways circuits of the supremacy-type circuits look like, the exact number of qubits for the sideways circuit varies when one contracts the circuits in different ways. We now assume the number of qubits of the sideways circuit being $\tilde{n}$, which is $O(d)$ for 1D original circuit and $O(d\sqrt{n})$ for $\sqrt{n} \times \sqrt{n}$ grid [NLPD+19, CLG+20]. Assuming each layer of the sideways circuit has at least $\tilde{n}_t$ T gates, then the stabilizer projector rank is $2^{\tilde{n}_t/2+(\tilde{n}-\tilde{n}_t)} = 2^{\tilde{n}-\tilde{n}_t/2}$.

We use the SPC algorithm [HL21] for the sideways circuit. One divides the sideways circuit like Figure 3.1 and store the whole "wave function" to run a direct simulation. The memory requirement is reduced when d is constant. The state is stored as a linear combination of stabilizer states. Each Clifford layer is handled by the Gottesman-Knill theorem. For each non-Clifford layer, one multiplies the linear

combination of stabilizer states from the previous cycle to the sum of stabilizer projectors that represent the layer of the gates by taking inner-product between stabilizer states, which can be calculated efficiently in $O(\tilde{n}^3)$ time [GMC12].

The number of states one needs to keep track of after the projection is the number of projectors for the sideways layer of gates. One iterates this till the end of the circuit. If we define the non-Clifford depth of the sideways circuit as $\tilde{d}_{nc}$, the algorithm in the worst case has scaling $O(\tilde{d}_{nc}\tilde{n}^4 2^{(\tilde{n}-\tilde{n}_t/2)})$ to obtain one amplitude $\langle x| U |0^n\rangle$. The reason this doesn't require $2^{2(\tilde{n}-\tilde{n}_t/2)}$ is because the phases of an orthonormal set of stabilizer states have patterns that enables one to use a fast Fourier transform to update the phases.

Assuming one has a linear combination of (computational-basis) stabilizer states $|\psi\rangle = \sum_{j=1} c_j \left|y^{(j)}\right\rangle$ acted on by a Clifford circuit C and then followed by a non-Clifford layer written as a sum of (computational-basis) stabilizer projectors $\sum_j a_j \left|x^{(j)}\right\rangle \left\langle x^{(j)}\right|$, the resulting state will be

$$\begin{aligned}\sum_i a_i \left|x^{(i)}\right\rangle \left\langle x^{(j)}\right| C |\psi\rangle &= \sum_i a_i \left|x^{(i)}\right\rangle \left\langle x^{(i)}\right| C \sum_{j=1} c_j \left|y^{(j)}\right\rangle \\ &= \sum_i \left(\sum_{j=1} c_j \left\langle x^{(i)}\right| C \left|y^{(j)}\right\rangle\right) a_i \left|x^{(i)}\right\rangle\end{aligned} \tag{4.16}$$

Here to avoid confusion later, we use $x^{(i)}$ and $y^{(j)}$ to denote different n-digit binary strings for computational basis vectors. Therefore the new amplitudes will be given by the sum in the parenthesis. When the Clifford circuit C contains $O(n)$ number of basis rotations, e.g. Hadamard gates, which takes the states in $\sum_{j=1} c_j \left|y^{(j)}\right\rangle$ out of the the computational basis, updating all the new amplitude will require naively $O(2^{2n})$ time. However, this scaling can be improved according to the following lemma:

Lemma 4.1.2. *Obtaining the 2^{2n} inner products between the computational basis states and any orthonormal set of stabilizer states basis takes $O(n2^n)$ time.*

Proof. As we discussed in Section 1.2.1, an n-qubit stabilizer states has a canonical

form [DDM03, Nes08]:

$$|\phi\rangle = \frac{1}{2^{k/2}} \sum_{u \in Z_2^k} i^{l(v)} (-1)^{q(v)} |v = Ru + t\rangle \tag{4.17}$$

The set of stabilizer states $C\left|y^{(1)}\right\rangle, ..., C\left|y^{(N)}\right\rangle$ all have this form with different $l(y)$, $q(y)$ and t, where $N = 2^n$. In fact, this set forms a orthonormal basis just as $\{\left|y^{(1)}\right\rangle, ..., \left|y^{(N)}\right\rangle\}$ does, therefore, the quadratic terms in $q(y)$ has to be the same for all N states. When these states are acted on by a operator with a computational-basis projector decomposition, as in eq. (4.16), the sum within the parenthesis in the last expression in eq. (4.16) becomes:

$$\begin{aligned} \sum_{j=1} c_j \left\langle x^{(i)} \right| C \left| y^{(j)} \right\rangle &= \sum_{j=1} c_j i^{l_j(x^{(i)})} (-1)^{q_j(x^{(i)})} \\ &= (-1)^{\sum_{kl} q_{kl} x_k^{(i)} x_l^{(i)}} \sum_{j=1} c_j i^{l_j(x^{(i)})} (-1)^{\sum_k x_k^{(i)} q_{j,k}} \end{aligned} \tag{4.18}$$

where the functions $l_j(y)$s differ for different subscripts j, and $q_j(y)$s only differ for the linear terms. $q_{j,k}$ is the linear term in the quadratic function where the first subscript indicates different functions for $C\left|y^{(1)}\right\rangle, ..., C\left|y^{(N)}\right\rangle$, and the second subscript represents the coefficients with respect to x. This sum has the same form as the discrete Fourier transform, and therefore sums for all x_is can be calculated in $O(N \log N)$ time using the fast Fourier transform algorithm instead of $O(N^2)$ time. □

A notable feature of our algorithms is that the time scaling of the simulation goes down if the circuit is dominated by Clifford gates or if the circuit has many T gates (diagonal non-Clifford gates in general). This is not surprising because we take advantage of the monotone methods here, which is faster with the increase of the number of diagonal gates, and T gate is one of them. While the original teleportation-based algorithm [CLG+20] doesn't handle Clifford gates by the Gottesman-Knill theorem and the layers of the sideways circuits are generally full rank, our algorithm improves the time scaling by a factor of $2^{\tilde{n}_t/2}$ in general,

with another constant multiplicative factor speedup.

4.2 Further reduction of stabilizer projector rank

We have the following observation of rank reduction for particular sequence of gates:

Observation 4.2.1. One notices that two overlapping sideways T gates, see Figure 4.3, have stabilizer projector rank 2 instead of 4:

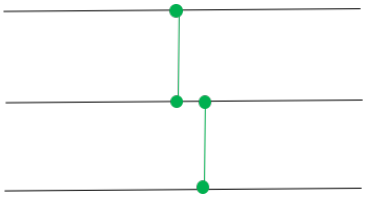

Figure 4.3: Two overlapping sideways T gates. Here we use the same symmetric gate symbol as the CZ gate but colored in green to represent the sideways T gate.

$$\begin{aligned}\widetilde{T}_{1,2} \otimes I \cdot I \otimes \widetilde{T}_{2,3} = &(|000\rangle\langle 000| + e^{i\pi/4} |110\rangle \langle 110| \\ &+ |001\rangle \langle 001| + e^{i\pi/4} |111\rangle \langle 111|) \\ &\cdot (|000\rangle\langle 000| + e^{i\pi/4} |011\rangle \langle 011| \\ &+ |100\rangle\langle 100| + e^{i\pi/4} |111\rangle \langle 111|) \\ &= |000\rangle\langle 000| + i |111\rangle \langle 111| .\end{aligned} \tag{4.19}$$

One can have further reduction with more overlapping sideways T gates on j qubits because only the $\left|0^{\otimes j}\right\rangle \left\langle 0^{\otimes j}\right|$ and $\left|1^{\otimes j}\right\rangle \left\langle 1^{\otimes j}\right|$ terms survive. However, this reduction replies heavily on a particular gate sequence.

