THE UNIVERSITY OF CHICAGO

HIGH-PERFORMANCE COMPUTING FOR QUANTUM INFORMATION SCIENCE

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF PHYSICS

BY
MINZHAO LIU

CHICAGO, ILLINOIS
JUNE 2024

To my parents, Gang Liu and Lihua Che

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I have had the fortune to pursue my studies in Physics in Illinois for eight years, which is roughly a third of my life. These eight years have had a profound impact on my attitude, ethics, values, skills, competence, knowledge, and vision. Along this journey, it is the people that truly infected me with their passion, dedication, ingenuity, and generosity, which still inspires me today to become a better scientist who makes a difference in the scientific world and pass down the torch of the collective scientific pursuit.

First, I would like to thank my advisor, Dr. Yuri Alexeev at Argonne National Laboratory, for his time and commitment to scientific discovery, pursuit of great impact, and dedication to mentorship. He always involves me in discussions of innovative new directions and ideas, and pushes me to explore a wide variety of topics to gain competence in a broad sense. I also deeply appreciate him for allowing me to take on a diverse set of roles with varied responsibilities, allowing me to growth as a scientist and a well-rounded academic.

I would also like to thank Professor Liang Jiang at the University of Chicago for the stimulating discussions and guidance, as well as the significant collaboration with Dr. Junyu Liu and Dr. Changhun Oh in association with his research group. They maintained high standards in our scientific inquiries, and taught me to be a scientist with originality and rigor. I would also like to thank Professor Woowon Kang and Professor David DeMille for advising me on my thesis committee, whose insights shaped me throughout my Ph.D. career.

What drove me into quantum information science is the influence from the professors of my alma mater, Illinois Wesleyan University. Professor Gabriel Spalding and Professor Narendra Jaggi dedicated their whole life in the education of their students, and I was fortunate to have them as my teachers, mentors, and inspiration.

I also am deeply indebted to my parents, for their generous support and sacrifice for my growth and persuits. Lastly, I would like to express gratitude to my wife, without whom my accomplishments would not be possible.

ABSTRACT

Quantum information science has gained significant attention in recent years, particularly due to the growing interest in quantum computing. In particular, computational tasks that are believed to take classical supercomputers years to complete have been executed in minutes with currently available noisy quantum computers, which refutes the extended Church-Turing thesis and has a monumental significance in theoretical computer science. Since we are still at the infancy of quantum computation, tremendous effort in simulating quantum systems with classical computers is needed to further our understanding of quantum information science. In this thesis, we focus on tensor network methods, and challenge some previous quantum supremacy claims. Specifically, we simulate Gaussian boson sampling quantum supremacy experiments on a classical supercomputer. Additionally, we show another numerical study that investigate the complexity of random quantum circuits. These exercises are crucial to the field as it elucidates the real requirements of achieving quantum supremacy.

CHAPTER 1

INTRODUCTION

Quantum technologies are positioned to revolutionize areas of science ranging from computation, cryptography, communication, and metrology to energy, pharmaceuticals, finance, and medicine. As the law that governs the fundamental interactions of the smallest particles of nature, quantum mechanics is the foundational theory that enables us to understand interactions and properties from electrons, atoms, and ions in the nanoscopic world to novel materials such as superconductors and topological insulators in the macroscopic world. More exhilarating, however, is the possibility for us to go beyond mere understanding of the nature, and leap to utilize the novel properties of quantum matters to achieve previously unimaginable scientific and engineering feats that can have monumental real-world impact. For example, superconductors allows currents to flow without resistance, which can be found in systems that require continuous large magnetic field generated by electrical current such as nuclear magnetic resonance (NMR) for medical imaging and the Large Hadron Collider (LHC) for particle physics. In the domain of communication, the quantum mechanical properties of light can be exploited to ensure that two parties can communicate securely, and intercepting information by the eavesdropper is forbidden by the laws of physics. In the domain of metrology, the sensing sensitivity can be improved by utilizing multiparty-entanglement between measurement nodes, surpassing the limit of classical measurement.

The most exciting domain of quantum information science, however, is perhaps quantum computing, in which quantum mechanics enables a new computational model that encompass and exceeds the classical computational model, hence opening up the possibility of solving problems that would be otherwise impossible with only classical computational capabilities. First proposed by Richard Feynman [6], quantum computers are posited to be able to simulate quantum systems exponentially more efficiently than classical computers. Since then, numerous application beyond simulation has been discovered, and active research efforts

remain strong to broaden the range of applicability of quantum computing. Applications for future full-scale fault-tolerance quantum computers include factoring large integers [7], unstructured search [8], Hamiltonian simulation [9, 10, 11, 12], solving linear systems [13], and more [14]. For near-term devices that are limited by device noise and imperfections, proposals to achieve quantum enhancement include combinatorial optimization [15], quantum chemistry [16], machine learning [17, 18, 19, 20, 21], protein docking [22], are more [14].

Understanding of the fundamental physics underpinning these potential applications depends on theoretical analysis, experimental investigations, and numerical simulations. For proposals investigating future technologies that are not available today, theoretical and numerical tools are necessary, and numerical tools becomes even more important when analytical methods are intractable for complex systems. However, due to the exponential size of the Hilbert space, naive methods of simulating quantum systems is impossible for systems of even moderate sizes. Therefore, novel numerical protocol that exploit the structure of the quantum states in which we are interested, such as ground states of Hamiltonians, states of shallow circuits, etc., are necessary to reduce the computational complexity. With the advancement in quantum simulation theories and high-performance computing technologies, large quantum systems simulations are feasible in many interesting cases. Specifically, tensor network methods, a class of state-of-the-art simulation technique applied to many quantum systems, have substantial potential. Although there has already been substantial understanding in these methods, the research field is still rapidly evolving. This thesis explores the many tensor network methods in which large quantum systems can be efficiently simulated, and address questions in several areas of quantum information theory including quantum supremacy.

1.1 Review of quantum information

1.1.1 Pure states

We start with a brief review of quantum mechanics. A *physical* quantum state is a ray living in the Hilbert space $\mathcal{H}$ equipped with a complex inner product $\langle \cdot, \cdot \rangle$. For two elements $x, y \in \mathcal{H}$, the following properties must be satisfied:

$$\langle x, y \rangle = \overline{\langle y, x \rangle}, \tag{1.1}$$

$$\langle ax_1 + bx_2, y \rangle = a\langle x_1, y \rangle + b\langle x_2, y \rangle, \tag{1.2}$$

$$\langle x, x \rangle > 0 \text{ if } x \neq 0 \tag{1.3}$$

$$\langle x, x \rangle = 0 \text{ if } x = 0. \tag{1.4}$$

For physical states, the norm must be 1. As a result, all physical states only differ in their 'direction' and are therefore considered as 'rays' in $\mathcal{H}$. It is convenient to use the 'bra-ket' notation to represent physical states. For example, $|x\rangle, |y\rangle \in \mathcal{H}$ are two physical state vectors, and their dual vectors are $\langle x|, \langle y| \in \mathcal{H}^*$, where $\mathcal{H}^*$ is the dual space of $\mathcal{H}$. In our case, $\mathcal{H}^*$ defines a set of linear maps $\phi = \langle x| : \mathcal{H} \to \mathbb{C}$, and $\phi(y) \equiv \langle x, y \rangle \equiv \langle x|y \rangle$. This also defines a bijective mapping between the two spaces.

A d-dimensional space $\mathcal{H}$ has an orthonormal basis of d vectors. For a 2-dimensional systems, this represents a 'qubit' since it is like a quantum 'bit', where a bit can take on one of two values. A state of a qubit is an element in the 2-dimensional Hilbert space which has basis $\{|0\rangle, |1\rangle\}$, and any normalized linear combinations of the basis states are physical. Therefore, a qubit state can be written in general as

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle, \tag{1.5}$$

where $\theta \in [0, 2\pi)$, and $\phi \in [0, \pi)$. For $d > 2$, such systems are called 'qudits' in general. If multiple qudits are present, the total system state resides in the Hilbert space that is the tensor product space of all individual Hilbert spaces of the composing subsystems. For example, the Hilbert space of a 2-qubit system with qubit A and B with local Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$ is $\mathcal{H}_{A,B} = \mathcal{H}_A \otimes \mathcal{H}_B$. The joint system Hilbert space has four basis vectors, which can be denoted as $\{|0\rangle, |1\rangle, |2\rangle, |3\rangle\}$ or $\{|0,0\rangle, |0,1\rangle, |1,0\rangle, |1,1\rangle\}$, where $|i,j\rangle \equiv |i\rangle_A \otimes |j\rangle_B$.

However, in more complex cases where the subsystems are composed of indistinguishable particles instead of distinguishable qudits, further restrictions on physical states are necessary. Exchange of identical particles must not change the state vector up to a phase factor, and this phase factor is $+1, -1, e^{i\theta}$ for bosons, fermions, and anyons, respectively. This restriction leads to different statistics of identical particles of different types.

Quantum states can be transformed using linear operators. Physical transformations are represented by unitary operators $U : U^\dagger = U^{-1}$. The quantum state $|x\rangle$ becomes $U|x\rangle$ after transformation by U, whose dual vector is $\langle x|U^\dagger$. It can be easily seen that a change of basis by transforming the entire Hilbert space by U preserves the structure of the Hilbert space since $\langle x|y\rangle \to \langle x|U^\dagger U|y\rangle = \langle x|y\rangle$.

Observables such as energy, momentum, and position, which can in principle be measured for a state if sufficiently many copies are provided, correspond to Hermitian operators $\hat{O}$: $\hat{O} = \hat{O}^\dagger$. In the laboratory, a measurement on a quantum state would randomly 'collapse' the state to an eigenstate $|a\rangle$ of $\hat{O}$ with probability $|\langle a|\psi\rangle|^2$ and the measurement outcome is the eigenvalue. Observables have have real eigenvalues, which is sensible as any real-world measured quantities should be real. The expectation value of an observable is given by $\langle\hat{O}\rangle = \langle\psi|\hat{O}|\psi\rangle$ since $\langle\psi|\hat{O}|\psi\rangle = \langle\psi|\sum_a \lambda_a|a\rangle\langle a|\psi\rangle = \sum_a \lambda_a|\langle a|\psi\rangle|^2$, the mean value of the eigenvalues of the measurement outcomes. The uncertainties in the observable is given by $\Delta O = \sqrt{\langle\psi|\hat{O}^2|\psi\rangle - |\langle\psi|\hat{O}|\psi\rangle|^2}$.

Given a Hamiltonian $\hat{H}$, which is the Hermitian operator corresponding to the energy observable, the system evolves according to the Schrodinger equation

$$i\hbar\frac{d}{dt}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle, \tag{1.6}$$

whose solution is

$$|\Psi(t)\rangle = e^{-i\hat{H}t/\hbar}|\Psi(0)\rangle. \tag{1.7}$$

Since $\hat{H}$ is Hermitian, $e^{-i\hat{H}t/\hbar}$ is unitary. If $\hat{H}$ changes with respect to time, the unitary describing the transformation from the initial state to the final state is more complex, but given by the Dyson series.

1.1.2 Mixed states

Above is a completely quantum mechanical description of any system where no classical randomness is present. However, in the case where there is classical randomness or only a subsystem is available, the above picture may not be sufficient. For example, if a qubit is prepared by a device with 50-50 chance in state $|0\rangle$ or $|1\rangle$, how do we characterize the quantum state? Ideally, we would like to do whatever we have to do to the quantum state separately for each distinct possibility.

Another example is when qubit A is *entangled* with qubit B, and an example state is given by $|\psi\rangle = \frac{1}{\sqrt{2}}(|0,0\rangle + |1,1\rangle)$. This is entangled because if A is in state $|i\rangle$, then B is also in state $|i\rangle$. This entangled system correlation is different from classical correlation where qubit A and B are prepared with the same state randomly. For example, if consider the $\{|+\rangle, |-\rangle\}$ basis where $|+\rangle = \frac{1}{2}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{2}(|0\rangle - |1\rangle)$, the entangled state can be written as $|\psi\rangle = \frac{1}{\sqrt{2}}(|+,+\rangle + |-,-\rangle)$, and measurement outcomes in the new basis is still correlated. However, the 50-50 preparation case would give us uncorrelated outcomes. In the entangled case, if only subsystem A is accessible, the experimentalist would not be able

to tell apart if A is prepared with classical randomness or entangled with another system that is not accessible.

As a result, we would like a convenient way to describe states with classical correlation. These are called mixed-states, as opposed to state vectors described in the previous subsection which are called pure states. Specifically, given a quantum system with a Hilbert space, a mixed state is represented by a *density operator* $\hat{\rho}$ which is a positive semi-definite Hermitian operator with unit trace. Expectation of observables are given by $\langle \hat{O} \rangle = \mathrm{Tr}(\hat{\rho}\hat{O})$, and probability of measurement is $\langle a|\hat{\rho}|a\rangle$ with post measurement state $|a\rangle\langle a|$. This explains why $\hat{\rho}$ must have unit trace due to probability conservation. More generally, measurement could collapse a state onto a subspace when the observable has degenerate eigenvalues and $O = \sum_i \lambda_i P_i$, where P_i's are projection operators onto the subspace spanned by the eigenvectors corresponding to λ_i. The post measurement state is then $\frac{P_i \hat{\rho} P_i}{\mathrm{Tr}(\hat{\rho} P_i)}$ with probability $\mathrm{Tr}(\hat{\rho} P_i)$. It is easy to check that for with classical probabilities of the system being in pure states $|\psi_i\rangle$, the corresponding predictions of the probabilities and density operators obtained from $\hat{\rho} = \sum_i P(i)|\psi_i\rangle\langle\psi_i|$, where $P(i)$ is the probability for the state to be in state $|\psi_i\rangle$, is consistent with what we would get by adding the probability contribution from each case.

If a unitary process is applied to the quantum state, the density operator simply becomes $U\hat{\rho}U^\dagger$. One can also easily check that this is sensible for the pure state case. However, since classical randomness is included in the discussion, we need to capture the possibility of the transformation of the quantum state. What we need is a general linear map $\Phi : L(H) \to L(H)$ from the space of linear operators on H onto itself, where H is the space of all density operators. It is, however, possible to also define linear maps that more generally map onto linear maps of a different Hilbert space. Moreover, we need the linear map to map density operators to density operators, so they have to preserve Hermiticity, positivity, and trace. Moreover, if the map extends its action onto a second Hilbert space with trivial action, this map should remain positive. Such maps are called *completely positive and trace preserving*

(CPTP) maps, and admits the Kraus operator representation

$$\mathcal{E}(\hat{\rho}) = \sum_i K_i \hat{\rho} K_i^\dagger, \tag{1.8}$$

where $\sum_i K_i^\dagger K_i = \mathbb{I}$. The Kraus representation of the quantum channel $\mathcal{E}$ does not have to be unique. $\mathcal{E}$ is called a *superoperator*.

1.2 Models of quantum computation

Much of this thesis will be in the context of quantum computing, so models of quantum computation will be discussed. The most widely used formalism is the circuit model, which is the standard language for universal quantum computers. Another model is continuous variable quantum computing, which is universal, and contains the computational task called boson sampling, which is not universal but of theoretical and experimental importance. Although other models of computation such as measurement based quantum computing, quantum annealing, and adiabatic quantum computing exist, we focus this thesis on the circuit model and continuous variable systems.

1.2.1 The circuit model of quantum computation

In classical computing, information is stored in bits, which can take on values of 0 or 1. Further, the physical medium of information is electricity, and computation is done using electrical circuits with transistors. Therefore, classical computation is often represented as circuit diagrams, where the flow of information of bits is shown by wires and operations are represented by gates. For example, Fig. 1.1 represents a circuit for classical computing. The classical information is stored in two bits, and each row of the circuit represents one bit. The bits start from the left, and go into an exclusive-or (XOR) gate, which takes the the values of the bits as input from the left, and produces one bit of information as the output. For the

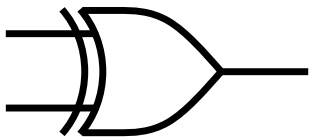

Figure 1.1: An XOR gate.