For more general reduction, we define the dyadic stabilizer projector rank:

$$\sigma(\hat{O}) = \min\{k : \hat{O} = \sum_{j=1}^{k} c_j |\phi_j\rangle \langle s_j| .\} \tag{4.20}$$

where $|\phi_j\rangle$s and $|s_j\rangle$s are two sets of stabilizer states. The 1-norm version of the dyadic decomposition is defined and discussed in [SRP+20]. Inserting dyadic stabilizer projector decomposition has the same effect of preserving the number of terms.

Now we will discuss some of the gates where one can use this notion to reduce the rank.

4.2.1 Reduction for single-qubit diagonal Clifford gates

A Hadamard gate turned sideways is proportional to a CZ gate, therefore is still Clifford. There are Clifford gates which are not Clifford anymore (even not unitary) when turned sideways. For example, the Phase gate:

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{4.21}$$

If we turn this gate sideways as for T gates in the last paragraph, it becomes

$$\widetilde{S} = \mathrm{diag}\{1, 0, 0, i\}, \tag{4.22}$$

which unlike $\sqrt{X}, \sqrt{Y}$, doesn't turn into a two-qubit Clifford gate. $\tilde{S}$ has stabilizer projector rank 2. However, we could reduce this rank further for this case by inserting an extra identity and turn the two-qubit gate $S \otimes I$ sideways. This gate is diagonal and the sideways single-qubit gate becomes

$$\widetilde{S \otimes I} = \begin{pmatrix} 1 & i \\ 1 & i \end{pmatrix} \propto |+\rangle \langle +i|. \tag{4.23}$$

Here we write this "gate" as a rank-1 dyadic version of stabilizer states. By inserting this dyadic term, the stabilizer rank of the state obtained is reduced to 1 after gate $S \otimes I$. The same procedure works for the Z gate. Though these procedure come at the price of potentially increasing the number of sideways qubits, they may still effectively reduce the rank as a result.

4.2.2 Reduction for two-qubit non-diagonal gates

So far we have only analyzed diagonal two-qubit gate turned sideways which become single-qubit gates. What about non-diagonal two-qubit gates? One can assume the two-qubit gate has the form

$$U = \sum_{m,n=0,1} |u_{mn}\rangle \langle mn| \tag{4.24}$$

where $|u_{mn}\rangle$ is the new basis that this unitary turns the computational basis into. Then based on eq. (4.2), one obtains

$$\langle jl| \widetilde{U} |ik\rangle = \langle kl| \sum_{m,n=0,1} |u_{mn}\rangle \langle mn|ij\rangle = \langle kl|u_{ij}\rangle \tag{4.25}$$

For $\widetilde{U}$ to be not full rank, one needs this amplitude to be zero for some assignments of i, k.

For the $CNOT$ gate,

$$|u_{mn}\rangle = |m, m \oplus n\rangle . \tag{4.26}$$

Eq. (4.25) becomes

$$\langle jl| \widetilde{U} |ik\rangle = \langle kl|i, i \oplus j\rangle = \delta_{ik}\delta_{i\oplus j,l} \tag{4.27}$$

The left hand side vanishes when $i \neq k$, and therefore $\widetilde{CNOT}$ has rank 2. For the $SWAP$ gate,

$$|u_{mn}\rangle = |mn\rangle \tag{4.28}$$

Eq. (4.25) becomes

$$\langle jl| \widetilde{U} |ik\rangle = \langle kl|ji\rangle = \delta_{il}\delta_{jk} \tag{4.29}$$

Therefore for all values of i, k, one can always find the corresponding j, l such that the above amplitude is not zero. Therefore, $SWAP$ gate turned sideways will still be full rank, as is the $iSWAP$ gate.

In fact, the $CNOT$ gate turned sideways becomes:

$$\widetilde{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \tag{4.30}$$

This can be written as a dyadic rank-2 decomposition:

$$\widetilde{CNOT} \propto |00\rangle \left\langle \Phi^+ \right| + |11\rangle \left\langle \Psi^+ \right| \tag{4.31}$$

The techniques discussed in this section apply to wherever these gates show up. For the SPC algorithm of sideways circuit, the cost scales with the largest rank of the non-Clifford layers. Therefore, these techniques need to be used depending on the structure of the sideways circuit. In the next section, we will show that there are different ways of turning circuit sideways. Given a circuit structure, one should choose among these different ways such that when combined with the rank reduction techniques, one results in the smallest cost.

4.3 Sideways simulation for different gate sequences: parallel vs. sequential

In this section, we investigate what the sideways circuits actually look like for the supremacy-type circuit with gate set $\{\sqrt{X}, \sqrt{Y}, T, CZ\}$. We will discuss the different structures of the sideways circuits when we assign the sideways qubits differently.

Using the teleportation prospective in [CLG$^+$20], we give an example in Figure 4.4 how the change of the input and output indices to the sideways gates can be simplified by the direction of the arrow, which will make our following discussion of different ways of turning sideways much easier. Following the direction of the arrow, the indices the arrow passes through first will be the input indices for every gate. The number of input indices always equals to the number of output indices.

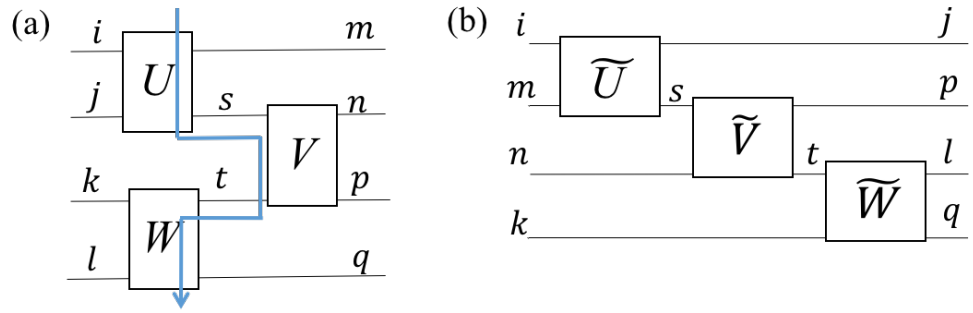

Figure 4.4: For the original circuit in (a), one can follow the the arrow, which goes through the top qubit first, therefore indices i, m are the input indices of the sideways circuit. Then it touches indices j, s, which should be the outputs of the sideways U. Following the arrow, s goes on to be the input of sideway V, so does n. Following this analysis, we obtain the sideways circuit with indices as shown in (b).

The amplitude represented in the diagram is:

$$\langle ijkl|\,\mathcal{C}\,|mnpq\rangle = \sum_{s,t} \langle np|\,V\,|st\rangle\,\langle tq|\,W\,|kl\rangle\,\langle ms|\,U\,|ij\rangle \tag{4.32}$$

Using eq. (4.2) which defines the sideways gates, we can transform this expression into

$$\begin{aligned} &\sum_{s,t} \langle tp|\,\widetilde{V}\,|sn\rangle\,\langle lq|\,\widetilde{W}\,|kt\rangle\,\langle js|\,\widetilde{U}\,|im\rangle \\ =&\sum_{s,t} \langle lq|\,\widetilde{W}\,|kt\rangle\,\langle tp|\,\widetilde{V}\,|sn\rangle\,\langle js|\,\widetilde{U}\,|im\rangle \end{aligned} \tag{4.33}$$

Now we see the input of $\widetilde{W}$ has become the output of $\widetilde{V}$, which results in the sideways circuit Figure 4.4(b). When U, V, W are diagonal gates, the arrow then represents the flow of a single sideways qubit, because $i = m$, $s = n$ and $t = k$.

As we can see in Figure 4.5, if one turn circuits with cycles of parallel gate sequences (the supremacy circuit for example) sideways, one obtains a circuit with a pattern of sequential gates. Because one wants as many non-full rank gates in each layer as possible, one decomposes the original circuit into layers such that each layer has gates acting non-trivially on as many qubits as possible. However, for the sequential circuit in figure 4.5(b), one can see a full layer of gates has depth $O(\tilde{n})$.

This is not suitable for easy layered decomposition into sum of stabilizer projectors. Are there other ways in which sideways qubits can flow that changes the sequential pattern of the sideways circuit?

In fact, one can choose the flow of the sideways qubits not only through space, but also time, see Figure 4.6. In this way, one not only reduces the depth of the sideways circuit but also is able to maintain the feature of cycles of parallel gates. Doing this might add the number of qubits of the sideways circuit, but in return we obtain a circuit whose non-Clifford layers contain as many non-full-rank sideways gates as possible and can be easily decomposed into a sum of stabilizer projectors, as well as using the dyadic decomposition in Section 4.2. In other words, when choosing sideways qubits to flow both horizontally and perpendicularly, one increases $\tilde{n}$ as well as $\tilde{n_t}$. We have a trade-off between the number of qubits in the sideways circuit and the number of non-full-rank gates in each layer of the sideways circuits. Therefore, given specific circuits, one wants to choose the flow of the sideways qubits such that $\tilde{n} - \tilde{n_t}$ is maximized. The optimal trade-off may in general be hard to find exactly, just as it is NP-hard to find the treewidth for the tensor network simulation [ACP87]. However, our method has the advantage that one can simulate the full-rank Clifford layers with the Gottesman-Knill theorem, only the maximal stabilizer projector (or dyadic) decomposition of the non-Clifford layers contributes to the exponential cost.

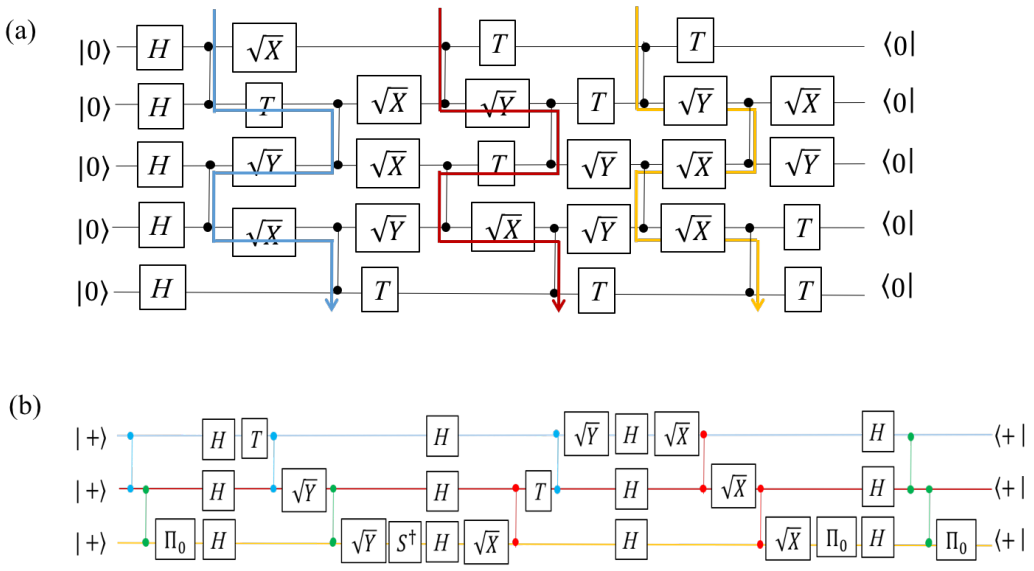

Figure 4.5: (a) A 5-qubit supremacy-type circuit that has 5 cycles (depth 11). The arrows of different dark colors represent the qubit-flow of its sideways circuit. Here the two-qubit gates are all Control-Z gates. (b) The sideways circuit of (a) which has depth 23. Here the two-qubit gates of bright blue, red, green colors corresponds to $\sqrt{X}$, $\sqrt{Y}$ and T gates respectively. Notice that we omit the normalization factor for the Hadamard gate (control-Z turned sideways). A layer in this circuit is sequential in a way that the two-qubit gates are commingled with single-qubit gates, while the original circuit has cycles that is composed of parallel gates.

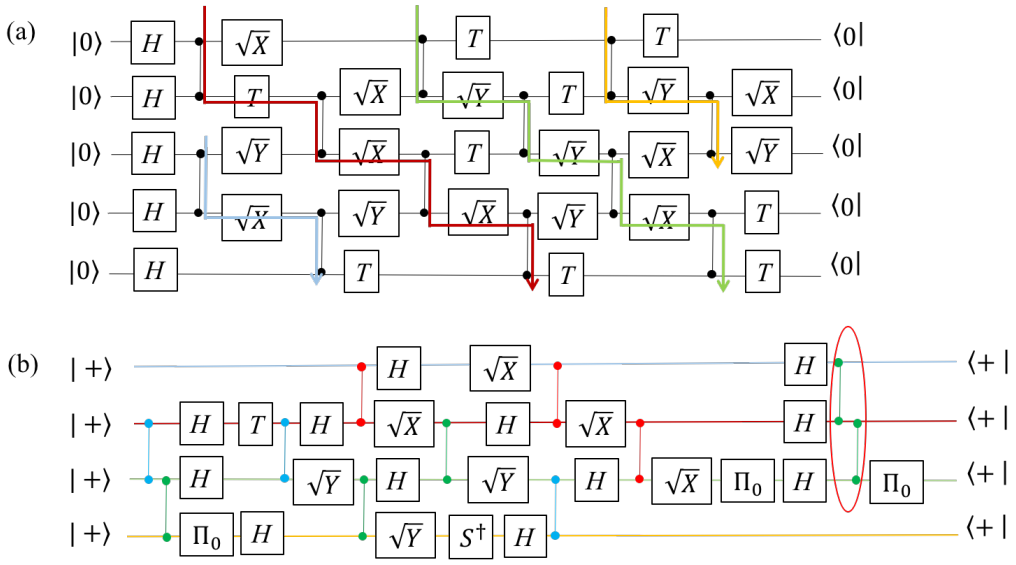

Figure 4.6: (a) The same 5-qubit supremacy-type circuit that has 5 cycles (depth 11). The arrows of different dark colors represents a different sideways qubit-flow from figure 4.6. The sideways qubits don't just run across space (qubits in the original circuit) but also time. (b) The sideways circuit of (a). Compared to 4.6, the sideways circuit here still has the parallel feature of the original circuit. The depth is reduced to 16.

Chapter 5

Conclusions and open problems

5.1 Summary of contributions

In this thesis we explored several direction of stabilizer-based simulation, more specifically the extension to qudits and algorithms that incorporates the advantage of other classes of simulation algorithms and the kinds of circuits that will benefit from these algorithms. Here we summarize our contribution in the following sections.

5.1.1 Qudit Bravyi-Gosset weak simulation

We generalized the notion of approximate stabilizer rank to the case of odd prime qudits. We partitioned the set of stabilizer states into d sets of orbits around a given qudit magic state. There are $d+1$ states in each orbit and they all have equal inner product with the magic state. We showed that all of these orbits can be used to form a non-orthogonal basis of the magic state, therefore we have $d+1$ options to write down an exact stabilizer decomposition of a single-qudit magic state.