XOR gate, the output bit is 1 only if the two input bits have different values. We illustrate all the combinations of input bits and the corresponding output bit in a *truth table* Table 1.1.

Input 1	Input 2	Output
T	T	F
T	F	T
F	T	T
F	F	F

Table 1.1: Truth table of an exclusive-or (XOR) gate.

There are many other types of gates other than the XOR gate, and they have different truth tables that describe the outputs depending on the inputs, and they are represented by different symbols. Outputs from one gate can be fed into another gate as inputs, and this allows information to flow between gates and get processed, describing the process of computation. As an example, Fig. 1.2 describes a classical computational circuit used for a binary full adder. It takes two bits (A, B) and a carry bit (C) as input, and computes the binary addition result. The S bit is the normal output of the sum, and the C_{out} bit is the carry bit of the output. Information flows from the inputs through one gate to the next, eventually going to the output.

We notice that the number of bits at every instance of time can change, and the XOR gate is obviously not reversible (there are two combinations of input bits that produce the same output bit, so we cannot take the output bit and inversely deduce what the input bits are). However, our linear algebraic, pure state formalism of quantum mechanics does not change the dimension of the Hilbert space that the quantum state lives, and is completely

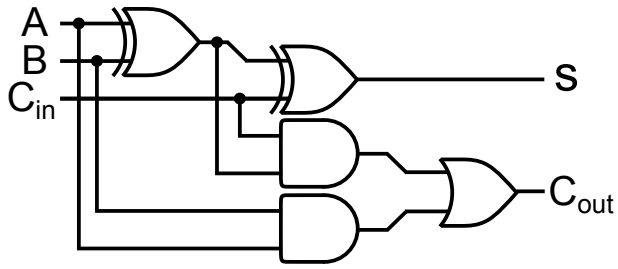

Figure 1.2: Circuit for a binary full adder.

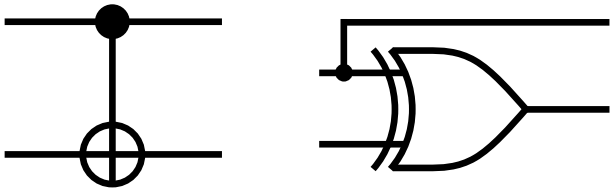

Figure 1.3: CNOT circuit. The left side is the representation used in quantum computation. The right side shows the classical logical gate equivalent.

reversible since unitary matrices are reversible. As a result, we would like to discuss a picture of classical computation that is more similar to our quantum formalism so that we can more easily generalize to quantum computation later. We introduce reversible circuits.

Take our XOR gate as an example. We can add an output bit to the gate by simply copying the value of the first input bit. In this case, we can view the effect of the XOR gate not as producing an output bit depending on the input bits, but conditionally flipping the second bit depending on the state of the first bit. The second bit is flipped only if the first gate is in state 1, and nothing happens to the first bit. In this case, we can deduce the input bits give two output bits. This is called the *controlled-not* (CNOT) gate, which is illustrated in Fig. 1.3 and the truth table is in Table 1.2.

Input 1	Input 2	Output 1	Output 2
T	T	T	F
T	F	T	T
F	T	F	T
F	F	F	F

Table 1.2: Truth table of an controlled-not (CNOT) gate.

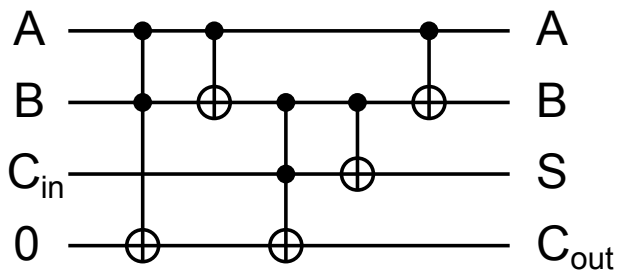

Figure 1.4: Full adder using only reversible gates [1].

The CNOT gate is nice due to its reversibility. In general, we can rewrite any classical computational program into circuit composed of reversible gates, with the same number of bits from start to end as shown in Fig. 1.4. Each gate acts on bits that are connected to them. The three-bit gates are called Toffoli gates, where a NOT operation is only applied when both bits of the solid dots are True. We can now apply the same picture to a quantum system, where each wire represents the change of qubit states in time, and gates are unitary operations that act on the connected qubits. What is different from the classical circuit model, however, is that the value of each qubit associated with each wire is no longer a deterministic classical value. First, each qubit can be an arbitrary normalized superposition (sum with complex amplitudes) of the states $\{|0\rangle, |1\rangle\}$. Second, the M-qubit system is in a 2^M dimensional Hilbert space, and the state can be an arbitrary normalized superposition of all M-qubit states $\{|0, 0, \ldots, 0\rangle, \ldots, |1, 1, \ldots, 1\rangle\}$. These M-body states cannot, in general, be written as tensor product states, which only delineates a tiny subset of physical states. The quantum correlation between qubits in general does not allow us to define the state of individual qubits.

It turns out that any arbitrary M-qubit unitary can be decomposed into only the two-qubit CNOT gate and all single qubit gates [23]. Further, the Solovay-Kitaev theorem states that the set of all single qubit unitaries can be efficiently approximated with an appropriately chosen set of single-qubit gates [24]. For example, the $\{\text{CNOT}, H, S, T\}$ gate set, where the

unitary matrices describing their actions on qubits are given by

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}, \tag{1.9}$$

can be used to efficiently approximate any unitary. There are different set of gates that can be combined to form arbitrary unitaries, and such sets are called universal gate sets. If a quantum computational device can manipulate the Hilbert space with a universal gate set, we can perform universal computation, meaning any pure-state quantum transformation is possible in principle.

1.2.2 Quantum information processing with continuous variable systems

We have introduced quantum computing with qubits, each of which lives in a two-dimensional Hilbert space, the full system lives in a finite-dimensional Hilbert space, and the measurement outcomes of these qubits take on discrete values. In some sense, one can understand this approach of quantum computing as digital. We now introduce quantum information processing with continuous variable systems, which can be understood as an analog version of quantum computing. Specifically, the observables we use (e.g. the strength of an electromagnetic field) take on continuous values, and the local Hilbert space is infinite dimensional. It is commonly realized using quantum optical systems. In this subsection, we follow the introduction outlined in [15], although notations and conventions may be slightly different.

The full system is composed of M modes, and we have a basis $\{|n_j\rangle\}$, where $n_j = 0, 1, \ldots,$ for the Hilbert space of the jth mode. Operator $a_j^\dagger$ is the input creation operator for mode j, and $|0\rangle$ is called the vacuum state. For photonic systems, each application of the creation operator raises the number of photons in the optical mode by one. Similarly, we have the annihilation operators a_j which reduce the number of photons by one. This is expressed in

the following equations:

$$a^\dagger|n\rangle = \sqrt{n+1}|n+1\rangle \tag{1.10}$$

$$a|n\rangle = \sqrt{n}|n-1\rangle \text{ for } n > 0 \tag{1.11}$$

$$a|0\rangle = 0 \tag{1.12}$$

$$N|n\rangle = a^\dagger a|n\rangle = n|n\rangle, \tag{1.13}$$

and it is easy to check that commutation relations $[a_i, a_j^\dagger] = \delta_{i,j}, [N, a^\dagger] = a^\dagger, [N, a] = a$ hold.

We have thus far been working with discrete observables still (the number of photons N), but we can define continuous observables for this infinite dimensional system. For photonic systems, each optical mode has creation and annihilation operators $a^\dagger, a$, and the position and momentum-like operators are:

$$q = \frac{1}{2}(a + a^\dagger),\ p = \frac{1}{2i}(a - a^\dagger). \tag{1.14}$$

It is easy to check that $[q_i, p_j] = \frac{i}{2}\delta_{i,j}$. The non-commutivity between q and p means that they do not have simultaneous eigenvectors, and therefore no state can have precisely determined p and q values at the same time, which is precisely the origin of the famous Heisenberg uncertainty principle. Specifically, $\Delta q \Delta p \geq \frac{1}{4}$.

For M-mode systems, one can introduce the vector of operators $\mathbf{S} = (q_1, \ldots, q_M, p_1, \ldots, p_M)^T$, and the commutation relations can be rewritten as:

$$[S_k, S_l] = \frac{i}{2}J_{kl}, \mathbf{J} = \begin{pmatrix} 0 & -\mathbb{I}_M \\ \mathbb{I}_M & 0 \end{pmatrix}. \tag{1.15}$$

To capture the covariance between the quadrature variables, we have the following covariance

matrix:

$$V_{kl} \equiv [\mathbf{V}]_{kl} = \frac{1}{2}\langle\{S_k, S_l\}\rangle - \langle S_k\rangle\langle S_l\rangle, \tag{1.16}$$

where the curly brackets indicate the anticommutator, and the angle bracket represents expectation value of the operator for the state, which is given by the trace for a mixed state. We can also look at the covariance of the mode operators, which is described by the covariance matrix $\boldsymbol{\sigma}$:

$$\sigma_{kl} = [\boldsymbol{\sigma}]_{kl} = \frac{1}{2}\langle\{R_k, R_l\}\rangle - \langle R_k\rangle\langle R_l\rangle = \frac{1}{2}\begin{pmatrix} \mathbb{I}_M & i\mathbb{I}_M \\ \mathbb{I}_M & -i\mathbb{I}_M \end{pmatrix} \mathbf{V} \begin{pmatrix} \mathbb{I}_M & \mathbb{I}_M \\ -i\mathbb{I}_M & i\mathbb{I}_M \end{pmatrix}, \tag{1.17}$$

where $\mathbf{R} = (a_1, \ldots, a_M, a_1^\dagger, \ldots, a_M^\dagger)$. Unlike $\mathbf{V}$, $\boldsymbol{\sigma}$ is complex-valued. We can also convert $\boldsymbol{\sigma}$ into $\mathbf{V}$ using the above equation.

For a particle with a pair of continuous conjugate observables (e.g. position and momentum) with density operator $\hat{\rho}$, these two variables form a phase space, and we can find a quasiprobability distribution of the particle in this phase space called the Wigner function:

$$W(x,p) \equiv \frac{1}{\pi\hbar}\int_{-\infty}^{\infty}\langle x+y|\hat{\rho}|x-y\rangle e^{2ipy/\hbar}dy. \tag{1.18}$$

One can similarly have a symmetrical definition of the Wigner function as an integral over the momentum space. It is called a quasiprobability because integration gives the correct marginal probability:

$$\int_{-\infty}^{\infty} W(x,p)dp = |\psi(x)|^2. \tag{1.19}$$

It is not an actual probability distribution because conjugate variables cannot both be well defined. In fact, the Wigner function can take on negative values.

Wigner function can be defined over the q and p operators defined above for photonic systems. For a vacuum state $|0\rangle$, the uncertainties in the two variables are Heisenberg limited (smallest product of uncertainties), and therefore has the smallest spread Wigner function.

Specifically, all operator expectation values are zero except the second moment, and the covariance matrix therefore gives a complete description of the state with $\mathbf{V} = \frac{1}{4}\mathbb{I}_{2M}$. There are other Heisenberg limited state as well, which corresponds to displacing and/or squeezing the state, which we will explain later.

Interaction Hamiltonians that are linear and bilinear in the optical modes are crucial to quantum information processing in such systems. In the context of boson sampling, both squeezing (used to produce the Gaussian input states) and beam splitter operations are such operations. The most general Hamiltonian is therefore:

$$H = \sum_{k=1}^{n} g_k^{(1)} a_k^\dagger + \sum_{k>l=1}^{n} g_{kl}^{(2)} a_k^\dagger a_l + \sum_{k,l=1}^{n} g_{kl}^{(3)} a_k^\dagger a_l^\dagger + h.c.. \tag{1.20}$$

Evolutions generated by such Hamiltonians are symplectic. Specifically, the coordinates transforms as:

$$\mathbf{S} \to \mathbf{Q}\mathbf{S} + \mathbf{d}_s, \tag{1.21}$$

where $\mathbf{Q}$ is symplectic and $\mathbf{d}_s$ is real. The covariance matrix evolves as:

$$\mathbf{V} \to \mathbf{Q}\mathbf{V}\mathbf{Q}^T. \tag{1.22}$$

The reason why symplectic transforms are interesting is because they preserve the commutation relations and equations of motion of the new coordinates, and Hamiltonians that generate them are experimentally realized in numerous ways.

The first term in the Hamiltonian is linear in the mode operators, and they generate the so called displacement operators:

$$D(\boldsymbol{\lambda}) = \bigotimes_{k=1}^{n} D_k(\lambda_k), \tag{1.23}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)^T$, $\lambda_k \in \mathbb{C}$, and $D_k(\lambda_k) = \exp(\lambda_k a_k^\dagger - \lambda_k^* a_k)$. It is called the displacement operator because

$$a_k \to a_k + \lambda_k, \tag{1.24}$$

$$\mathbf{S} \to \mathbf{S} + \mathbf{K}, \tag{1.25}$$

where $\mathbf{K} = (a_1, \ldots, a_n, b_1, \ldots, b_n)^T$ and $\lambda_k = a_k + ib_k$. Therefore, the Wigner function gets displaced after the displacement operator. Single mode coherent states $|\alpha\rangle$, commonly used as an idealized description of laser and can often be analyzed using classical descriptions, can be obtained by applying displacement on a single mode vacuum state:

$$|\alpha\rangle = e^{-\frac{1}{2}|\alpha|^2 \sum_{k=0}^{\infty} \frac{\alpha^k}{\sqrt{k!}}|k\rangle} = D(\alpha)|0\rangle. \tag{1.26}$$

Since displacement only shifts the Wigner function and the vacuum state is a minimum uncertainty state, coherent states are also minimum uncertainty states.

The second term in the Hamiltonian describe mixing of two modes, where creation of photons in one mode must be accompanied by destruction of photons in another (the total photon number is conserved). This can be realized by a beam splitter, and is called two-mode mixing. For mixing between mode j and k, the evolution is

$$U(\zeta) = \exp(\zeta a_j^\dagger a_k - \zeta^* a_j a_k^\dagger), \tag{1.27}$$

where $\zeta = \phi e^{i\theta} \in \mathbb{C}$. The transformation of the mode operators and the phase space

coordinates are:

$$\begin{pmatrix} a_j \\ a_k \end{pmatrix} \to \mathbf{B}_\zeta \begin{pmatrix} a_j \\ a_k \end{pmatrix}, \tag{1.28}$$

$$\mathbf{S} \to \mathbf{N}_\zeta \mathbf{S}, \tag{1.29}$$

$$\mathbf{N}_\zeta = \begin{pmatrix} \Re[\mathbf{B}_\zeta] & -\Im[\mathbf{B}_\zeta] \\ \Im[\mathbf{B}_\zeta] & \Re[\mathbf{B}_\zeta] \end{pmatrix}, \tag{1.30}$$

$$\mathbf{B}_\zeta = \begin{pmatrix} \cos\phi & e^{i\theta}\sin\phi \\ -e^{-i\theta}\sin\phi & \cos\phi \end{pmatrix}. \tag{1.31}$$

The final term is of the form $g^{(3)}a_j^{\dagger 2} + h.c.$ and $g^{(3)}a_j^\dagger a_k^\dagger + h.c.$, which describe single-mode and two-mode squeezing, respectively. The evolution of single-mode squeezing is

$$S(\xi) = \exp(\frac{1}{2}\xi a^{\dagger 2} - \frac{1}{2}\xi^* a^2). \tag{1.32}$$

Evolution of the phase space coordinates is given by:

$$\mathbf{S} \to \mathbf{\Sigma}_\xi \mathbf{S}, \tag{1.33}$$

$$\mathbf{\Sigma}_\xi = \mu \mathbb{I}_2 + \mathbf{R}_\xi \tag{1.34}$$

$$\mathbf{R}_\xi = \begin{pmatrix} \Re[\nu] & \Im[\nu] \\ \Im[\nu] & -\Re[\nu] \end{pmatrix}, \tag{1.35}$$

where $\mu = \cosh r, \nu = e^{i\psi}\sinh r, \xi = re^{i\psi}$, and the mean photon number is $|\nu|^2$. Application of squeezing to the vacuum state results in the so-called squeezed vacuum state:

$$|\xi\rangle = \frac{1}{\sqrt{\mu}} \sum_{k=0}^{\infty} \left(\frac{\nu}{2\mu}\right)^k \frac{\sqrt{(2k)!}}{k!} |2k\rangle. \tag{1.36}$$

The evolution of two-mode squeezing is

$$S_2(\xi) = \exp(\xi a_j^\dagger a_k^\dagger - \xi^* a_j a_k), \tag{1.37}$$

where we can similarly define μ, ν, ψ, r as in the single mode squeezing case. Evolution of the phase space coordinates is given by:

$$\mathbf{S} \to \mathbf{P}_{23}\mathbf{\Sigma}_{2\xi}\mathbf{P}_{23}\mathbf{S}, \tag{1.38}$$

$$\mathbf{\Sigma}_{2\xi} = \begin{pmatrix} \mu\mathbb{I}_2 & \mathbf{R}_\xi \\ \mathbf{R}_\xi & \mu\mathbb{I}_2 \end{pmatrix}, \tag{1.39}$$

$$\mathbf{P}_{23} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{1.40}$$

where $\mathbf{R}_\xi$ is from E.q. 1.35.

1.2.3 Boson sampling

Although the gate based circuit model of quantum computation is highly useful, implementing universal quantum computation using gates is challenging due to the need to precisely manipulate an experimental quantum system as described by the target unitaries. It is sensible to consider the possibility of exploring models of quantum computation that are more restrictive such that they are experimentally achievable. One prominent example is boson sampling, where only linear operations on bosons are considered, and the computation is not universal [25]. Despite the lack of universality, boson sampling is proven to be hard to simulate under plausible complexity theoretic conjectures [25, 26, 27]. Therefore, boson sampling is a very promising way to achieve near-term quantum supremacy.

We consider boson sampling where N independent input optical modes are sent into a linear optical interferometer. The interferometer has M modes, which can be larger than

N, making $M - N$ modes at the input vacuum states. As photons interact throughout the interferometer, the quantum state gets transformed according to a unitary matrix describing the interferometer, and photons eventually exit the M optical modes with non-trivial correlation. For boson sampling, the claim is that this process is hard to simulate for a sufficiently random unitary describing the interferometer.

Formally, the quantum state of N independent and identical modes can be written as:

$$|\psi_{\text{in}}\rangle = \otimes_{j=1}^{N} |\psi\rangle_j = \otimes_{j=1}^{N} \left(\sum_{n=0}^{\infty} c_n \frac{a_j^{\dagger n}}{\sqrt{n!}} \right) |0\rangle. \tag{1.41}$$

The action of an M-mode beam splitter array is to transform input creation operators:

$$a_j^\dagger \to \hat{b}_j^\dagger = \sum_{k=1}^{M} U_{jk} a_k^\dagger, \tag{1.42}$$

where the subscript j means they are operators on the jth optimal mode, and the $b^\dagger$ operators are creation operators on the output optical modes.

After the interferometer, the output state's photon numbers ($\mathbf{n} = \{n_0, n_1, \dots n_M\}$) at each mode are measured. Computing the output probability of boson sampling is #P-hard [25, 26]. For Fock state boson sampling, the initial quantum state is $|s_1, s_2, \dots, s_M\rangle$, which corresponds to $c_{s_j} = 1$ and all other coefficients are zero for each mode j. The probability of measuring the output state $|n_1, n_2, \dots, n_M\rangle$ is given by the permanent of a matrix:

$$p(\mathbf{n}) = \langle n_1, n_2, \dots, n_M | U | s_1, s_2, \dots, s_M \rangle = \frac{|\text{Per}(U_{S,T})|^2}{t_1! \dots t_M! s_1! \dots s_M!}, \tag{1.43}$$

where $\text{Per}(X)$ is called the permanent of an $N \times N$ matrix X:

$$\text{Per}(X) \equiv \sum_{\sigma \in S_N} \prod_{i=1}^{N} X_{i,\sigma(i)}, \tag{1.44}$$

and S_N is the symmetric group. The matrix $U_{S,T}$ is obtained from U by repeating the jth column of U t_j times to construct U_T and then repeat the jth row of U_T s_j times.

For Gaussian boson sampling, we have instead single-mode or two-mode squeezed vacuum states as the initial state. The state before measurement can be described by a quadrature operator covariance matrix $\mathbf{V}$ or a mode operator covariance matrix $\boldsymbol{\sigma}$. The probability is given by the hafnian of a matrix:

$$p(\mathbf{n}) = \frac{\mathrm{haf}(\mathbf{A}_\mathbf{n})}{n_1! \cdots n_M! \sqrt{\det(\boldsymbol{\Sigma})}} \tag{1.45}$$

$$\mathbf{A} = \mathbf{X}[\mathbb{I} - \boldsymbol{\Sigma}^{-1}] \tag{1.46}$$

$$\boldsymbol{\Sigma} = \boldsymbol{\sigma} + \frac{1}{2}\mathbb{I} \tag{1.47}$$

$$\mathbf{X} = \begin{pmatrix} 0 & \mathbb{I} \\ \mathbb{I} & 0 \end{pmatrix}, \tag{1.48}$$

where $\boldsymbol{\Sigma}$ is called the Husimi covariance matrix, and $\mathrm{haf}(X)$ is called the hafnian of a $2N \times 2N$ matrix X:

$$\mathrm{haf}(X) \equiv \frac{1}{N! 2^N} \sum_{\sigma \in S_{2N}} \prod_{j=1}^{N} X_{\sigma(2j-1),\sigma(2j)}, \tag{1.49}$$

and $A_\mathbf{n}$ is the matrix obtained by repeating the ith row and columns of the matrix A n_i times.

However, measuring the photon number at each output mode is not always possible. One class of experiments use threshold detectors, where the detectors can only differentiate between the presence or absence of photons, and $n > 0$ is always registered as 1. In this case, the probability is given by the torontonian of a matrix.

$$p(\mathbf{n}) = \frac{\mathrm{tor}(\mathbf{O})}{\sqrt{\det(\boldsymbol{\Sigma})}} \tag{1.50}$$

$$\mathbf{O} = \mathbb{I} - \boldsymbol{\Sigma}^{-1}, \tag{1.51}$$

where $\mathrm{tor}(X)$ is called the torontonian of a $2N \times 2N$ matrix X:

$$\mathrm{tor}(X) = \sum_{\sigma \in P([|N|])} \frac{(-1)^{|\sigma|}}{\sqrt{\det(\mathbb{I} - X_\sigma)}}, \tag{1.52}$$

where $P([|N|])$ is the power set of $[|N|] = \{1, 2, \ldots, N\}$ and X_σ is the sub matrix of X with rows j and $j + N$ and columns j and $j + N$, for all j in the set σ.

We have discussed the evolution of the covariance matrix under squeezing and two mode mixing, which is sufficient for describing lossless GBS. However, lossless GBS is unrealistic, and the difference between the two is significant. To incorporate photon loss into the description, a different approach is used to compute the probability. Let us first consider photon loss as coupling from the detectable system subspace to the undetectable environment. We can write a unitary matrix $U = \begin{pmatrix} T & P \\ Q & R \end{pmatrix}$ including these effects, where T describes the coupling between input modes and output modes in the detectable subspace. Therefore, T is non-unitary. By considering the environment input modes as having vacuum states, we can write down the overall system-environment final covariance matrix and obtain the detectable subspace covariance matrix by tracing out the environment. This leads to the final detectable subspace Husimi covariance matrix:

$$\mathbf{\Sigma} = \mathbb{I} - \frac{1}{2} \begin{pmatrix} T & 0 \\ 0 & T^* \end{pmatrix} \begin{pmatrix} T^\dagger & 0 \\ 0 & T^T \end{pmatrix} + \begin{pmatrix} T & 0 \\ 0 & T^* \end{pmatrix} \boldsymbol{\sigma}_{\mathrm{in}} \begin{pmatrix} T^\dagger & 0 \\ 0 & T^T \end{pmatrix}, \tag{1.53}$$

where $\boldsymbol{\sigma}_{\mathrm{in}}$ is the covariance matrix after single-mode or two-mode squeezing.

CHAPTER 2

TENSOR NETWORKS

With the formalism of representing and manipulating both pure and mixed states, it is in principle possible to simulate any quantum state and quantum processes on a classical computer. However, as we will discuss, this becomes impossible for large systems.

Consider a quantum system consisted of M qubits. The local Hilbert space of each qubit is 2-dimensional, and the M-qubit Hilbert space is the tensor product of all the local Hilbert spaces. Therefore, the total Hilbert space $\mathcal{H}$ is 2^M-dimensional. As a result, $\mathcal{H}$ has 2^M basis states forming the basis set $\{|i_0, i_2, \ldots, i_M\rangle : \forall i_0, i_2, \ldots, i_M \in \{0, 1\}\}$. An arbitrary state in $\mathcal{H}$ is a normalized superposition of the 2^M basis states

$$|\Psi\rangle = \sum_{i_0,i_2,\ldots,i_M \in \{0,1\}} c_{i_0,i_2,\ldots,i_M} |i_0, i_2, \ldots, i_M\rangle, \tag{2.1}$$

and requires 2^M complex amplitudes to describe. We see that if we scale up the system size, the memory required to store the quantum state in the classical computer would grow exponentially, and only small systems can be simulated in this way.

While multiple methods can simulate quantum systems efficiently, we focus on tensor networks in this thesis. A d-dimensional tensor has d indices, and can be graphically represented by a node with d bonds labeled by the respective index label. For example, a 1-dimensional tensor/vector $\mathbf{x}$ has 1 index i taking different values, enumerating all elements x_i. This is graphically represented as a node with name x and a bond i. Similarly, a 4-dimensional tensor $T_{i,j,k,l}$ has four indices and is represented by a node named T with bonds i, j, k, l. Contraction of tensors sums indices with the same name. For example, matrix multiplication $\mathbf{E} = \mathbf{CD}$ can be written in index notation as $E_{i,k} = C_{i,j} D_{j,k}$, where the shared index j is summed over. Graphically, this is represented by joining the j bond between the C, D nodes. Unsummed indices are open bonds. Fig. 2.1 provides a graphical representation of

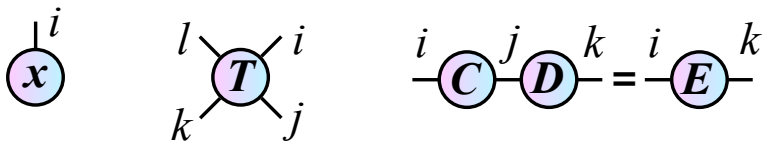

Figure 2.1: Graphical representation of a vector, a rank four tensor, and matrix multiplication.

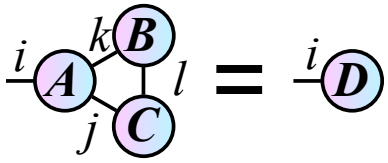

Figure 2.2: Tensor network of E.q. 2.2.

the above.

Tensor contractions generally results in tensors as well, which could be a constant as well in the case of 0-dimensional tensors. Since contraction of tensors can be represented by connecting shared bonds, this can be represented by a network of connected nodes, and we therefore obtain a *tensor network*. A tensor network is a graph that represents an equation of tensor contraction. An example of such an equation is

$$D_i = \sum_{j,k,l} A_{i,j,k} B_{k,l} C_{j,l}, \tag{2.2}$$

where three tensors A, B, C are contracted at their shared indices j, k, l, resulting in the output tensor D with one index i. One can see that matrix multiplications are a special kind of tensor contraction. The graphical representation of E.q. 2.2 is shown in Fig. 2.2.

An $M \times M$ unitary matrix can be expressed as matrix multiplications of individual unitaries acting on different subspaces. It is therefore possible to write the unitary of the quantum circuit as a contraction equation of individual gate unitaries and graphically represent it as a tensor network. It is instructive to consider the M-qubit unitary that corresponds to the product of unitaries on subsets of qubits. More concretely, $U = \prod_i U^{(i)}$, where $U^{(i)}$

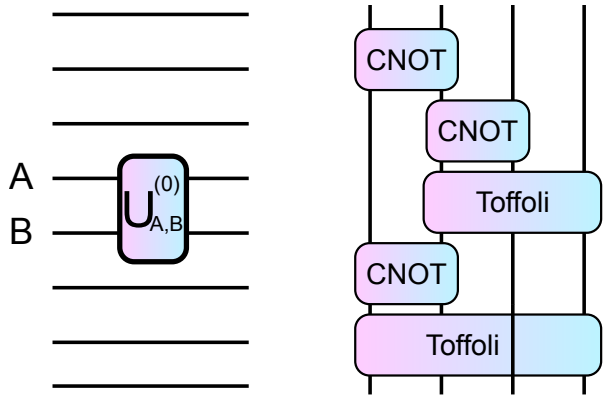

Figure 2.3: Left: graphical representation of $U^{(0)} = U^{(0)}_{A,B} \otimes \mathbb{I}^{\otimes M-2}$ acting non-trivially on qubit A and B. Right: graphical representation of the circuit in Fig. 1.4, oriented vertically.

acts non-trivially on a subset of qubits. For example, if $U^{(0)}$ only acts on qubit A and B, $U^{(0)} = U^{(0)}_{A,B} \otimes \mathbb{I}^{\otimes M-2}$, where $U^{(0)}_{A,B}$ is a 2-qubit unitary describing the action on qubit A and B. Therefore, $U^{(0)}$ can be graphically illustrated as in the left of Fig. 2.3, where lines represent identities. Layers of unitaries can be connected since all indices are contracted in matrix multiplication. As a result, the resulting unitary can be expressed as a tensor network of individual non-trivial sub-unitaries acting on subsystems. As an example, the right of Fig. 2.3 illustrates the tensor network corresponding to the quantum circuit of Fig. 1.4, oriented vertically.

2.1 Matrix product states and operators

This section is based heavily on [28, 29]. As we discussed earlier, the classical memory cost of an M-body pure state grows exponentially with M. One method of circumventing this issue is to representing the state efficiently using a tensor network. More explicitly, given an M-body pure state

$$|\Psi\rangle = \sum_{i_1,\ldots,i_M=0}^{d-1} c_{i_1,\ldots,i_M} |i_1,\ldots,i_M\rangle, \tag{2.3}$$

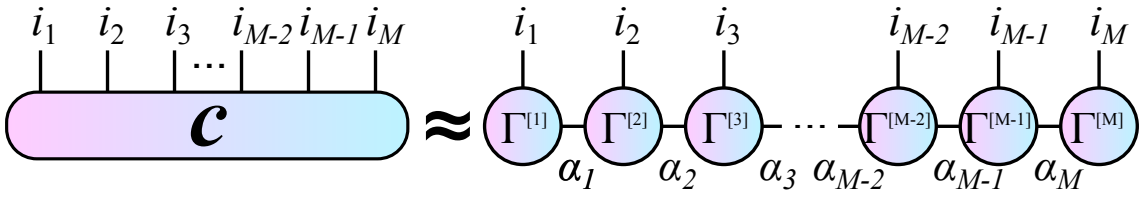

Figure 2.4: Matrix product representation of the M-body quantum state amplitude tensor $\mathbf{c}$ in E.q. 2.4. The graphical representation ignores the λ tensors since they can be easily absorbed into neighboring tensors.

we can efficiently represent it via its corresponding matrix product state (MPS) [30]

$$c_{i_1,\ldots,i_M} = \sum_{\alpha_0,\ldots,\alpha_{M-1}=0}^{\chi-1} \Gamma^{[1]i_1}_{\alpha 1} \lambda^{[1]}_{\alpha_1} \Gamma^{[2]i_2}_{\alpha_1 \alpha 2} \lambda^{[2]}_{\alpha_2} \times \ldots \lambda^{[M-1]}_{\alpha_{M-1}} \Gamma^{[M]i_M}_{\alpha_{M-1}}, \tag{2.4}$$

where d is the local Hilbert space dimension (a fixed property of the individual bodies) and χ is called the bond dimension. The MPS is graphically represented in Fig. 2.4 The tensor $c_{i_1,\ldots,i_M}$ fully characterizes the state $|\Psi\rangle$, but is M dimensional with d indices each, leading to M^d entries in storage. The MPS, however, represents this large tensor as a contraction (sum over the dummy or virtual α indices) of a chain of tensors, making it especially suitable for one-dimensional systems. One can observe that the i indices representing the physical degrees of freedom remain open (unsummed). The memory complexity of the MPS is $O(\chi^2 dM)$ [28, 29], and χ can be adjusted to represent c with the desired accuracy. Further, one can efficiently perform local unitary operations on the MPS and calculate expectation values of local observables with complexity $O(d^4\chi^3)$ [28, 29], allowing efficient simulation of one-dimensional systems.