Using n copies of these orbits, one can randomly select a subspace of these basis stabilizer states and and use them to approximate the magic state. Analogous to the qubit case, we show that using the orbit that has the largest overlap with the magic state, one can use the fewest stabilizer states to achieve a desired accuracy of approximation. In other words, the larger the stabilizer fidelity of a qudit magic

state is, the smaller the approximate stabilizer rank is. In fact, we show that these two quantities are inversely proportional to each other. We also give the stabilizer fidelity for several commonly used qudit magic states for $d = 3, 5, 7$.

5.1.2 Boosting recursive Feynman path-integral simulation by stabilizer projector decomposition

To explore in the possible connection of stabilizer-based simulation and monotone methods, we identified the kind of monotone simulation that would benefit from treating Clifford and non-Clifford gates separately and use of Gottesman-Knill theorem. By introducing the stabilizer projector decomposition of unitaries, we decomposed the non-Clifford gates into stabilizer projectors and incorporated that into the recursive Feynman path-integral simulation. The algorithm we propose, called the SPIR method, improves the previous known bound for polynomial-space simulation of quantum circuits from $O((2d)^n)$ to $O((2d_{nc})^n)$ for calculating a single amplitude. At the same time, we laid out an exponential-space simulation algorithm based on the stabilizer projector decomposition, called SPC algorithm, which is developed further in our next work.

We compared the performance of our method to the Bravyi-Gosset algorithm in the context of an ensemble of random circuits with a parametrized probability of non-Clifford gates. As expected, the Bravyi-Gosset algorithm is better for the regime of $t < n$, while our algorithm performs better in the regime of a higher density of non-Clifford gates. We also analyze the time required for simulating the supremacy circuit. We remark despite the fact that the scaling of our algorithm has not been able to challenge state-of-the-art tensor network algorithms, it provides a better polynomial-space simulation and builds a bridge between stabilizer-based algorithms and monotone methods. This connection sheds light on some future algorithms that combines the advantages of the two and performs the best for some certain class of circuits.

5.1.3 Taking advantage of sideways gates in stabilizer projector simulation

Following the direction of combining stabilizer-based algorithms with monotone methods, we explored approaches that rewrite the output amplitude of a circuit with non-unitary operations. In Chapter IV, we invoked the spirit of different orders for tensor network contraction of a circuit and the teleportation-inspired algorithm to turn shallow circuits sideways. These sideways gates are not necessarily unitary (and therefore full-rank) anymore, which makes this approach a better candidate for using the SPC algorithm from Chapter III.

We discussed the Clifford properties of Clifford gates turned sideways and showed that all single-qubit Clifford gates turned sideways are proportional to Clifford gates as long as the four entries are all non-zero. We also showed that typical two-qubit Clifford gate CZ turned sideways is also proportional to a Clifford gate. Looking at the non-Clifford gate T turned sideways, we pointed out that it won't be full-rank anymore and therefore we obtained an algorithm that scales with the biggest stabilizer projector rank of the non-Clifford layers of the sideways circuit. We showed a interesting fact that the more T gates there are in the circuit, the more likely our simulation will turn out faster. In fact, the algorithm in the worst case has a time cost that is exponential in $O(2^{(\tilde{n}-\tilde{n}_t/2)})$ to obtain one amplitude $\langle x| U |0^n\rangle$, where $\tilde{n}$ is the number of qubits in the sideways circuit and $\tilde{n}_t$ is the minimal number of sideways T gates (or any sideways diagonal non-Clifford gates) in the non-Clifford layer of the sideways circuit. Using a dyadic stabilizer decomposition of sideways gates, we demonstrated ways to write down a sparser stabilizer projector decomposition.

We investigated the possibility of different ways of turning the circuit sideways and pointed out that there is a trade-off between the number of qubits in the sideways circuit and the number of non-full-rank gates in each layer of the sideways circuits.

5.2 Future directions

5.2.1 Obtaining sparse decomposition by l_1 minimization

The number of stabilizer states is $O(2^{O(n^2)})$ [AG04], hence the task of finding the stabilizer rank or the SPM rank becomes intractable very quickly. Finding a sparsest solution within an overcomplete basis set is a non-convex optimization problem and NP-hard [Nat95]. The l_1 minimization in the definition of RoM (1.33) is a convex optimization problem.

What kind of stabilizer pseudomixture decomposition gives the minimum l_1 norm, i.e., the robustness of magic? What kind of stabilizer pseudomixture decomposition gives the smallest l_0 norm, i.e., SPM rank? Are there any states where these two cases coincide?

To address these questions, can we replace l_0 optimization of rank by convex l_1 optimization? This kind of question has previously been considered in the compressed sensing literature [DE03, GN03]. For states where the minimum decomposition of the two norms coincide, l_1 optimization gives the exact SPM rank. All methods of finding sparser stabilizer decompositions rely on finding a sparse decomposition for fewer than 8-qubit magic states and then using the tensor product of these qubit blocks to obtain decomposition for general n-qubit magic states [BSS16, BBC$^+$19]. Because l_1 minimization is computationally more tractable than l_0 minimization, one can trade off the requirement of finding the decomposition that gives the exact rank to find sparse decomposition for bigger blocks of qubits by l_1 minimization. By having tensor product of sparse decomposition results for bigger blocks, one maybe able to obtain sparser decomposition for general n-qubit states.

To explore this trade-off, one can define new ranks over subsets of the stabilizer states in which the maximum magnitude of the inner product between any pair of distinct states is M. The smaller the subset is, the smaller M is, the denser the best representation is but the easier the rank determination problem is. In fact, it is already explored in [DE03] that the smaller M is, the better chance there is

to obtain the rank with l_1 minimization techniques. For the whole stabilizer states, $M = 2^{-1/2}$, and we don't expect l_1 minimization can obtain the SPM rank over all of the stabilizer states. At the other extreme we have a subset similar to the Mutual Unbiased Basis (MUB) rank in which $M = 2^{n/2}$ [Woo87] and we hope that l_1 minimization can generate a good enough stabilizer subset rank somewhere in the middle. Limiting M greatly reduce the number of stabilizer states in the subset, for example, the total number of n-qubit stabilizer states scales with $2^{O(n^2)}$ and the MUB set has $O(2^{2n})$ stabilizer states. By reducing the number of stabilizer states in the subset we choose, we can deal with stabilizer subset rank with l_1 minimization for bigger blocks of qubits. Our hope is that the decompositions obtained by this method can outperform the decomposition obtained by doing l_0 minimization on full set of stabilizer states of smaller blocks.

5.2.2 Approximate simulation with the stabilizer projector methods

NISQ devices suffer from noisy operations. Therefore the output distributions of quantum circuits run on NISQ devices don't really follow the ideal distribution one calculates assuming perfect gates. Calculating the exact probability distribution is of interest for benchmarking NISQ devices and for exploring the complexity separation between the power of classical and fault-tolerant quantum devices. On the other hand, it suffices for classical simulation to refute quantum supremacy claims if one can come up with efficient classical algorithms that mimic the behavior of shallow noisy quantum circuits. In another word, as long as we have an approximate classical simulation that output samples that can spoof the verifier of its true identity, we should consider it a successful simulation.

Therefore, it is worth exploring how our stabilizer projector algorithms for recursive Feynman path-integral simulation and sideways gate simulation can be relaxed (or extended) into an approximate simulation. The interesting question will then be, will there be a exponential improvement in the time scaling by relaxing the exact simulation condition, like the $2^{0.47t}$ to $2^{0.23t}$ improvement of the Bravyi-Gosset

algorithm?

The next two decades are crucial for the scaling and commercialization of quantum computing devices. As numerous exciting experimental and engineering breakthroughs that is happening, it is the author's hope that our knowledge of classical simulation will be more interactive with hardware advancements and play an important role in this era, in terms of understanding the power of quantum computing better, benchmarking and certification of devices in the landscape, better compiling of circuits that saves hardware resources and other potential applications.

Appendices

Appendix A

More SPM decompositions of various resource states

The SPM rank-7 decomposition of $|T\rangle\langle T|^{\otimes 2}$ is:

$$\begin{aligned}
|T\rangle\langle T|^{\otimes 2} = &\frac{1}{\sqrt{3}}|00\rangle\langle 00| - \frac{1}{\sqrt{3}}\left|\Phi^-\right\rangle\left\langle\Phi^-\right| \\
&+ \frac{1}{3}\left|\Phi^{+i}\right\rangle\left\langle\Phi^{+i}\right| - \frac{1}{\sqrt{3}}|--\rangle\langle--| \\
&+ \frac{1}{\sqrt{3}}|+i,+i\rangle\langle+i,+i| \\
&+ \frac{1}{3}(\frac{1}{\sqrt{2}}(|+i,+i\rangle + i\,|-i,-i\rangle)) \\
&(\frac{1}{\sqrt{2}}(\langle+i,+i| - i\,\langle-i,-i|)) \\
&+ \frac{1}{3}(\frac{1}{\sqrt{2}}(|++\rangle - i\,|--\rangle))(\frac{1}{\sqrt{2}}(\langle++| + i\,\langle--|))
\end{aligned} \tag{A.1}$$

where $|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$.

The SPM rank-12 decomposition of $|H\rangle\langle H|^{\otimes 3}$ is:

$$
\begin{aligned}
|H\rangle\langle H|^{\otimes 3} = {} & (\frac{3}{4} - \frac{\sqrt{2}}{2}) \, |\langle -XXY, -XYX, -YXX\rangle\rangle \\
& \langle\langle -XXY, -XYX, -YXX\rangle| \\
& + (\frac{1}{4} - \frac{\sqrt{2}}{4}) \\
& |\langle -XXY, -XYX, YXX\rangle\rangle \, \langle\langle -XXY, -XYX, YXX\rangle| \\
& + (\frac{\sqrt{2}}{4} - \frac{1}{2}) \\
& |\langle -IXY, -IYX, XII\rangle\rangle \, \langle\langle -IXY, -IYX, XII\rangle| \\
& + (\frac{1}{4} - \frac{\sqrt{2}}{4}) \\
& |\langle -IZZ, XIZ, -YXX\rangle\rangle \, \langle\langle -IZZ, XIZ, -YXX\rangle| \\
& + (\frac{1}{4} - \frac{\sqrt{2}}{4}) \\
& |\langle -IZZ, -XIZ, YXX\rangle\rangle \, \langle\langle -IZZ, -XIZ, YXX\rangle| \\
& + (\frac{\sqrt{2}}{4} - \frac{1}{2}) \\
& |\langle IXI, -XIX, -YIY\rangle\rangle \, \langle\langle IXI, -XIX, -YIY\rangle| \\
& + (\frac{\sqrt{2}}{4} - \frac{1}{2}) \\
& |\langle IIX, -XYI, -YXI\rangle\rangle \, \langle\langle IIX, -XYI, -YXI\rangle| \\
& + \frac{1}{2} \, |\langle IIX, IXI, XII\rangle\rangle \, \langle\langle IIX, IXI, XII\rangle| \\
& + \frac{\sqrt{2}}{4} \, |\langle IIX, IYI, YII\rangle\rangle \, \langle\langle IIX, IYI, YII\rangle| \\
& + \frac{\sqrt{2}}{4} \, |\langle IIY, IXI, YII\rangle\rangle \, \langle\langle IIY, IXI, YII\rangle| \\
& + \frac{\sqrt{2}}{4} \, |\langle IIY, IYI, XII\rangle\rangle \, \langle\langle IIY, IYI, XII\rangle| \\
& + (\frac{1}{2} - \frac{\sqrt{2}}{4}) \\
& |\langle IYY, -XXX, YIX\rangle\rangle \, \langle\langle IYY, -XXX, YIX\rangle|
\end{aligned}
\tag{A.2}
$$

where we use the stabilizer generator notation $\langle\rangle$ to represent the stabilizer states.

Appendix B

Conventions and definitions for stabilizer states

We can for convenience label the single qubit stabilizer states as follows: $|+\rangle = |x\rangle$, $|-\rangle = |\bar{x}\rangle$, $|+i\rangle = |y\rangle$, $|-i\rangle = |\bar{y}\rangle$, $|0\rangle = |z\rangle$, $|1\rangle = |\bar{z}\rangle$. Given a pair of labels of basis a and b where $a, b \in \{x, y, z\}$ we can define the following states:

$$\begin{aligned}
\left|\Phi^{\pm}_{ab}\right\rangle &= \frac{1}{\sqrt{2}}\left(|ab\rangle \pm \left|\bar{a}\bar{b}\right\rangle\right) \\
\left|\Phi^{\pm i}_{ab}\right\rangle &= \frac{1}{\sqrt{2}}\left(|ab\rangle \pm i\left|\bar{a}\bar{b}\right\rangle\right) \\
\left|\Psi^{\pm}_{ab}\right\rangle &= \frac{1}{\sqrt{2}}\left(\left|a\bar{b}\right\rangle \pm |\bar{a}b\rangle\right) \\
\left|\Psi^{\pm i}_{ab}\right\rangle &= \frac{1}{\sqrt{2}}\left(\left|a\bar{b}\right\rangle \pm i\,|\bar{a}b\rangle\right)
\end{aligned} \tag{B.1}$$

This is the notation for two qubit stabilizer states that we will use below. There are six orthogonal bases of four stabilizer states that are maximally entangled, therefore the notation we use here doesn't correspond one-to-one to the stabilizer states.

Appendix C

Gates in the Google supremacy experiment and their Sum-over-Clifford decompositions

The single-qubit Clifford gates are:

$$\sqrt{X} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \tag{C.1}$$

and

$$\sqrt{Y} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}. \tag{C.2}$$

The non-Clifford single-qubit gate is:

$$\begin{aligned} \sqrt{W} &= T^\dagger \sqrt{X} T \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -\sqrt{i} \\ \sqrt{-i} & 1 \end{pmatrix}. \end{aligned} \tag{C.3}$$

One Sum-over-Clifford decomposition for the T gate is

$$\begin{aligned} T &= \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} = e^{i\pi/8} \begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix} \\ &= \frac{e^{i\pi/8}}{2\cos(\pi/8)}(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + e^{i\pi/4} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}) = \frac{e^{i\pi/8}}{2\cos(\pi/8)}(I + e^{i\pi/4}Z). \end{aligned} \tag{C.4}$$

Therefore we have an upper bound for the rank of the sum-over-Clifford for the $\sqrt{W}$ gate as 4.

The two qubit gate is

$$\text{fSim}(\pi/2, \pi/6) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 0 & e^{-i\pi/6} \end{pmatrix}. \tag{C.5}$$

which is non-Clifford.