For systems with low entanglement growth following area laws such as ground states of local Hamiltonians in D-dimensions with $M \propto l^D$ bodies, the entanglement grows as $O(l^{D-1})$ [31]. MPS also follows 1-dimensional system area law (constant) with $S = \log(\chi)$ [30], so to model higher dimensional systems, χ need to scale appropriately to capture the entanglement of the system.

The MPS formulation is especially convenient for quantifying entanglement. If we perform the Schmidt decomposition on the quantum state, which is to express the wavefunction as the sum of tensor products of states of two subsystems A and B

$$|\Psi\rangle = \sum_{\alpha} \lambda_\alpha |\alpha_A\rangle |\alpha_B\rangle, \tag{2.5}$$

where $\{|\alpha\rangle\}$ forms a basis set for each subsystem, we reveal the entanglement between the two subsystems, and the entanglement entropy (EE) given by

$$-\sum_{\alpha} \lambda_\alpha^2 \log \lambda_\alpha^2 \tag{2.6}$$

quantifies how much entanglement there is. Conveniently, if the subsystems are bipartitions of the MPS at site ℓ, the λ_α's would be $\lambda_{\alpha_\ell}^{[\ell]}$, allowing us to compute the MPS EE.

For a mixed state described by the density operator $\hat{\rho}$, the exact tensor is 2-dimensional with incoming and outgoing indices for rows and columns. The resulting graphical representation, therefore, has M input bonds and M output bonds. Similar efficient representation can be achieved via the matrix product operator (MPO) tensor network as illustrated in Fig. 2.5. We can similarly quantify entanglement in a mixed state. Specifically, we can treat all input bond the same way as output bonds, which results in an MPS with local dimension d^2. This corresponds to obtaining a state vector by flattening the density operator. We can formally perform Schmidt decomposition on the vectorized mixed state, identify the singular values λ_α with $\lambda_{\alpha_\ell}^{[\ell]}$ in the matrix product opoerator (MPO) representation, and similarly compute the MPO EE.

If unitary updates are applied to the MPS, the resulting wavefunction can be graphically represented as a tensor network in Fig. 2.6. To obtain the final state, one could proceed to contract the tensor network. However, it is easy to see that the MPS quickly becomes a high-rank tensor, incurring large costs. Therefore, we would like to maintain the low-

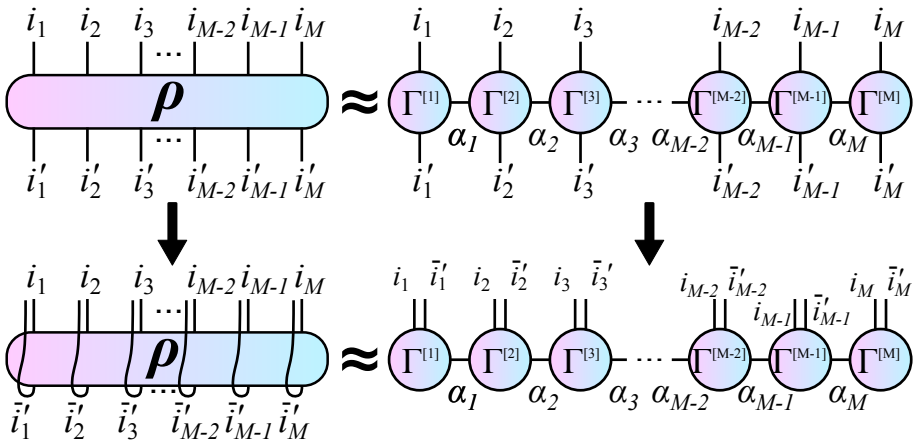

Figure 2.5: Matrix product operator of the M-body density operator ρ. The dual indices are denoted as primed indices i'. We can alternatively treat the dual indices as normal indices by vectorization, which changes i' into $\bar{i}'$.

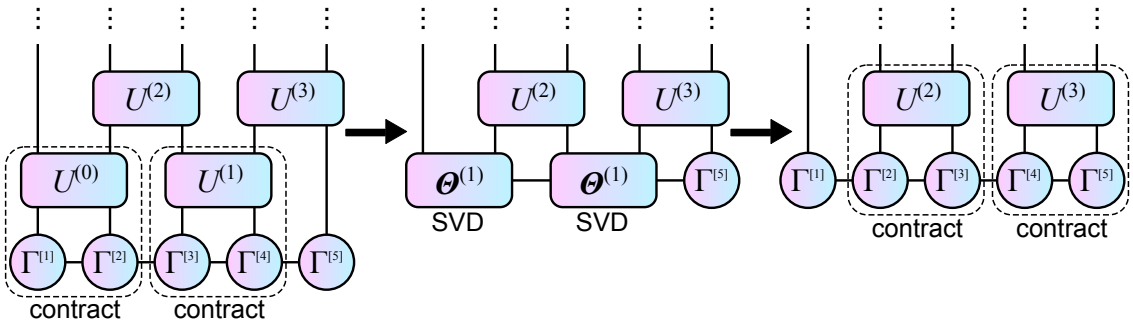

Figure 2.6: Time evolution of the approximate state vector by applying a sequence of unitaries. The tensors in the MPS are contracted with the unitaries, and the MPS form is restored by SVD. Singular values are truncated to keep at most χ bonds.

rank MPS structure throughout contraction and the desirable linear cost in M. For a local two-qudit unitary, the two tensors contracts with the 4-dimensional unitary, resulting in a 4-dimensional tensor. The total memory cost of the resulting tensor is $d^2\chi^2$ instead of $2d\chi^2$ before contraction (the cost of the unitary is d^4 and therefore ignored for large χ). We can restore the low linear memory cost by decomposing the 4-dimensional resulting tensor into two 3-dimensional tensors with a shared bond like in the original MPS. This can be accomplished via *singular value decomposition* (SVD). The full procedure is illustrated in Fig. 2.6.

We now mathematically describe the procedure discussed above. To apply a local unitary update on particle k and $k+1$ in an MPS, the unitary matrix needs to be contracted with the wavefunction at the physical indices, leading to the resulting tensor

$$\Theta^{j_k,j_{k+1}}_{\alpha_{k-1}\alpha_{k+1}} = \sum_{i_k,i_{k+1}=0}^{d-1} \sum_{\alpha_k=0}^{\chi-1} U^{j_k,j_{k+1}}_{i_k,i_{k+1}} \lambda^{[k-1]}_{\alpha_{k-1}} \Gamma^{[k]i_k}_{\alpha_{k-1}\alpha_k} \lambda^{[k]}_{\alpha_k} \Gamma^{[k+1]i_{k+1}}_{\alpha_k\alpha_{k+1}} \lambda^{[k+1]}_{\alpha_{k+1}}, \tag{2.7}$$

where the lower and upper indices of U represent input and output degrees of freedom, respectively.

The result of this computation is a single tensor of size $d^2\chi^2$, which should be used in the new representation of the wavefunction to replace $\lambda^{[k-1]}, \Gamma^{[k]i_k}, \lambda^{[k]}, \Gamma^{[k+1]i_{k+1}}, \lambda^{[k+1]}$. However, this is no longer in the form of an MPS. To restore the MPS form, singular value decomposition (SVD) is performed on Θ to produce

$$\Theta^{j_k,j_{k+1}}_{\alpha_{k-1}\alpha_{k+1}} = \sum_{\beta_k=0}^{d\chi-1} V_{(j_k,\alpha_{k-1}),\beta_k} \tilde{\lambda}^{[k]}_{\beta_k} W_{\beta_k,(j_{k+1},\alpha_{k+1})}. \tag{2.8}$$

By retaining only the χ largest singular values, we can identify new Γ tensors as

$$\tilde{\Gamma}^{[k]i_k}_{\alpha_{k-1}\alpha_k} = V_{(j_k,\alpha_{k-1}),\beta_k} / \lambda^{[k-1]}_{\alpha_{k-1}} \tag{2.9}$$

$$\tilde{\Gamma}^{[k+1]i_{k+1}}_{\alpha_k\alpha_{k+1}} = W_{\beta_k,(j_{k+1},\alpha_{k+1})} / \lambda^{[k+1]}_{\alpha_{k+1}}, \tag{2.10}$$

which restores the MPS form.

On the other hand, simulation of the evolution of mix states via MPOs require the use of quantum channels, which can be described by Kraus operators. Once again, if we only consider location operations on two qudits, the Kraus operators would contract with the local MPO tensors by summing over the input and output bonds. Further, unlike the MPS case, an additional bond connects the two Kraus operators due to the sum in the action of the quantum channel. Decomposition of the resulting tensor is similarly carried out to

restore the MPO form in order to control the complexity.

2.2 Circuit tensor network contraction

We discussed two important classes of tensor networks used in many-body physics, and they are especially suitable for 1-dimensional systems. We now discuss the simulation of arbitrary quantum circuits through tensor network contraction, which is similar to the aforementioned tensor network formalism but also has important differences.

As we discussed earlier, the unitary composed of local unitaries can be represented by a tensor network. If we want to compute the probability of measuring a bit-string s_{out} given the input bit string s_{in} and the unitary, it can be written as follows

$$P(s_{\text{out}}|s_{\text{in}}, U) = \langle s_{\text{out}}|U|s_{\text{in}}\rangle, \tag{2.11}$$

where $|s_{\text{in}}\rangle, |s_{\text{out}}\rangle$ are the input and output states. Since the input and output states are bit strings, this is nothing but the $s_{\text{in}}, s_{\text{out}}$-th entry of the unitary matrix $U_{s_{\text{in}},s_{\text{out}}}$, which can be found by setting the input and output bonds of the tensor network according to the bit strings as shown in Fig. 2.7. If we want to compute this quantity, we can contract the tensor network. For shallow circuits, we can find orders of tensor network contraction that allows us to incur minimal computational cost. For example, by contracting the tensor network in the order as shown in Fig. 2.7, the largest tensor that a classical computer needs to store has only a number of bonds proportional to the circuit depth and independent on the number of qubits. In general, circuits with limited connectivity and depth can be efficiently contracted if an appropriate contraction order is found.

This is in contrast to the MPS/MPO approach, where the contraction order is in time, and SVD is used to control the computational cost. The advantage of this approach is that it is exact and not limited to 1-dimensional systems. Although tensor networks like

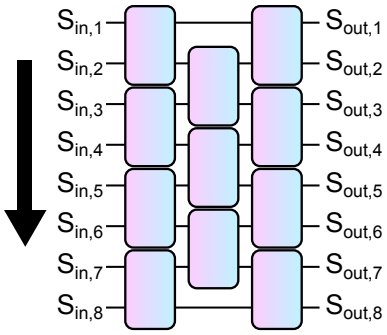

Figure 2.7: A good contraction order of a tensor network corresponding to the transition amplitude of a shallow quantum circuit is lateral.

MPS and MPO exist for higher dimensional systems, they are limited to use cases they are designed for if computational efficient is to be maintained. The disadvantage of the arbitrary circuit tensor network contraction approach, however, is that it is limited to shallow depth systems and cannot control the cost by approximation or exploiting limited entanglement. For circuits with large depths, the best contraction order leads to a cost comparable to a statevector simulator with additional overheads.

CHAPTER 3

EARLY EVIDENCE FOR EFFICIENT TENSOR NETWORK SIMULATION OF GAUSSIAN BOSON SAMPLING

This chapter is based heavily on [28]. Minzhao Liu is the primary author, who is responsible for the theoretical derivation of the entanglement entropy scaling, as well as the implementation of the custom GPU kernel and high performance parallelization simulation algorithm, building on Changhun Oh's previous CPU based algorithm.

As discussed in Chapter 1.2.3, boson sampling is a quantum computational model that has the potential of realizing quantum supremacy with noisy hardwares. It is non-universal, but has been proposed as a platform for practical applications such as calculating the molecular vibronic spectra [32, 33], molecular docking [22], and solving graph theory problems [34].

To demonstrate quantum supremacy with boson sampling, scientists at USTC [2] and Xanadu [4] performed Gaussian boson sampling in 2020 (Jiuzhang) and 2022 (Borealis), respectively. Additionally, USTC performed another boson sampling quantum supremacy in 2021 (Jiuzhang2.0) [3], and their most recent experiment is performed in 2023 (Jiuzhang3.0) [5]. All of the experiments are believed to require an astronomical amount of time to simulate on the most powerful classical supercomputer. However, other scientists immediately started to build better classical algorithms to simulate these quantum experiments and challenged the quantum supremacy claims to vary degrees. Specifically, our recent work shows that Gaussian boson sampling quantum supremacy experiments can in fact be simulated fairly quickly on a classical supercomputer, and the quality of the simulation is higher than the experiment under all thus far testable metrics [35]. This work will be discussed in depth in Chapter 4.

The workhorse of our algorithm is the MPS (MPO) tensor network. If photon loss in the system scales sufficiently rapidly with the system size, efficient classical simulation is possible in the asymptotic sense. Namely the crucial scaling that separates efficient and inefficient

classical simulation (at least for the best known classical algorithms [36, 37, 38, 39, 29, 28, 35]) is $N_{\text{out}} \propto \sqrt{N_{\text{in}}}$, meaning the number of photons surviving the interferometer scales as the square root of the number of input photons. For approximate tensor networks to be efficient, the entanglement entropy cannot increase too rapidly. Oh *et al.* [29] numerically and analytically show that the MPO entanglement entropy scales logarithmically for single photon boson sampling when photon loss is higher than the aforementioned scaling, which strongly suggest that efficient simulation may be possible.

We extend this result to the more experimentally relevant case of Gaussian boson sampling [28], which is the defacto scheme due to its scalability and is the currently dominant approach for demonstrating quantum supremacy. Specifically, an analytical derivation of the MPO entanglement entropy is provided, as well as numerical simulations to provide more evidence for the simulation complexity.

3.1 Analytical derivation of MPO entanglement entropy

In this section, we discuss the derivation of the operator EE scaling. We consider boson sampling where N independent input optical modes are sent into a linear optical interferometer. The interferometer has M modes, which can be larger than N, making $M - N$ modes at the input vacuum states. As photons interact throughout the interferometer, the quantum state gets transformed according to a unitary matrix describing the interferometer, and photons eventually exit the M optical modes with non-trivial correlation. For boson sampling, the claim is that this process is hard to simulate for a sufficiently random unitary describing the interferometer.