This gate has sum-over-Clifford rank 2 because

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-i\pi/6} \end{pmatrix} = \frac{e^{i\pi/12}}{2\cos(\pi/12)}(I + e^{i\pi/6}CZ). \tag{C.6}$$

and

$$\text{fSim}(\pi/2, \pi/6) = iSWAP^\dagger \cdot \text{diag}\{1, 1, 1, e^{i\pi/6}\}. \tag{C.7}$$

Appendix D

Stabilizer projector decomposition for non-Clifford gates in the Google supremacy experiment

Here we give one stabilizer projector decomposition for each gate that gives the upper bounds in Table 3.2. Notice the coefficients are not unique. The optimal support of the stabilizer projectors may also not be unique for the bounds.

$$\begin{aligned}\mathrm{fSim}(\pi/2,\pi/6) = &|zz\rangle\langle zz| + e^{i\pi/6}|\bar{z}\bar{z}\rangle\langle\bar{z}\bar{z}| \\ &- i\left|\Psi^+_{zz}\right\rangle\left\langle\Psi^+_{zz}\right| + i\left|\Psi^-_{zz}\right\rangle\left\langle\Psi^-_{zz}\right|\end{aligned} \tag{D.1}$$

$$\begin{aligned}\sqrt{W_1}\sqrt{W_2} &= \frac{1}{2}\begin{pmatrix} 1 & -\sqrt{i} & -\sqrt{i} & i \\ \sqrt{-i} & 1 & -1 & -\sqrt{i} \\ \sqrt{-i} & -1 & 1 & -\sqrt{i} \\ -i & \sqrt{-i} & \sqrt{-i} & 1, \end{pmatrix} \\ &= -\sqrt{2}i\left|\Phi^-_{zz}\right\rangle\left\langle\Phi^-_{zz}\right| + \left|\Psi^-_{zz}\right\rangle\left\langle\Psi^-_{zz}\right| \\ &+ \frac{i}{\sqrt{2}}\left|\Phi^{+i}_{zz}\right\rangle\left\langle\Phi^{+i}_{zz}\right| + (1+\frac{i}{\sqrt{2}})\left|\Phi^{-i}_{zz}\right\rangle\left\langle\Phi^{-i}_{zz}\right| \\ &- \sqrt{2}i\left|xx\right\rangle\left\langle xx\right| + \sqrt{2}i\left|\bar{y}\bar{y}\right\rangle\left\langle\bar{y}\bar{y}\right|\end{aligned} \tag{D.2}$$

$$\begin{aligned}&\mathrm{fSim}(\pi/2,\pi/6)\cdot\sqrt{W_1}\sqrt{W_2} = (0.134-0.5i)\left|zz\right\rangle\left\langle zz\right| \\ &+ (1.954+1.171i)\left|\Phi^+_{zz}\right\rangle\left\langle\Phi^+_{zz}\right| - (1.644+0.329i)\left|\Psi^+_{zz}\right\rangle\left\langle\Psi^+_{zz}\right| \\ &+ 2i\left|\Psi^-_{zz}\right\rangle\left\langle\Psi^-_{zz}\right| - (1.866+0.5i)\left|\Phi^{+i}_{zz}\right\rangle\left\langle\Phi^{+i}_{zz}\right| \\ &+ (0.156+0.966i)\left|\bar{x}\bar{x}\right\rangle\left\langle\bar{x}\bar{x}\right| \\ &+ (1.903+0.778i)\left|yy\right\rangle\left\langle yy\right| \\ &+ (2.869+2.451i)\left|\bar{y}\bar{y}\right\rangle\left\langle\bar{y}\bar{y}\right| \\ &- (1.673+1.862i)\left|\Psi^+_{yy}\right\rangle\left\langle\Psi^+_{yy}\right| \\ &- (0.966+1.673i)\left|\Psi^{+i}_{xx}\right\rangle\left\langle\Psi^{+i}_{xx}\right|.\end{aligned} \tag{D.3}$$

$$
\begin{aligned}
\mathrm{fSim}(\pi/2,\pi/6)\cdot\sqrt{W_1} &= (1.866+0.5i)\,|zz\rangle\langle zz| \\
&+ (0.413+0.646i)\,|x\bar{z}\rangle\langle x\bar{z}| + (1.319+0.354i)\,|y\bar{z}\rangle\langle y\bar{z}| \\
&+ (1.378+2.319i)\,|\bar{z}\bar{x}\rangle\langle\bar{z}\bar{x}| + (0.354-1.319i)\,|z\bar{y}\rangle\langle z\bar{y}| \\
&- (1.378+2.319i)\left|\Phi^-_{zz}\right\rangle\left\langle\Phi^-_{zz}\right| - (1.220+2.112i)\left|\Psi^+_{zz}\right\rangle\left\langle\Psi^+_{zz}\right| \\
&- (0.159+0.207i)\left|\Psi^-_{zz}\right\rangle\left\langle\Psi^-_{zz}\right| + (-0.036+2.319i)\,|\bar{x}x\rangle\langle\bar{x}x| \\
&+ (1.061+1.319i)\,|\bar{y}\bar{y}\rangle\langle\bar{y}\bar{y}| \\
&- (1.378+0.905i)\left|\Psi^-_{yy}\right\rangle\left\langle\Psi^-_{yy}\right| \\
&- (0.354+0.095i)\left|\Psi^{-i}_{xx}\right\rangle\left\langle\Psi^{-i}_{xx}\right|.
\end{aligned}
\tag{D.4}
$$

$$
\begin{aligned}
\sqrt{W_2}\cdot iSWAP\cdot CZ\cdot\sqrt{W_1} &= 2\,|zz\rangle\langle zz| + 2\sqrt{2}\,|\bar{x}\bar{y}\rangle\langle\bar{x}\bar{y}| \\
&- ((\sqrt{2}+1)+(\sqrt{2}-1)i)\left|\Psi^{-i}_{zz}\right\rangle\left\langle\Psi^{-i}_{zz}\right| \\
&+ ((\sqrt{2}+1)-(3+\sqrt{2})i)\left|\Psi^{+i}_{zz}\right\rangle\left\langle\Psi^{+i}_{zz}\right| \\
&- 2(1+(\sqrt{2}-1)i)\left|\Psi^-_{yy}\right\rangle\left\langle\Psi^-_{yy}\right| \\
&+ 2(1+\sqrt{2}i)\left|\Psi^-_{zz}\right\rangle\left\langle\Psi^-_{zz}\right| + 2(-1+i)\left|\Psi^{-i}_{zx}\right\rangle\left\langle\Psi^{-i}_{zx}\right| \\
&+ \sqrt{2}(-1+i)\left|\Phi^{-i}_{xz}\right\rangle\left\langle\Phi^{-i}_{xz}\right| \\
&+ ((-2-\sqrt{2})+0.586i)\left|\Psi^{-i}_{xz}\right\rangle\left\langle\Psi^{-i}_{xz}\right| \\
&+ 2(1+(\sqrt{2}-1)i)\left|\Psi^{+i}_{xx}\right\rangle\left\langle\Psi^{+i}_{xx}\right|.
\end{aligned}
\tag{D.5}
$$

Appendix E

Stabilizer projector rank for $\sqrt{W} \otimes \sqrt{W}$

Our starting point is the spectral decomposition of $\sqrt{W} \otimes \sqrt{W}$. The spectrum of $\sqrt{W} \otimes \sqrt{W}$ has a doubly-degenerate eigenvalue 1 eigenspace and two eigenvectors with eigenvalues $\pm i$. The degenerate eigenspace is spanned by stabilizer states $|\Psi^-\rangle$ and $\left|\Phi^{+i}\right\rangle$. the non-degenerate, non-stabilizer eigenvectors are:

$$\begin{aligned} |v_{+i}\rangle &= \frac{1}{2}(e^{i\pi/4}, -1, -1, e^{-i\pi/4}) \\ |v_{-i}\rangle &= \frac{1}{2}(e^{-i\pi/4}, 1, 1, e^{i\pi/4}) \end{aligned} \tag{E.1}$$

We can write the Pauli operator expansion of their contribution to the spectral decomposition as follows:

$$i(|v_{+i}\rangle\langle v_{+i}| - |v_{-i}\rangle\langle v_{-i}|) = -\frac{i}{2\sqrt{2}}(IX + XI + IY + YI) \tag{E.2}$$

The spectral decomposition of this Pauli expansion will of course be a stabilizer projector decomposition. We group commuting Pauli operators and use the fact that these sums have zero eigenvalues which reduces the number of terms in their

spectral decomposition:

$$\begin{aligned}\frac{1}{2}(IX + XI) &= |xx\rangle\langle xx| - |\bar{x}\bar{x}\rangle\langle\bar{x}\bar{x}| \\ \frac{1}{2}(IY + YI) &= |yy\rangle\langle yy| - |\bar{y}\bar{y}\rangle\langle\bar{y}\bar{y}|\,.\end{aligned} \tag{E.3}$$

Hence the stabilizer projector rank of $\sqrt{W} \otimes \sqrt{W}$ is six and the decomposition is:

$$\begin{aligned}\sqrt{W} \otimes \sqrt{W} = &\left|\Psi^{+}_{zz}\right\rangle\!\left\langle\Psi^{+}_{zz}\right| + \left|\Phi^{+i}_{zz}\right\rangle\!\left\langle\Phi^{+i}_{zz}\right| \\ &- \frac{i}{\sqrt{2}}\Big(|xx\rangle\langle xx| - |\bar{x}\bar{x}\rangle\langle\bar{x}\bar{x}| \\ &+ |yy\rangle\langle yy| - |\bar{y}\bar{y}\rangle\langle\bar{y}\bar{y}|\Big)\end{aligned} \tag{E.4}$$

Bibliography

[AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 333–342, 2011.

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[AC17] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. In *32nd Computational Complexity Conference (CCC 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[ACP87] Stefan Arnborg, Derek G Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in ak-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987.

[AG04] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004.

[App09] DM Appleby. Properties of the extended Clifford group with applications to SIC-POVMs and MUBs. *arXiv preprint arXiv:0909.5233*, 2009.

[BBC+19] Sergey Bravyi, Dan Browne, Padraic Calpin, Earl Campbell, David Gosset, and Mark Howard. Simulation of quantum circuits by low-rank stabilizer decompositions. *Quantum*, 3:181, 2019.

[Bel64] John S Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1(3):195, 1964.

[BG16] Sergey Bravyi and David Gosset. Improved classical simulation of quantum circuits dominated by clifford gates. *Physical review letters*, 116(25):250501, 2016.

[BIS+18] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J Bremner, John M Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595–600, 2018.

[BISN17] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, and Hartmut Neven. Simulation of low-depth quantum circuits as complex undirected graphical models. *arXiv preprint arXiv:1712.05384*, 2017.

[BJS11] Michael J Bremner, Richard Jozsa, and Dan J Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2126):459–472, 2011.

[BK05] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Physical Review A*, 71(2):022316, 2005.

[BK19] Kaifeng Bu and Dax Enshan Koh. Efficient classical simulation of clifford circuits with nonstabilizer input states. *Physical review letters*, 123(17):170502, 2019.

[BRS12] Stephen D Bartlett, Terry Rudolph, and Robert W Spekkens. Recon-

struction of gaussian quantum mechanics from Liouville mechanics with an epistemic restriction. *Physical Review A*, 86(1):012103, 2012.

[BSS16] Sergey Bravyi, Graeme Smith, and John A Smolin. Trading classical and quantum computational resources. *Physical Review X*, 6(2):021043, 2016.

[CAB12] Earl T Campbell, Hussain Anwar, and Dan E Browne. Magic-state distillation in all prime dimensions using quantum reed-muller codes. *Physical Review X*, 2(4):041021, 2012.

[CLG+20] Ming-Cheng Chen, Riling Li, Lin Gan, Xiaobo Zhu, Guangwen Yang, Chao-Yang Lu, and Jian-Wei Pan. Quantum-teleportation-inspired algorithm for sampling large random quantum circuits. *Physical Review Letters*, 124(8):080502, 2020.

[Cop02] Don Coppersmith. An approximate fourier transform useful in quantum factoring. *arXiv preprint quant-ph/0201067*, 2002.

[Cuf17] Michael E Cuffaro. On the significance of the gottesman–knill theorem. *British Journal for the Philosophy of Science*, 68(1):91–121, 2017.

[CZH+18] Jianxin Chen, Fang Zhang, Cupjin Huang, Michael Newman, and Yaoyun Shi. Classical simulation of intermediate-size quantum circuits. *arXiv preprint arXiv:1805.01450*, 2018.

[CZX+18] Zhao-Yun Chen, Qi Zhou, Cheng Xue, Xia Yang, Guang-Can Guo, and Guo-Ping Guo. 64-qubit quantum circuit simulation. *Science Bulletin*, 63(15):964–971, 2018.

[DDM03] Jeroen Dehaene and Bart De Moor. Clifford group, stabilizer states, and linear and quadratic operations over GF(2). *Physical Review A*, 68(4):042318, 2003.

[DE03] David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[Dev] Luc Devroye. Non-uniform random variate generation.

[ECP10] Jens Eisert, Marcus Cramer, and Martin B Plenio. Colloquium: Area laws for the entanglement entropy. *Reviews of Modern Physics*, 82(1):277, 2010.

[FE08] Christopher Ferrie and Joseph Emerson. Frame representations of quantum mechanics and the necessity of negativity in quasi-probability representations. *Journal of Physics A: Mathematical and Theoretical*, 41(35):352001, 2008.

[Fey82] Richard P Feynman. Simulating physics with computers. *Int. J. Theor. Phys*, 21(6/7), 1982.

[FHS17] Christopher A Fuchs, Michael C Hoang, and Blake C Stacey. The SIC question: History and state of play. *Axioms*, 6(3):21, 2017.

[FNW92] Mark Fannes, Bruno Nachtergaele, and Reinhard F Werner. Finitely correlated states on quantum spin chains. *Communications in mathematical physics*, 144(3):443–490, 1992.

[GC99] Daniel Gottesman and Isaac L Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402(6760):390–393, 1999.

[GD12] Vibhav Gogate and Rina Dechter. A complete anytime algorithm for treewidth. *arXiv preprint arXiv:1207.4109*, 2012.

[GM14] Hector J Garcia and Igor L Markov. Simulation of quantum circuits via stabilizer frames. *IEEE Transactions on Computers*, 64(8):2323–2336, 2014.

[GMC12] Hector J Garcia, Igor L Markov, and Andrew W Cross. Efficient inner-product algorithm for stabilizer states. *arXiv preprint arXiv:1210.6646*, 2012.

[GN03] Rémi Gribonval and Morten Nielsen. Sparse representations in unions of bases. *IEEE transactions on Information theory*, 49(12):3320–3325, 2003.

[Got98a] Daniel Gottesman. Fault-tolerant quantum computation with higher-dimensional systems. In *NASA International Conference on Quantum Computing and Quantum Communications*, pages 302–313. Springer, 1998.

[Got98b] Daniel Gottesman. The heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006*, 1998.

[Gro96] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[Has07] Matthew B Hastings. An area law for one-dimensional quantum systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(08):P08024, 2007.

[HC17] Mark Howard and Earl Campbell. Application of a resource theory for magic states to fault-tolerant quantum computing. *Physical review letters*, 118(9):090501, 2017.

[HDDM05] Erik Hostens, Jeroen Dehaene, and Bart De Moor. Stabilizer states and Clifford operations for systems of arbitrary dimensions and modular arithmetic. *Physical Review A*, 71(4):042315, 2005.

[HHL09] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[HL19] Yifei Huang and Peter Love. Approximate stabilizer rank and improved weak simulation of clifford-dominated circuits for qudits. *Physical Review A*, 99(5):052307, 2019.