Formally, the quantum state of N independent and identical modes can be written as:

$$|\psi_{\text{in}}\rangle = \otimes_{j=1}^{N} |\psi\rangle_j = \otimes_{j=1}^{N} \left(\sum_{n=0}^{\infty} c_n \frac{a_j^{\dagger n}}{\sqrt{n!}} \right) |0\rangle, \tag{3.1}$$

where $a_j^\dagger$ is the input creation operator for mode j. The corresponding density operator is

$$\hat{\rho}_{\rm in} = \otimes_{j=1}^{N} |\psi\rangle_j \langle\psi|_j = \otimes_{j=1}^{N} \hat{\rho}_{j,{\rm in}}, \tag{3.2}$$

where the single input mode density operator is

$$\hat{\rho}_{j,{\rm in}} = \sum_{n,m=0}^{\infty} c_n c_m^* |n\rangle_j \langle m|_j. \tag{3.3}$$

The action of an M-mode beam splitter array is to transform input creation operators:

$$a_j^\dagger \to \hat{b}_j^\dagger = \sum_{k=1}^{M} U_{jk} a_k^\dagger, \tag{3.4}$$

where $\hat{b}_j^\dagger$ is the output creation operator for mode j. To study the entanglement entropy between bipartitions separated at the l-th mode, we can define the normalized up and down bipartition creation operators $\hat{B}_{{\rm u},j}^\dagger, \hat{B}_{{\rm d},j}^\dagger$ as

$$\cos\theta_j \hat{B}_{{\rm u},j}^\dagger = \sum_{k=1}^{l} U_{jk} a_k^\dagger, \quad \sin\theta_j \hat{B}_{{\rm d},j}^\dagger = \sum_{k=l+1}^{M} U_{jk} a_k^\dagger, \tag{3.5}$$

with normalizations

$$\cos^2\theta_j = \sum_{k=1}^{l} |U_{jk}|^2, \quad \sin^2\theta_j = \sum_{k=l+1}^{M} |U_{jk}|^2. \tag{3.6}$$

In the collision-free cases where $M \geq N^2$, the bipartition creation operators satisfy the canonical commutation relations

$$\begin{aligned} [\hat{B}_{{\rm u},j}, \hat{B}_{{\rm u},k}^\dagger] = \delta_{jk}, \quad [\hat{B}_{{\rm d},j}, \hat{B}_{{\rm d},k}^\dagger] = \delta_{jk}, \\ [\hat{B}_{{\rm u},j}, \hat{B}_{{\rm d},k}] = 0, \quad [\hat{B}_{{\rm u},j}, \hat{B}_{{\rm d},k}^\dagger] = 0. \end{aligned} \tag{3.7}$$

As a result, one can define the mutually orthogonal bipartition number states

$$\hat{B}^{\dagger k}_{\text{side},j}|0\rangle = \frac{|k\rangle_{\text{side},j}}{\sqrt{k!}}, \text{side} \in \{\text{u,d}\}. \tag{3.8}$$

The above formalism, described in [29], allows us to calculate the MPO EE without explicitly constructing the output state given the unitary representing the interferometer. Specifically, the details of the unitary matrix are hidden in the $|k\rangle_{\text{side},j}$ states constructed to satisfy orthogonality.

In this picture, the action of the unitary is to transform the input basis in the following way:

$$|n\rangle_j \to \sum_{k_j=0}^{n} \sqrt{\binom{n}{k_j}} \cos^{k_j}\theta_j \sin^{n-k_j}\theta_j |k_j\rangle_{\text{u},j}|n-k_j\rangle_{\text{d},j}, \tag{3.9}$$

and therefore (omitting the j index),

$$\langle k_\text{u}, k_\text{d}|U|n\rangle = \sqrt{\binom{n}{k_\text{u}}} \cos^{k_\text{u}}\theta \sin^{k_\text{d}}\theta \delta(k_\text{u}+k_\text{d}-n). \tag{3.10}$$

If we apply this basis transform due to the unitary to the input lossless density operator $\hat{\rho}_\text{int}$, each single mode input density operator $\hat{\rho}_{j,\text{in}}$ in Eq. 3.2 would transform independently. The full density operator remains a product of input modes in the new basis, and we have

$$\hat{\rho}_\text{out} = \otimes_j^N \hat{\rho}_{j,\text{out}}. \tag{3.11}$$

Although each $\hat{\rho}_{j,\text{out}}$ can be identified with an input mode j, $\hat{\rho}_{j,\text{out}}$ is no longer a single mode state, and is instead supported over all modes. Each $\hat{\rho}_{j,\text{out}}$ has some EE because it describes a state over both partitions, and the full system EE is additive in j due to the tensor product structure of $\hat{\rho}_\text{out}$. Therefore, the system EE scales linearly with the number of input modes N, and classical simulation of lossless boson sampling is always inefficient in N.

We can extend this analysis to lossy cases. Assuming that loss is uniform throughout the interferometer, loss commutes with all linear optical transforms and can be applied to the initial pure state. The basis transform 3.9 due to the unitary is still independent on j, and the total output density operator is still in a product form with

$$\hat{\rho}_{j,\text{out}} = U\mathscr{E}_{\text{loss}}(\hat{\rho}_{j,\text{in}})U^{\dagger}. \tag{3.12}$$

Therefore, the linear scaling of EE in N remains, and MPO simulations of lossy boson sampling is also inefficient in N.

However, as the number of input modes N increases, the complexity of the interferometer must grow as well in order to maintain reasonable randomness in the interferometer unitary and hardness of classical simulation. As a result, the depth of the interferometer should scale with the N, which leads to scaling of the transmission rate μ in N. The entanglement entropy for each $\hat{\rho}_{j,\text{out}}$ decreases as N increases, leading to an overall entanglement entropy that grows sublinearly, potentially allowing efficient simulation. To understand the scaling in loss and transmission, consider the Kraus operators corresponding to the single input state photon loss channel in the limit of small μ (from now on we ignore the mode index j)

$$\hat{\rho}_{\text{lossy}} = \mathscr{E}_{\text{loss}}(\hat{\rho}_{\text{in}}) = \sum_{n_{\text{loss}}=0}^{n_{\max}} K^{(n_{\text{loss}})}\hat{\rho}_{\text{in}}K^{(n_{\text{loss}})\dagger}, \tag{3.13}$$

$$K^{(n_{\text{loss}})} = \sum_{n_{\text{out}},n_{\text{in}}=0}^{n_{\max}} K^{(n_{\text{loss}})}_{n_{\text{out}},n_{\text{in}}}|n_{\text{out}}\rangle\langle n_{\text{in}}| \tag{3.14}$$

$$K^{(n_{\text{loss}})}_{n_{\text{out}},n_{\text{in}}} = \begin{cases} \sqrt{\binom{n_{\text{in}}}{n_{\text{out}}}\mu^{n_{\text{out}}}(1-\mu)^{n_{\text{loss}}}} & \text{if } n_{\text{in}} - n_{\text{out}} = n_{\text{loss}} \\ 0 & \text{otherwise,} \end{cases} \tag{3.15}$$

where $K^{(n_{\text{loss}})} \in \mathbb{C}^{n_{\max}+1,n_{\max}+1}$ captures processes that lose n_{loss} photons, and we limit the maximum photon number to $n_{\max}$. The lossy density operator can be given in the input

$|n\rangle$ basis in index notation:

$$\begin{aligned}\rho_{\text{lossy } m,n} &= \sum_{n_{\text{loss}}=0}^{n_{\max}} \sum_{k,l} K^{(n_{\text{loss}})}_{m,k} \rho_{\text{in } k,l} K^{(n_{\text{loss}})\dagger}_{l,n} \\ &= \sum_{n_{\text{loss}}=0}^{n_{\max}} O(\mu^{\frac{m}{2}}) \rho_{\text{in } m+n_{\text{loss}},n+n_{\text{loss}}} O(\mu^{\frac{n}{2}}) \\ &= O(\mu^{\frac{m+n}{2}}), \end{aligned} \tag{3.16}$$

where the second line is due to the requirement that $k - m = n_{\text{loss}}$ and $l - n = n_{\text{loss}}$ from non-zero Kraus operator elements.

We can now apply the basis transform due to the unitary as described in Eq. 3.9:

$$\begin{aligned}\hat{\rho}_{\text{out}} &= U \hat{\rho}_{\text{lossy}} U^\dagger \\ &= U \left(\sum_{m,n=0}^{n_{\max}} |m\rangle \rho_{\text{lossy } m,n} \langle n| \right) U^\dagger. \end{aligned} \tag{3.17}$$

In index notation in the bipartition number state basis,

$$\begin{aligned}\rho_{\text{out } k_{\text{u}},k_{\text{d}};k'_{\text{u}},k'_{\text{d}}} &= \langle k_{\text{u}}, k_{\text{d}} | \hat{\rho}_{\text{out}} | k'_{\text{u}}, k'_{\text{d}} \rangle \\ = \sum_{m,n=0}^{n_{\max}} \langle k_{\text{u}}, k_{\text{d}} | U | m \rangle \rho_{\text{lossy } m,n} \langle n | U^\dagger | k'_{\text{u}}, k'_{\text{d}} \rangle. \end{aligned} \tag{3.18}$$

Substituting Eq.3.10 and 3.16 into the above expression yields

$$\begin{aligned}&\rho_{\text{out } k_{\text{u}},k_{\text{d}};k'_{\text{u}},k'_{\text{d}}} \\ &= \sum_{n_{\text{loss}}=0}^{n_{\max}} \rho_{\text{in } k_u+k_d+n_{\text{loss}},k'_u+k'_d+n_{\text{loss}}} O(\mu^{\frac{k_{\text{u}}+k_{\text{d}}+k'_{\text{u}}+k'_{\text{d}}}{2}}). \end{aligned} \tag{3.19}$$

To compute the contribution to the full system MPO entanglement entropy from $\hat{\rho}_{j,\text{out}}$, we need to vectorize the density operator to obtain $|\hat{\rho}_{j,\text{out}}\rangle\rangle$ so that we can pretend it is a

pure state and compute its entanglement entropy. The standard procedure of computing the entanglement entropy of a pure state is to obtain the density operator by taking the outer product, obtain the reduced density operator by taking the partial trace over one subsystem, find the reduced density operator's eigenvalues, and take the log average of the eigenvalues. The only difference for the MPO entanglement entropy is that our 'pure' state is actually a vectorized density operator, and the eigenvalues may not be normalized since the vectorized state is not L^2 normalized.

Vectorization of the density operator, which corresponds to flattening of the matrix and changing kets into bras, is defined as:

$$\begin{aligned}
\hat{\rho}_{\text{out}} &= \sum_{k_\text{u},k_\text{d},k'_\text{u},k'_\text{d}} |k_\text{u},k_\text{d}\rangle \rho_{\text{out }k_\text{u},k_\text{d};k'_\text{u},k'_\text{d}} \langle k'_\text{u},k'_\text{d}| \\
\rightarrow |\hat{\rho}_{\text{out}}\rangle\rangle &= \sum_{k_\text{u},k_\text{d},k'_\text{u},k'_\text{d}} \rho_{\text{out }k_\text{u},k_\text{d};k'_\text{u},k'_\text{d}} |k_\text{u},k_\text{d};k'_\text{u},k'_\text{d}\rangle \\
&= \sum_{K_\text{u},K_\text{d}} \rho_{\text{out }K_\text{u},K_\text{d}} |K_\text{u},K_\text{d}\rangle,
\end{aligned} \tag{3.20}$$

where K_side is defined as the combined index of k_side and k'_side. Next, we take the partial trace of its outer product over one bipartition to obtain $\hat{\rho}' = \text{tr}_\text{u}(|\hat{\rho}_{j,\text{out}}\rangle\rangle\langle\langle\hat{\rho}_{j,\text{out}}|)$, yielding

$$\begin{aligned}
\rho'_{K_\text{d},\bar{K}_\text{d}} &= \sum_{K_\text{u}} \rho_{\text{out }K_\text{u},K_\text{d}}\ \rho^*_{\text{out }K_\text{u},\bar{K}_\text{d}} \\
&= \sum_{k_\text{u},k'_\text{u}} O(\mu^{\frac{k_\text{u}+k_\text{d}+k'_\text{u}+k'_\text{d}}{2}}) O(\mu^{\frac{k_\text{u}+\bar{k}_\text{d}+k'_\text{u}+\bar{k}'_\text{d}}{2}}) \\
&= O(\mu^{\frac{k_\text{d}+k'_\text{d}+\bar{k}_\text{d}+\bar{k}'_\text{d}}{2}}),
\end{aligned} \tag{3.21}$$

where $\bar{K}$ is the dual of K, and $k_\text{u}, k'_\text{u} = 0$ terms are dominant.

The above analysis is general and independent of the input states. From now on, we will use the fact that the input state is a squeezed state. In GBS, $\rho_{\text{in }m,n} = 0$ if either m

or n is odd. Therefore, looking at Eq. 3.19, $k_u + k_d + n_{\text{loss}}$ and $k'_u + k'_d + n_{\text{loss}}$ (or, more concisely, $k_u + k_d$ and $k'_u + k'_d$) must have the same parity for $\rho_{\text{in } k_u+k_d+n_{\text{loss}},k'_u+k'_d+n_{\text{loss}}}$ to be non-zero. This means that we do not have to consider terms like $\rho_{\text{out } 0,0;0,1}, \rho_{\text{out } 1,0;0,0}$, etc. As a result, no half-integer powers of μ occur in any terms of the output density operator $\hat{\rho}_{\text{out}}$ or the reduced density operator $\hat{\rho}'$ of the vectorized state.

In this case, it turns out that $\hat{\rho}'$ has exactly one constant order eigenvalue, no first order eigenvalues, and all other eigenvalues are at least second order. To show this, it is sufficient to find all eigenvalues to the first order. Let us write down the form of $\hat{\rho}'$ to the first order, with the first row corresponding to $K_{\text{d}} = k_{\text{d}} = k'_{\text{d}} = 0$ and the first column corresponding to $\bar{K}_{\text{d}} = \bar{k}_{\text{d}} = \bar{k}'_{\text{d}} = 0$:

$$\hat{\rho}' = \begin{bmatrix} \hat{\rho}'_{1,1} & \hat{\rho}'_{1,2} & \hat{\rho}'_{1,3} & \cdots \\ \hat{\rho}'^{*}_{1,2} & 0 & 0 & \vdots \\ \hat{\rho}'^{*}_{1,3} & 0 & 0 & \vdots \\ \vdots & \cdots & \cdots & \ddots \end{bmatrix}, \tag{3.22}$$

which has eigenvalues

$$\lambda^2 = \frac{1}{2}\left(\hat{\rho}'_{1,1} \pm \sqrt{\hat{\rho}'^{2}_{1,1} + 4\sum_{n=2}^{(n_{\max}+1)^2} |\hat{\rho}'_{1,n}|^2}\right), \tag{3.23}$$

and all other eigenvalues are 0. Note that we call the singular values of the Schmidt decompositions λ and the eigenvalues of the reduced density matrices λ^2. However, $|\hat{\rho}'_{1,n}|^2$ is at least $O(\mu^2)$, and the Taylor expansion of the square root will be dominated by the constant and first order contributions from $\hat{\rho}'_{1,1}$. Therefore, the only non-zero first order eigenvalue is $\lambda_1^2 = \hat{\rho}'_{1,1}$, which is $O(1)$.