[HL21] Yifei Huang and Peter Love. Feynman-path-type simulation using stabilizer projector decomposition of unitaries. *Physical Review A*, 103(2):022428, 2021.

[HNS20] Cupjin Huang, Michael Newman, and Mario Szegedy. Explicit lower bounds on strong quantum simulation. *IEEE Transactions on Information Theory*, 2020.

[How15] Mark Howard. Maximum nonlocality and minimum uncertainty using magic states. *Physical Review A*, 91(4):042103, 2015.

[HV12] Mark Howard and Jiri Vala. Qudit versions of the qubit $\pi/8$ gate. *Physical Review A*, 86(2):022316, 2012.

[HWVE14] Mark Howard, Joel Wallman, Victor Veitch, and Joseph Emerson. Contextuality supplies the 'magic'for quantum computation. *Nature*, 510(7505):351, 2014.

[HZN+20] Cupjin Huang, Fang Zhang, Michael Newman, Junjie Cai, Xun Gao, Zhengxiong Tian, Junyin Wu, Haihong Xu, Huanjun Yu, Bo Yuan, et al. Classical simulation of quantum supremacy circuits. *arXiv preprint arXiv:2005.06787*, 2020.

[JCZ+00] D Jaksch, JI Cirac, P Zoller, SL Rolston, R Côté, and MD Lukin. Fast quantum gates for neutral atoms. *Physical Review Letters*, 85(10):2208, 2000.

[Jon13] Cody Jones. Low-overhead constructions for the fault-tolerant toffoli gate. *Physical Review A*, 87(2):022328, 2013.

[KL17] Lucas Kocia and Peter Love. Discrete Wigner formalism for qubits and noncontextuality of Clifford gates on qubit stabilizer states. *Physical Review A*, 96(6):062134, 2017.

[KL18] Lucas Kocia and Peter Love. Measurement contextuality and planck's constant. *New Journal of Physics*, 20(7):073020, 2018.

[KL19] Lucas Kocia and Peter Love. The non-disjoint ontic states of the grassmann ontological model, transformation contextuality, and the single qubit stabilizer subtheory. *Journal of Physics A: Mathematical and Theoretical*, 52(9):095303, 2019.

[KLM01] Emanuel Knill, Raymond Laflamme, and Gerald J Milburn. A scheme for efficient quantum computation with linear optics. *nature*, 409(6816):46–52, 2001.

[KTG+07] Jens Koch, M Yu Terri, Jay Gambetta, Andrew A Houck, DI Schuster, J Majer, Alexandre Blais, Michel H Devoret, Steven M Girvin, and Robert J Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Physical Review A*, 76(4):042319, 2007.

[LBMW03] Dietrich Leibfried, Rainer Blatt, Christopher Monroe, and David Wineland. Quantum dynamics of single trapped ions. *Reviews of Modern Physics*, 75(1):281, 2003.

[ME12] Andrea Mari and Jens Eisert. Positive Wigner functions render classical simulation of quantum computation efficient. *Physical Review Letters*, 109(23):230503, 2012.

[MFIB18] Igor L Markov, Aneeqa Fatima, Sergei V Isakov, and Sergio Boixo. Quantum supremacy is both closer and farther than it appears. *arXiv preprint arXiv:1807.10749*, 2018.

[MMK+95] Chris Monroe, David M Meekhof, Barry E King, Wayne M Itano, and

David J Wineland. Demonstration of a fundamental quantum logic gate. *Physical review letters*, 75(25):4714, 1995.

[MS08] Igor L Markov and Yaoyun Shi. Simulating quantum computation by contracting tensor networks. *SIAM Journal on Computing*, 38(3):963–981, 2008.

[Nat95] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[NC11] Michael A Nielsen and Isaac L Chuang. Quantum computation and quantum information: 10th. *New York, NY, USA: Cambridge University Press*, 1107002176:9781107002173, 2011.

[Nes08] M Nest. Classical simulation of quantum computation, the gottesman-knill theorem, and slightly beyond. *arXiv preprint arXiv:0811.0898*, 2008.

[Nes09] M Nest. Simulating quantum computers with probabilistic methods. *arXiv preprint arXiv:0911.1624*, 2009.

[NLPD+19] John Napp, Rolando L La Placa, Alexander M Dalzell, Fernando GSL Brandao, and Aram W Harrow. Efficient classical simulation of random shallow 2d quantum circuits. *arXiv preprint arXiv:2001.00021*, 2019.

[OFV09] Jeremy L O'brien, Akira Furusawa, and Jelena Vučković. Photonic quantum technologies. *Nature Photonics*, 3(12):687–695, 2009.

[Pre18] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[PWB15] Hakop Pashayan, Joel J Wallman, and Stephen D Bartlett. Estimating outcome probabilities of quantum circuits using quasiprobabilities. *Physical Review Letters*, 115(7):070501, 2015.

[PZ21] Feng Pan and Pan Zhang. Simulating the sycamore quantum supremacy circuits. *arXiv preprint arXiv:2103.03074*, 2021.

[SB09] Dan Shepherd and Michael J Bremner. Temporally unstructured quantum computation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2105):1413–1439, 2009.

[Shi17] Andrew Shi. Recursive path-summing simulation of quantum computation. *arXiv preprint arXiv:1710.09364*, 2017.

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[Spe08] Robert W Spekkens. Negativity and contextuality are equivalent notions of nonclassicality. *Physical Review Letters*, 101(2):020401, 2008.

[Spe16] Robert W Spekkens. Quasi-quantization: classical statistical theories with an epistemic restriction. In *Quantum Theory: Informational Foundations and Foils*, pages 83–135. Springer, 2016.

[SRP+20] James R Seddon, Bartosz Regula, Hakop Pashayan, Yingkai Ouyang, and Earl T Campbell. Quantifying quantum speedups: improved classical simulation from tighter magic monotones. *arXiv preprint arXiv:2002.06181*, 2020.

[Sto83] Larry Stockmeyer. The complexity of approximate counting. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 118–126. ACM, 1983.

[SWM10] Mark Saffman, Thad G Walker, and Klaus Mølmer. Quantum information with rydberg atoms. *Reviews of modern physics*, 82(3):2313, 2010.

[TD02] Barbara M Terhal and David P DiVincenzo. Adaptive quantum computation, constant depth quantum circuits and arthur-merlin games. *arXiv preprint quant-ph/0205133*, 2002.

[Tod91] Seinosuke Toda. Pp is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.

[Vid04] Guifré Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Physical review letters*, 93(4):040502, 2004.

[VWFE13] Victor Veitch, Nathan Wiebe, Christopher Ferrie, and Joseph Emerson. Efficient simulation scheme for a class of quantum optics experiments with non-negative physical Review Letters representation. *New Journal of Physics*, 15(1):013037, 2013.

[Woo87] William K Wootters. A Wigner-function formulation of finite-state quantum mechanics. *Annals of Physics*, 176(1):1–21, 1987.

[WSB+04] Andreas Wallraff, David I Schuster, Alexandre Blais, Luigi Frunzio, R-S Huang, Johannes Majer, Sameer Kumar, Steven M Girvin, and Robert J Schoelkopf. Strong coupling of a single photon to a superconducting qubit using circuit quantum electrodynamics. *Nature*, 431(7005):162–167, 2004.

[Zhu10] Huangjun Zhu. SIC POVMs and Clifford groups in prime dimensions. *Journal of Physics A: Mathematical and Theoretical*, 43(30):305305, 2010.

[ZLC00] Xinlan Zhou, Debbie W Leung, and Isaac L Chuang. Methodology for quantum logic gate construction. *Physical Review A*, 62(5):052316, 2000.