The above analysis shows that to the second order, the eigenvalues are

$$\{a + b\mu + c\mu^2, O(\mu^2), O(\mu^2), O(\mu^2), \cdots\}. \tag{3.24}$$

After normalization of the eigenvalues, the entropy contribution due to λ_1^2 is

$$\begin{aligned} &- \frac{a+b\mu+c_1\mu^2}{C} \log \frac{a+b\mu+c_1\mu^2}{C} \\ &= \frac{(d-c_1)\mu^2}{a \ln 2} + O(\mu^3) = O(\mu^2), \end{aligned} \tag{3.25}$$

where c_i is the second order coefficient of λ_i^2, $d = \sum_i c_i$, and $C = a + b\mu + d\mu^2$ is the normalization that must be treated explicitly and not as as a constant. Contribution of other eigenvalues are

$$-\frac{c_i\mu^2}{C} \log_2 \frac{c_i\mu^2}{C} = O(-\mu^2 \log_2 \mu). \tag{3.26}$$

Overall, the entanglement entropy scales as $O(\mu^2 \log \mu)$. We would like to understand the scaling of the MPO EE under various loss scalings with the number of input optical modes. Generically, one can consider the situation where the number of surviving photons scales as $N_{\text{out}} \propto N^\gamma$, making the transmission rate $\mu = \beta N^\gamma / N$. Since the total entanglement entropy is the sum of all N input modes, we obtain

$$\begin{aligned} S_1(|\hat{\rho}\rangle\rangle) &= O\left(N \left(\frac{\beta N^\gamma}{N} \right)^2 \log_2 \left(\frac{\beta N^\gamma}{N} \right) \right) \\ &= O(N^{2\gamma-1} \log_2 N). \end{aligned} \tag{3.27}$$

Similarly, for the Rényi entropy, contribution from a single $\hat{\rho}_{j,\text{out}}$ is

$$\begin{aligned} &\frac{1}{1-\alpha} \log_2 \left[\left(\frac{a+b\mu+c_1\mu^2}{C} \right)^\alpha + \sum_{i\neq 1} \left(\frac{c_i\mu^2}{C} \right)^\alpha \right] \\ \approx &\frac{1}{(1-\alpha)\ln 2} \left(-\frac{d-c_1}{a} \alpha\mu^2 + \frac{1}{a} \sum_{i\neq 1} c_i^\alpha \mu^{2\alpha} \right). \end{aligned} \tag{3.28}$$

Therefore, for $\alpha < 1$, the second term dominates, and we have the familiar

$$S_\alpha = O(N^{1-2(1-\gamma)\alpha}). \tag{3.29}$$

Similarly, for $\alpha > 1$, the first term dominates, and we have

$$S_\alpha = O(\frac{\alpha}{1-\alpha} N^{2\gamma-1}). \tag{3.30}$$

In the case where $N_{\text{out}} \propto \sqrt{N_{\text{in}}}$ which is $\gamma = 1/2$, EE only grows logarithmically. This has a significant consequence in the simulation complexity. For an MPS algorithm, a logarithmic scaling of the MPS EE already rigorously implies a polynomial time complexity for the tensor network algorithm at fixed 2-norm distance between the ideal and approximate state. This implies efficient fixed fidelity simulation. The situation for the MPO algorithm is trickier. The logarithmic MPO EE now implies efficient simulation for fixed 2-norm distance between the vectorized states, which is also the 2-norm distance between the density operators. However, for fixed fidelity, one needs to bound the 1-norm distance, and the relationship $K||A||_2 \geq ||A||_1$, where K is the dimension of the Hilbert space, means that the one norm cannot be efficiently bounded. In some cases, the MPO EE decreases as the system size increases, reducing the required bond dimension to bound the 2-norm distance, but the required bond dimension to bound the 1-norm distance may still increase. This is the case for a sufficiently low γ such as $\gamma = \frac{1}{4}$. Overall, our result is evidence for efficient simulation when loss is greater than $\gamma = 1/2$, but it is not rigorous. As a result, we later provide numerical evidence on the simulation complexity directly.

3.2 Numerical simulation of MPO

We estimate the asymptotic MPO EE under photon survival scaling $N_{\text{out}} \propto N^\gamma$ with $\gamma = \frac{1}{4}, \frac{1}{2}, 1$. To make a fair comparison against SPBS, the squeezing parameter is fixed at $r = 0.88$,

which averages to approximately one photon per squeezed mode.

Fig. 3.1 shows the asymptotic estimates with $n_{\max} = 8$ (maximum number of photons per density operator ρ_j that is simulated) for large system sizes. Similar to what is observed in SPBS simulations, GBS shows MPO EE reduction when the loss is sufficiently high for $\gamma = \frac{1}{4}$, logarithmic scaling for $\gamma = \frac{1}{2}$, and linear scaling for $\gamma = 1$. A similar linear increase in MPO EE with β is also observed in all three cases. Further, we also show the numerical convergence of our asymptotic MPO EE estimates by increasing the cut-off of the initial maximum photon number $n_{\max}$ for the squeezed states.

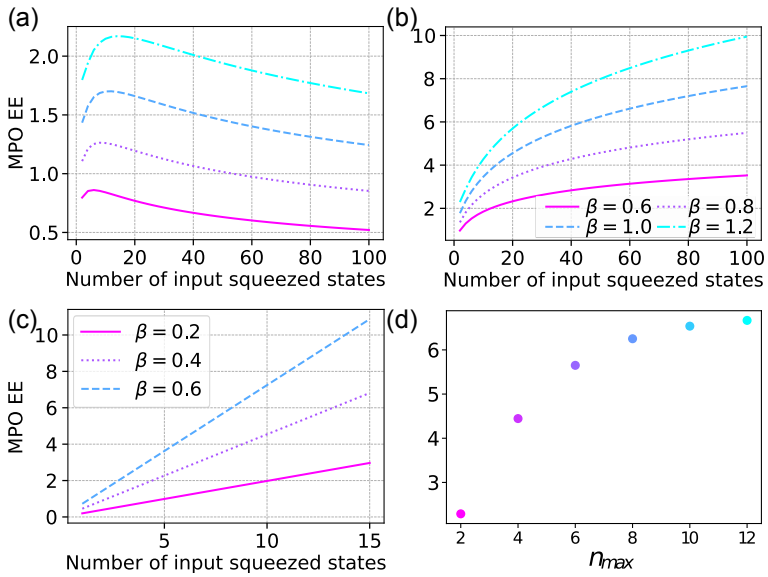

Figure 3.1: Operator entanglement entropy vs. the number of input squeezed modes for different photon survival scaling $N_{\text{out}} = \beta N^{\gamma}$ at $r = 0.88$. (a) $\gamma = \frac{1}{4}$. (b) $\gamma = \frac{1}{2}$. (c) $\gamma = 1$. (d) Convergence of MPO EE with increasing $n_{\max}$ for $N = 50, \beta = 1, \gamma = \frac{1}{2}, r = 0.88$.

We further conduct full MPO simulations of GBS using $U(1)$ symmetry and numerically calculate the MPO EE. The simulation algorithm is described in detail in Appendix A. As discussed in the analytic derivation of the MPO EE, all photon loss can be applied to the initial MPO, and the simulation afterwards is lossless. Therefore, the total photon number is preserved in the system. This leads to the so called $U(1)$ symmetry of time evolution, which can be exploited to reduce the memory and time cost of the simulation. We discuss

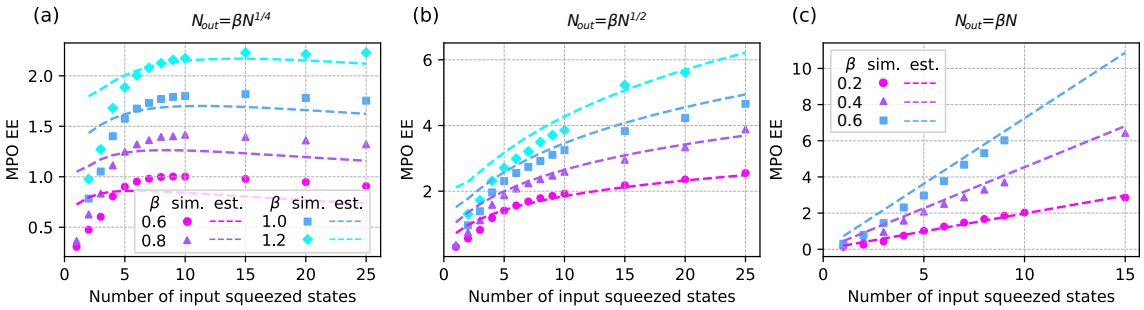

Figure 3.2: Operator entanglement entropy vs. the number of input squeezed modes for different photon survival scaling $N_{\text{out}} = \beta N^{\gamma}$ at $r = 0.88$. Details of experiment configurations can be found in Methods. (a) $\gamma = \frac{1}{4}$. (b) $\gamma = \frac{1}{2}$. (c) $\gamma = 1$. Dots are results obtained from full simulations using $U(1)$ symmetry. Dashed lines are estimates using asymptotic assumptions.

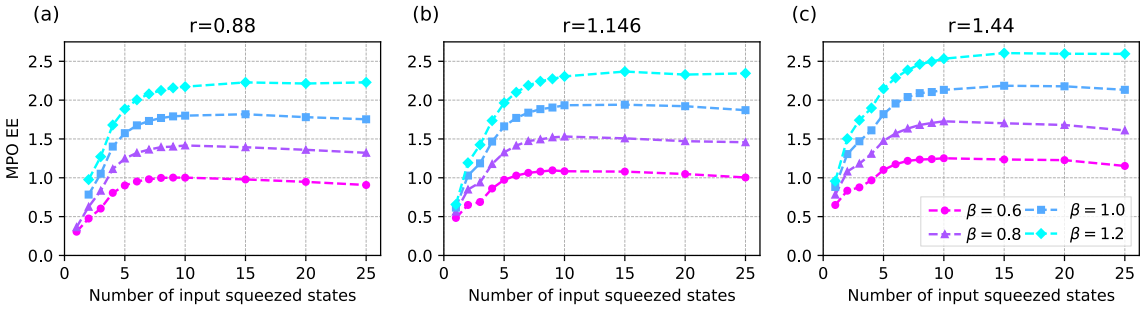

Figure 3.3: Operator entanglement entropy vs. the number of input squeezed modes for different squeezing parameters r. Dashed lines are guides to the eye. (a) $r = 0.88$, averaging approximately 1 photon per mode. (b) $r = 1.146$, averaging approximately 2 photons per mode. (b) $r = 1.44$, averaging approximately 4 photons per mode.

$U(1)$ symmetry in detail in Appendix A. The MPO EE obtained from the full simulations and asymptotic estimates agree quantitatively, as shown in Fig. 3.2. However, we observe that the quality of agreement is poor when MPO EE is small such as in many $\gamma = \frac{1}{4}$ data points when the number of input squeezed states N is small. In the regime of small MPO EE but large N, we attribute the disagreement to the formal differences between regular MPOs and MPOs in a $U(1)$ symmetric form. For small N, we expect the quality of the approximation to be poor because we are no longer in the asymptotic limit. Further disagreement can also be attributed to the fact that the full simulations are limited by the

bond dimension. We ensure that all plotted data points are simulated to $1 - \mathrm{Tr}(\hat{\rho}) < 0.1$, which previous work established as a good proxy to the fidelity and the total variation error that is computationally lightweight [40, 29].

Lastly, we investigate the effect of squeezing on MPO EE with our full simulations. We choose to investigate $\gamma = \frac{1}{4}$ for easier simulation. Fig. 3.3 shows an increase in MPO EE with increasing squeezing parameter r. It is important to note that the average number of output photons scales with the average number of input photons N, not the number of squeezed states. This means that for the same number of input squeezed states and β, a higher squeezing parameter has a higher loss. Increasing the average number of photons per squeezed mode from 1 to 2 and 4 only moderately increases the MPO EE compared to increasing N. This observation is similar to the previous finding for Fock state boson sampling: if the number of input modes stays the same and the number of photons per mode increases, the MPO EE grows slowly and can be efficiently simulated [29].

Our numerical findings on the MPO EE growth for different loss scalings have complexity implications, but there is a lack of rigorous correspondence between MPO EE and simulation time. To make the statement on simulation complexity more direct, we validate the bond dimension growth explicitly. This is helpful in particular because the computational complexity is qubic in the bond dimension, both due to SVD and matrix multiplication. We show in Fig. 3.4 the growth of bond dimension in the system size for fixed accuracy of $1 - \mathrm{Tr}(\hat{\rho}) = 0.02$. Previous work has established that $1 - \mathrm{Tr}(\rho)$ is a good proxy for the fidelity [40] and the total variational distance [29], which is the gold standard benchmark for boson sampling sample quality. It is clear that constant loss leads to exponential growth in the bond dimension. In higher loss cases, growth is much more moderate and appears sub-exponential. We also validate that increasing the bond dimension efficiently reduces the simulation error. We choose three experiments and simulated them with different bond dimensions.

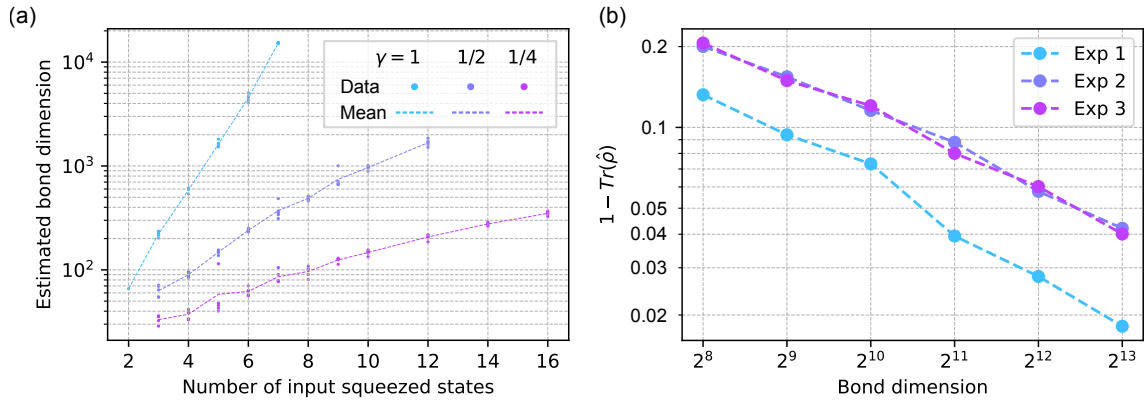

Figure 3.4: Analysis of bond dimension, system size, and error. Details can be found in Methods. (a) Bond dimension needed to reach accuracy $1 - \mathrm{Tr}(\hat{\rho}) = 0.02$ vs. the number of input squeezed modes photon survival scaling $N_{\mathrm{out}} = 0.4N^{\gamma}$ at $r = 0.88$. Dots are individual estimates of the bond dimension obtained from full simulations using $U(1)$ symmetry. Dashed lines are the means. (b) Reduction in $1 - \mathrm{Tr}(\hat{\rho})$ error as bond dimension increases for three different experimental configurations.

CHAPTER 4

SIMULATION OF SUPREMACY-SCALE GAUSSIAN BOSON SAMPLING EXPERIMENTS

This section is heavily based on [35], which is forthcoming in Nature Physics. Minzhao Liu is an equal contribution author, responsible for vectorization of the MPS construction scheme, implementation on GPU and high performance computing platforms, and conducting intermediate to large-scale simulations. He is responsible for testing and improving the time-evolution based MPS construction approach, conceptualizing and implementing the conversion between $U(1)$ symmetric tensor network and regular tensor networks to improve the efficiency, which did not make into the final published approach. He is also responsible for devising but not implementing the method of efficiently estimating the singular values of very large tensor networks, which is used to generate plots of the quantum supremacy frontier.

The results in the previous section is very encouraging to the efficient simulation of lossy GBS using tensor networks, which inspired our subsequent work that uses MPS for simulating lossy GBS. Although we have obtained evidence that under sufficiently high loss, namely $N_{\text{out}} \propto \sqrt{N_{\text{in}}}$, tensor network simulation of GBS is efficient, this algorithm is not capable of simulating supremacy-scale experiments. The photon loss in these experiments are only around 0.4, so the bond dimension required is still very large. Further, we have to model the density operator, which means we need to have a quadratic cost over the cost of representing the state vector. To combat this, we develop a novel algorithm that separates a lossy GBS state into a lossless pure state with much fewer photons and a classical part that contains most of the photons [35]. This means that we can simulate the lossless pure quantum state without the quadratic cost penalty for dealing with the density operator, and we also reduce the number of photons required for the quantum simulation and the bond dimension of the corresponding MPS. As a result, we demonstrate that this decomposition

algorithm achieves higher benchmark scores compared to the largest quantum supremacy experiments.

4.1 The decomposition method

Recall that GBS states with squeezed vacuum states as input followed by an interferometer implementing numerous two-mode mixing operations can be described by a covariance matrix. Let us first consider a highly classical state of light, the thermal state. An example of thermal state is light produced by a light bulb. The emission process is highly random, and there is little preference of one wavelength over the other. Light at different wavelengths oscillate at different frequencies, and it does not make sense to have coherence between different frequencies. This is because the relative position of each wave in their oscillation (at peak or trough) will change rapidly as space and time goes due to the difference in frequencies, so interference between the two waves is unpredictable. Therefore, thermal light can be understood as a classical statistical mixture of coherent light at different frequencies, which means the density operator of a thermal light can be written as a sum of density operators of coherent light. In the quadrature formalism, a coherent state is obtained by displacement on the vacuum, and the Wigner function becomes a displaced minimum envelope Gaussian. Since the Wigner function (as well as the covariance matrix) is linear in the density operator, the thermal state Wigner function is a sum of infinitely many displaced Gaussian envelopes, resulting in a sum that is another Gaussian with a larger spread in the quadrature. Therefore, the covariance matrix of the thermal state is the same as a vacuum except the variance is larger.

To simulate a thermal state, one can start with a vacuum state represented by a tensor network. Since the thermal state is a classical mixture of vacuum with displacements, we can apply random displacements to the vacuum. A random displacement operation can be represented as an operator, which can be applied to the tensor network. Further, displacement

operators are local, so applying displacements to tensor networks is cheap. The further advantage is that the number of photons in the state represented by the tensor network before displacement is smaller, so the correlation in the system that needs to be captured should be smaller, reducing the bond dimension requirement. We should note that simulating sampling for a thermal state is trivial since there is no correlation between optical modes. However, this serves as an illustrative example to convey the intuition of our decomposition scheme.

Let us now consider a general lossy state described by a covariance matrix. How can we separate it into a quantum part represented by a tensor network, and a classical part that can be added back in with random displacement? First, we need to note that the tensor network must represent a physical state, and the corresponding Wigner function must satisfy the uncertainty principle. The great property of Gaussians is that the convolution of two Gaussians is another Gaussian. Therefore, a final large spread Gaussian state Wigner function can be thought of as the convolution between a smaller spread Gaussian satisfying the uncertainty principle and a different Gaussian. The first Gaussian is the state of the tensor network, and the second Gaussian is the probability distribution of the random displacement. Therefore, to find a decomposition that minimize the number of photons, we find the Wigner function that is physical and can be convolved with another Gaussian to yield the final Gaussian.

We now introduce a formal description of the algorithm. First, let us consider a single-mode squeezed state with the following covariance matrix:

$$V = \eta V_0 + (1-\eta)\mathbb{I}_2 = \begin{pmatrix} e^{2s} & 0 \\ 0 & e^{-2s} \end{pmatrix} + \begin{pmatrix} \eta e^{2r} + 1 - \eta - e^{2s} & 0 \\ 0 & 0 \end{pmatrix} \equiv V_p + W \tag{4.1}$$

where V_0 is the covariance matrix of a squeezed state with squeezing parameter r, η is the photon survival probability, and $e^{-2s} \equiv \eta e^{-2r} + (1-\eta)$. The first part V_p decomposition is the same as a squeezed state with a different squeezing parameter s, and the second part W can be understood as classical displacement. This choice of the quantum state for the decomposition actually has the minimum number of photons. First, all eigenvalues of

W must be greater than zero since is describes the covariance matrix of the probability distribution of random displacements. Second, to minimize the number of photons in V_p, the effective squeezing parameter s must be minimized. Together, the two requirements yield the above decomposition. It should be noted that any pure Gaussian state has minimum uncertainty in a sense that:

$$\mathrm{Det}[V] = 2^{-4n}, \tag{4.2}$$

as described in E.q. 2.6 of [15]. Therefore, to minimize the number of photons, we wish to find the Gaussian with the minimal spread and as little variation in the uncertainties in any direction as possible.

More generally, for any Gaussian state on M optical modes, we aim to find the decomposition with the following constraints:

$$\min \mathrm{Tr}[V_p]_{V_p} \text{ with } V - V_p \geq 0, V_p \geq i\mathbf{J}, \tag{4.3}$$

where minimization of the trace minimizes the number of photons, the first constraint ensures physical displacement, and the second constraint ensures a physical covariance matrix. This optimization problem can be easily solved with semidefinite programming. Overall, we illustrate our decomposition scheme in Fig. 4.1.

4.2 Reduction in the number of quantum photons after decomposition

Recent experiments aimed to increase the classical simulation hardness by increasing the number of photons. While increasing the number of input optical modes is difficult, increasing the number of photons per input mode by increasing squeezing is not too difficult. Although hardness does increase, this increase is minor as the previous MPO method suggests that the increase the the MPO EE is minor. We now analyze the effect of high squeezing

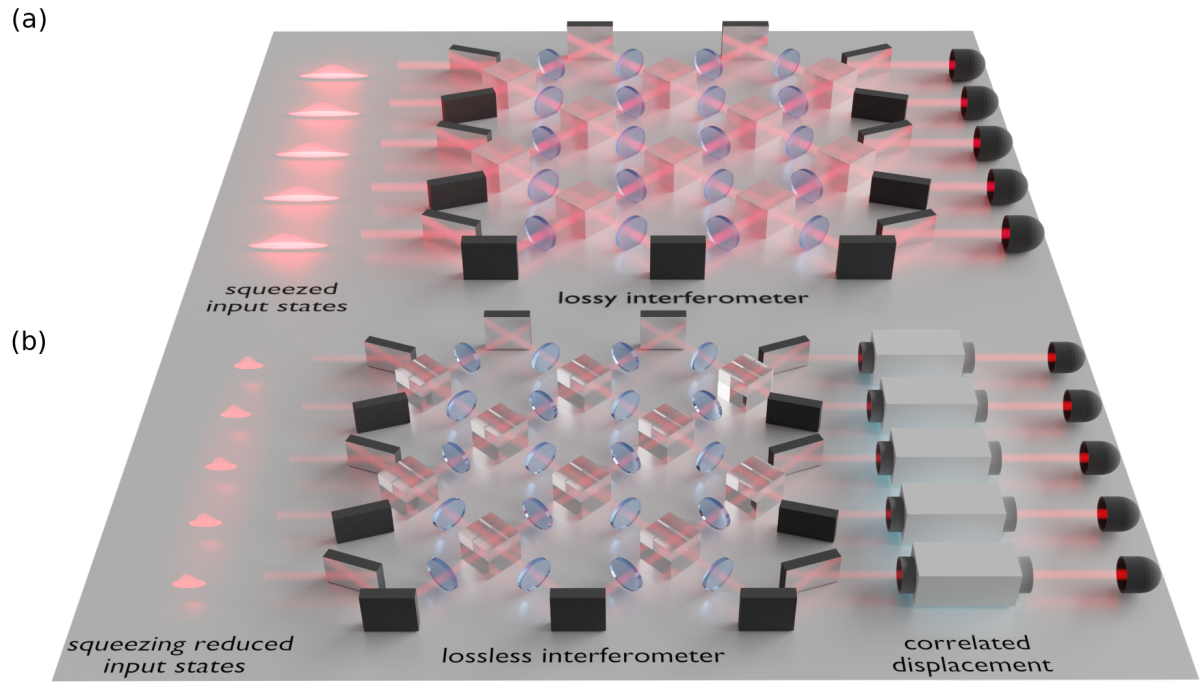

Figure 4.1: (a) Gaussian boson sampling with input squeezed vacuum states that pass through a lossy beam splitter network. (b) Using the decomposition introduced in the main text, we decompose the output state as pure input squeezed states with reduced squeezing, followed by a lossless beam splitter network and Gaussian random displacement channel. Note that the random displacement follows a Gaussian distribution that is generally correlated over different modes.

on the number of photons in the quantum part of our decomposition.

Consider our single mode decomposition the limit where input squeezing r is infinite, which results in single mode effective squeezing

$$s = -1/2 \log(1 - \eta) \tag{4.4}$$

after the decomposition. Therefore, as the transmission η decreases, the effective squeezing strength decreases rapidly, which means that the number of photons is very low. This effect is illustrated in Fig. 4.2. For the actual quantum supremacy experiments, we show the number of photons and number of photons in the quantum part of the decomposition (so called actual squeezed photons) in Table. 4.1.

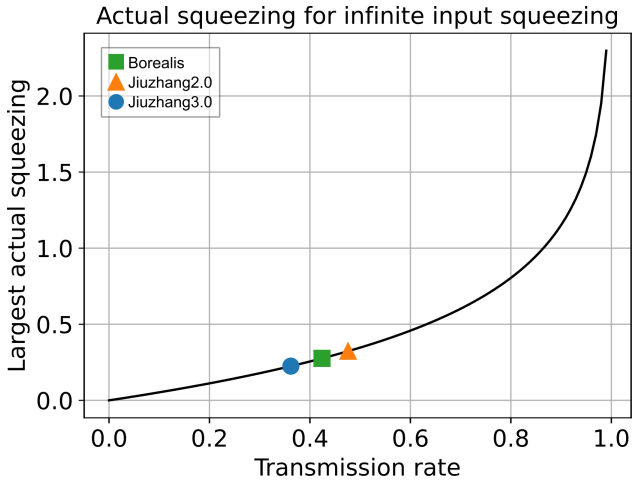

Figure 4.2: Characteristics of the squeezed state V_p from the decomposition for single-mode cases. Actual squeezing parameter and squeezed photon numbers when the input squeezing parameter is infinite. The dots represent the Borealis, Jiuzhang2.0, and Jiuzhang3.0's circuit's transmission rate and their largest actual squeezing and squeezed photons, assuming that infinite input squeezing is used.

4.3 Benchmarking

In order to evaluate the sample quality of the experiment and the simulation, we use several benchmarks. The simulation implementation is discussed in Appendix B.

A sampling task seeks to produce samples x based on an ideal target probability distribution $p(x)$, where producing the outcome x has probability $p(x)$. For boson sampling, this probability distribution is given by the probability of measuring bitstring $\mathbf{n}$ from the quantum device, which is given in E.q. 1.43, 1.46, and 1.51 for the different cases. Note that in the context of benchmarking lossy boson sampling, the ideal probability distribution is computed using $\mathbf{\Sigma}$ that incorporates the effect of photon loss obtained via E.q. 1.53. However, due to various imperfections other than photon loss, the actual experimental sampling process may deviate from the ideal case. One example is partial distinguishability of the photons, where photons that should interfere do not, potentially due to reasons such as spatial mismatch, frequency mismatch, temporal mismatch, etc. The input photon source may also be imperfect due to various reasons. Therefore, the sampler has a different probability

Dataset	Experiment	Input squeezing	Input photons	Trans. rate	Actual squeezing	Output photons	Actual squeezed photons	Ratio
M16	Borealis	0.88-0.89	16.248	0.368	0.14-0.22	5.98	0.549	0.0917
M72	Borealis	0.88-0.89	72.77	0.317	0.093-0.2	23.056	1.74	0.0755
M216 (low)	Borealis	0.52-0.54	67.38	0.321	0.06-0.154	21.622	3.09	0.143
M216 (high)	Borealis	1.09-1.11	388.57	0.324	0.087-0.235	**125.85**	**6.54**	0.052
M288	Borealis	1.00-1.02	407.64	0.362	0.102-0.247	**147.65**	**10.687**	0.0724
M100	Jiuzhang1.0	1.35-1.84	277.64	0.283	0.08-0.26	78.62	1.5	0.019
M144 (P125-1)	Jiuzhang2.0	0.47-0.56	14.593	0.539	0.16-0.255	7.87	2.337	0.297
M144 (P125-2)	Jiuzhang2.0	0.72-0.94	43.09	0.538	0.215-0.342	23.189	4.281	0.1846
M144 (P65-1)	Jiuzhang2.0	0.4-0.545	13.415	0.476	0.124-0.217	6.39	1.628	0.255
M144 (P65-2)	Jiuzhang2.0	0.56-0.76	28.267	0.476	0.154-0.270	13.454	2.53	0.188
M144 (P65-3)	Jiuzhang2.0	0.80-1.08	65.86	0.476	0.18-0.32	31.34	3.62	0.115
M144 (P65-4)	Jiuzhang2.0	1.04-1.41	133.75	0.476	0.20-0.36	63.636	4.385	0.069
M144 (P65-5)	Jiuzhang2.0	1.34-1.81	295.15	0.476	0.212-0.379	**140.38**	**4.965**	0.035
M144 (low)	Jiuzhang3.0	1.14-1.26	113.04	0.424	0.185-0.299	47.93	3.08	0.064
M144 (median)	Jiuzhang3.0	1.33-1.47	183.46	0.424	0.193-0.314	77.80	3.37	0.043
M144 (high)	Jiuzhang3.0	1.49-1.66	274.22	0.424	0.198-0.323	**116.29**	**3.556**	0.031

Table 4.1: Parameters of different Gaussian boson sampling experiments from Refs. [2, 3, 4, 5]. We display the actual squeezing parameters and the actual squeezed photons for each experiment obtained by the optimal decomposition introduced in the main text.

distribution $q(x)$. A classical simulation method such as ours may also include approximation errors, which means that it will also have a slightly different probability distribution. When evaluating if a sampler indeed performs the desired sampling task, one seeks to evaluate the difference between the two distributions. The simplest measure is the total variation distance (TVD):

$$\mathrm{TVD} = \sum_x |p(x) - q(x)|. \tag{4.5}$$

However, evaluating the TVD is very challenging. We first need to obtain the probability distribution $q(x)$. For the experiment, we need to measure sufficiently many samples such that all possible outcomes occur sufficiently many times to estimate $q(x)$ with small error for all x. Then, all exponentially many subtractions and additions need to be performed. For a large experiment with many optical modes, photons, and therefore possible outcomes, this measure is impossible to compute.

One experimentally feasible metric to compute is cross-entropy benchmarking (XEB).

Provided that the experiment produces K samples, forming the set $\{S_i\}_{i=1}^K$ where each S_i is a measurement outcome (could repeat), we can commpute the XEB:

$$\mathrm{XEB}(\{S_i\}_{i=1}^K) = \frac{1}{K}\sum_{i=1}^{K} \log p(S_i). \tag{4.6}$$

This metric is sample efficient since it provides a statistical estimate and does not require the estimation of $q(x)$. Consider the limit where $K \to \infty$, the above expression becomes:

$$\mathrm{XEB}_{K\to\infty} = \sum_x q(x) \log p(x) \tag{4.7}$$

This reveals the reason for the name cross-entropy. A normal entropy is the log average of the probability of a distribution, but here two probability distributions are 'crossed' together.

However, the computation of $p(x)$ for each x is still is exponentially hard. The full distribution $p(x)$ captures the nontrivial correlation between the photon numbers on all output ports. However, we may not need the full correlation to capture the quantum state. In fact, the marginal probability distributions of GBS are far from uniform. Specifically, to compute the marginal probability of a subset of optical modes, we use the expressions in E.q. 1.46 and 1.51, except the only part of the Husimi covariance matrix $\mathbf{\Sigma}$ that corresponds to the subset of modes should be used. We can then compute the k-th order correlation of photon numbers:

$$\kappa(n_1, n_2, \ldots, n_k) = \mathbb{E}(n_1, n_2, \ldots, n_k) - \sum_{p\in P_k} \prod_{b\in p} \kappa[(n_i)_{i\in b}], \tag{4.8}$$

where P_k is the set of all partitions of $\{1, 2, \ldots, k\}$, and expectation values are taken from the ideal distribution or estimated from the samples. If we capture all M orders of correlation, the full probability distribution is captured. However, it is argued that capturing the first few orders of correlation may already capture the true distribution quite well. Specifically,

the correlation vanishes exponentially with k.

In order to benchmark the quantum supremacy experiments and the our simulations of these experiments, we can only calculate the k-th order correlation. However, we do not know that having good agreement on the correlations imply low TVD. Therefore, we use small and intermediate size experiments to construct an argument for the validity of using correlation as a benchmark. With small size experiments, we confirm that low TVD corresponds to high XEB in our algorithm; with intermediate size experiments, we confirm that high XEB corresponds to high agreement with the second-order correlation. Finally, we show our simulation performs better on correlations upto the 5th order for the supremacy experiments.

4.3.1 Small size

We first simulate the small-size experiment from Ref. [4] which has 16 optical modes. Since the experiment's Hilbert space dimension is small, we can compute all the probabilities and the TVD between the probability distributions obtained by samples and the ground-truth distribution. We implement the MPS simulation with different bond dimensions and show the estimated probability distributions in Fig. 4.3 (a) and (b). We now study the relation between TVD and XEB to justify using XEB for larger systems as a proxy of TVD in Fig. 4.3 (c) and (d). The photon number sector refers to conditioning on the measuring different numbers of photons, so the probability distributions we use are normalized by the probability to measure the number of photons in the sector. They clearly show that XEB and TVD are well correlated: when the MPS's TVD is larger than the experiment, its XEB is smaller than the experiment, and vice versa. Using this observation for our cases, we will use the XEB as a proxy of TVD for intermediate scales.

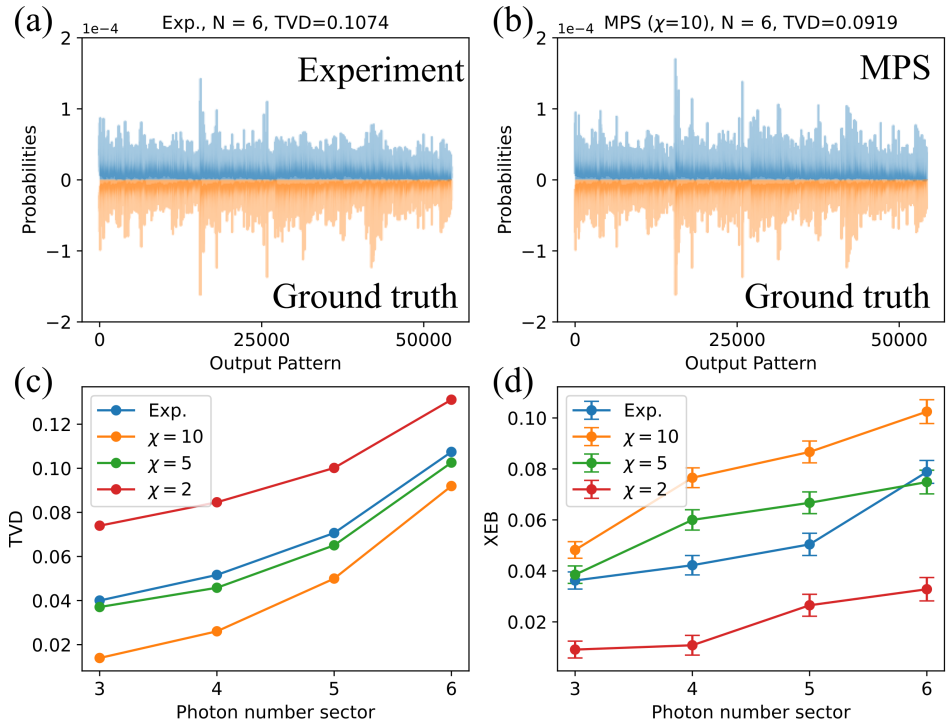

Figure 4.3: (a)(b) Example output probability distributions. (c) The TVD and (d) the XEB for different photon number sectors. Here, for the TVD we use the empirically obtained probability distribution with 10 million samples for each sector, and we use 10,000 samples for XEB for each sector. The error bar is obtained by 1,000 bootstrapping resamples. They clearly show the agreement between the XEB and TVD.

4.3.2 Intermediate size

Unlike the previous small-size case, we can no longer compute the TVD because the number of outcomes is too large. Therefore, based on the observation that the XEB may be a proxy of TVD, we will focus on the XEB. We focus on Borealis's intermediate-scale experiment with $M = 72$. We choose various bond dimensions with local dimension $d = 6$. After sampling, we compute XEB for different photon number sectors, as shown in Fig. 4.4 (a). One can clearly see that, overall, the bond dimensions we chose render larger XEB scores than the experiments do.

We then analyze the second-order correlation functions of all pairs of 72 modes and compare them with the ground-truth values, presented in Fig. 4.4(b). Each data point

corresponds to a possible subset of two optical modes, for which we compute the ideal and sample second-order correlation. If the samples reproduce the ideal correlations well, the data points should be close to the diagonal line with slope 1. We see that as the bond dimension increases, the second-order correlations become closer to the ideal cases. By examining the slopes of linear fits, we can say that the simulation starts to achieve better second-order correlation with χ larger than 160. Further, MPS samples with worse second-order correlations still achieves better XEB than the experiment. This gives us the confidence that if MPS samples have better second-order correlations, the XEB is likely also better than the experiment.

We also analyze other statistical quantities such as Pearson correlation of second-order correlations and two-norm distance of correlations between a sampler and the ground-truth's. As shown in Table 4.2, from $\chi = 160$, the MPS algorithm achieves a larger correlation and smaller distance. The additional quantities may explain the reason that the XEB of the MPS with $\chi = 120$ is better than the experiment, even though the former has a smaller slope in the two-point correlation's linear fit.

4.3.3 Largest scale

We now simulate the largest Borealis, Jiuzhang2.0, and Jiuzhang3.0 experiments, which were used to claim quantum computational advantage. For the benchmark we use kth-order correlation functions because of the computationally large cost for XEB and the fact that they generally agree. Here we choose the bond dimension $\chi = 10000$ for all the cases and the cutoff $d = 4$ for the MPS construction and $d = 10$ for sampling. For second-order correlations, Fig. 4.5 clearly shows that our classical algorithm performs significantly better than the experiments in terms of the slope of the linear fit.

For higher order correlations, we present the Spearman correlation instead of Pearson correlation for consistency with Refs. [3, 5]. Up to the 5th order, the MPS samples' corre-

Dataset	Bond χ	Slope (Exp./MPS)	Correlation (Exp./MPS)	Distance (Exp./MPS)	Truncation error
B-M72	120	**0.877**/0.861	0.977/**0.984**	0.049/0.049	0.048
B-M72	160	0.877/**0.884**	0.977/**0.989**	0.049/**0.043**	0.040
B-M72	200	0.877/**0.899**	0.977/**0.990**	0.049/**0.039**	0.032
B-M72	240	0.877/**0.907**	0.977/**0.991**	0.049/**0.036**	0.026
B-M216-l	600	0.919/**0.935**	0.936/**0.952**	0.021/**0.018**	0.012
B-M216-h	10000	0.935/**0.972**	0.964/**0.980**	0.199/**0.151**	0.006
B-M288	10000	0.887/**0.937**	0.960/**0.970**	0.207/**0.197**	0.017
J2-P65-1	1000	0.943/0.943	0.977/**0.991**	0.007/**0.005**	0.008
J2-P65-2	2000	0.936/**0.939**	0.981/**0.993**	0.015/**0.010**	0.017
J2-P65-3	10000	0.927/**0.968**	0.986/**0.996**	0.030/**0.017**	0.014
J2-P65-4	10000	0.927/**0.972**	0.988/**0.997**	0.048/**0.025**	0.025
J2-P65-5	10000	0.902/**0.980**	0.989/**0.998**	0.067/**0.029**	0.036
J3-high	10000	0.954/**0.982**	0.993/**0.998**	0.048/**0.026**	0.014

Table 4.2: Second-order correlation function benchmarking for different scales of experiments. We present the slope, the Pearson correlation, and the two-norm distance to the ground-truth distribution's second-order correlations. We highlight the better scores.

lations manifestly correlate more with the ground-truth values. For the 6th order, although Jiuzhang3.0's case has a slightly larger correlation than the MPS samples have, the difference still lies within the error bar. Therefore, up to the 6th order, we did not observe a clear advantage from experiments over our classical simulator. We did not conduct the same analysis of higher-order correlations for Borealis experiments because the number of provided samples in Ref. [4] is insufficient for higher-order correlation analysis.

4.4 Measuring the randomness of the experimental unitary

Although Borealis has significantly more optical modes and comparable transmission compared to Jiuzhang, the simulation hardness is not significantly different. It is therefore interesting to understand why Borealis is not as hard to simulate as expected. It turns out that this is due to the limited connectivity of the Borealis device, where all optical modes indirectly interact with each other but not directly interact with each other. Specifically, all

pulses are generated by a single source but are separated in time, forming time-separated optical modes. Interactions between optical modes is accomplished by splitting light into two paths with a beam splitter. Photons in one path goes through, and photons in the other goes through an optical delay loop. This delay allows earlier optical modes to interact with later optical modes. There are three loops with delay time equivalent to the time between 1, 6, and 36 pulses, respectively. As a result, there is non-trivial correlation between all 216 optical modes, but the interaction is limited.

On the other hand, Jiuzhang's connectivity is better, although it is still to generate fully random unitary transformations. A random unitary can be sampled from all unitaries that are valid operators between two Hilbert spaces, and the natural uniform measure in this case is the 'Haar measure', which samples the unitary group $U(M)$ uniformly for an M-mode optical system. If a device has limited connectivity, the entanglement it can generate is similarly limited. Therefore, to study the effect of not having Haar random unitaries on the MPS algorithm bond dimension requirement, we plot the bond dimensions required to achieve different levels of singular value truncation error for the Jiuzhang and Borealis quantum state, as well as the respective states if the unitaries were Haar random. To do this, we compute the singular values of a bipartition of the state, which is easy for Gaussian states with known covariance matrices that can be obtained from the input state and the unitary. Fig. 4.7 shows that Borealis has a very high bond dimension requirement if the unitary was Haar random, but the actual experimental unitary generates low entanglement due tot he limited connectivity and can be simulated with significantly smaller bond dimension. On the other hand, the difference between the experimental and the Haar random bond dimensions for Jiuzhang is very small, indicating that the connecitivity is sufficiently good, at least for the purpose of simulation hardness against our MPS algorithm. Overall, this indicates that simply increasing the number of optical modes is not sufficient, and the connectivity can have a significant impact of the simulation hardness.

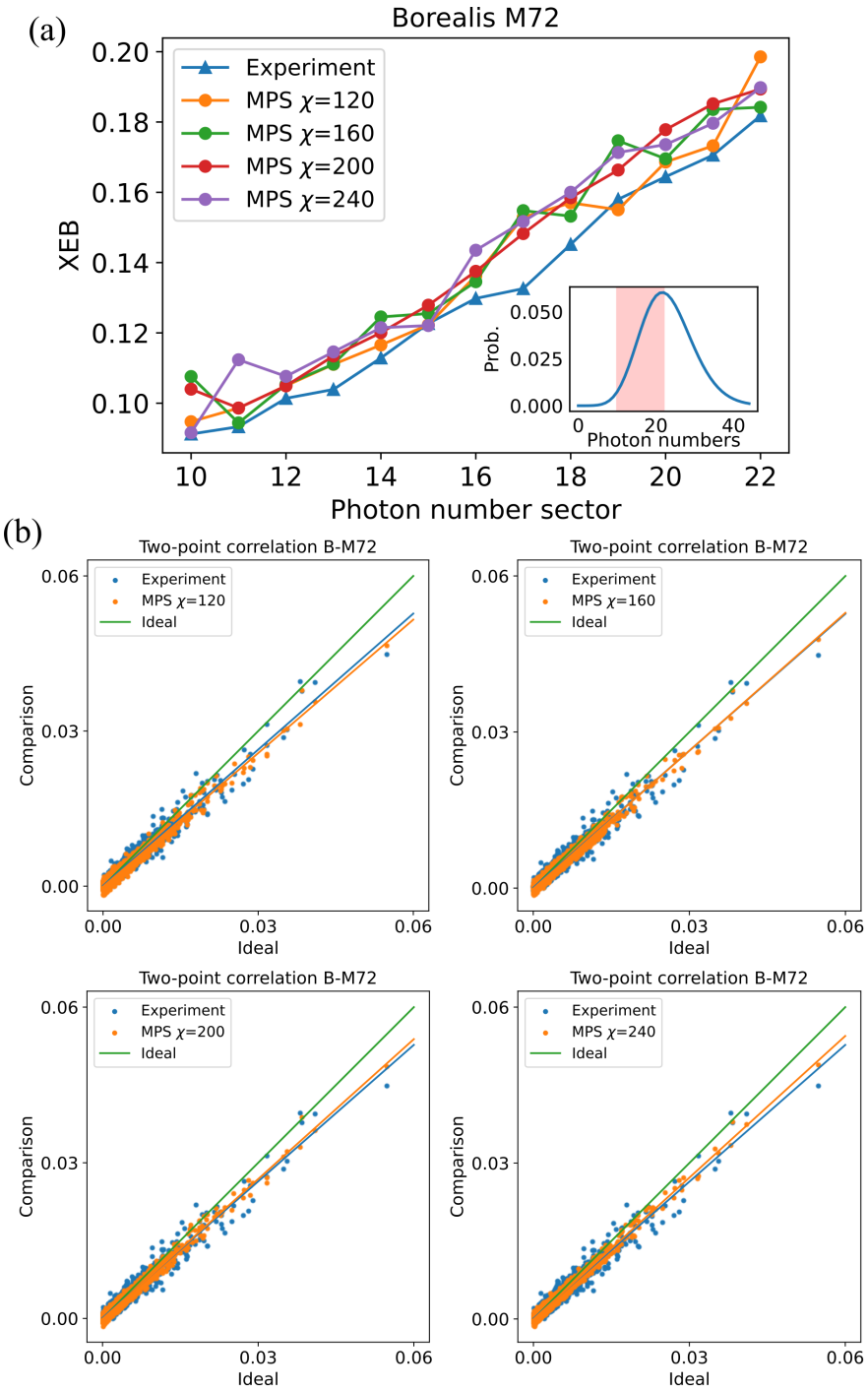

Figure 4.4: Simulation results of Borealis $M = 72$ case with the MPS algorithm. (a) XEB; (b) two-point correlation with different bond dimensions $\chi = 120, 160, 200, 240$. For the two-point correlation function calculation, we have used 1 million samples for all cases. The inset of (a) represents the total photon number distribution, and the shaded region is the sectors we used for XEB.

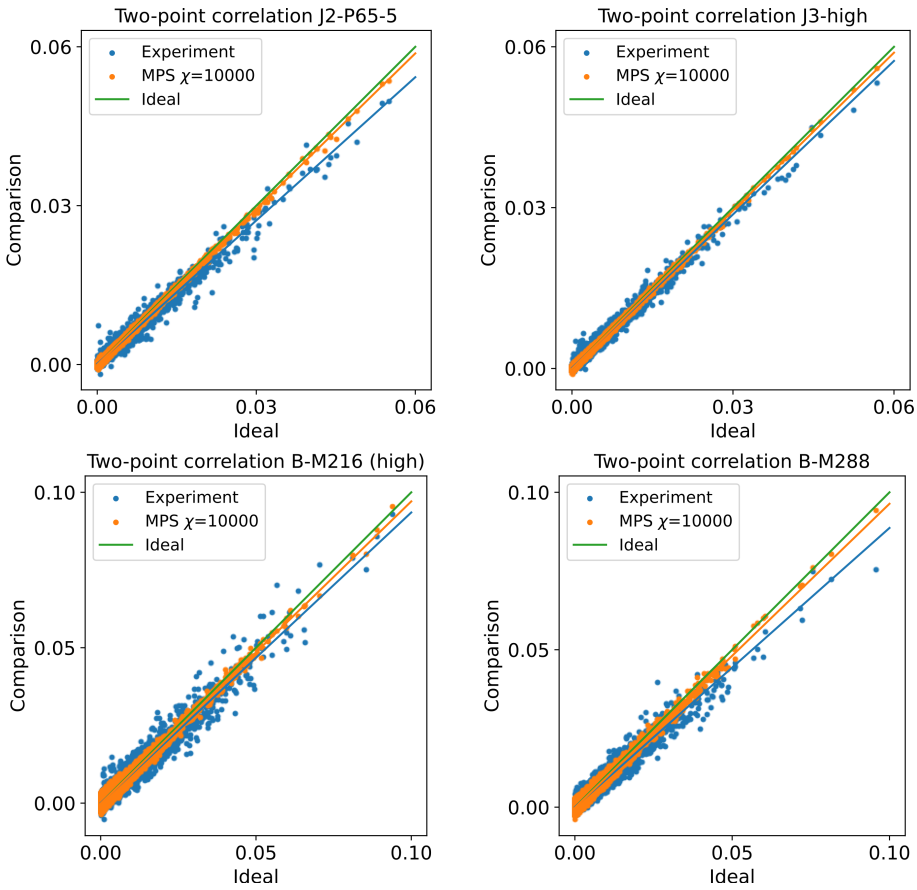

Figure 4.5: Second-order correlation functions of experiments and our MPS sampler for Jiuzhang2.0's P65-5 with $M = 144$, Jiuzhang3.0's high with $M = 144$, and Borealis $M = 216$ (high) and $M = 288$. We use 1 million samples.

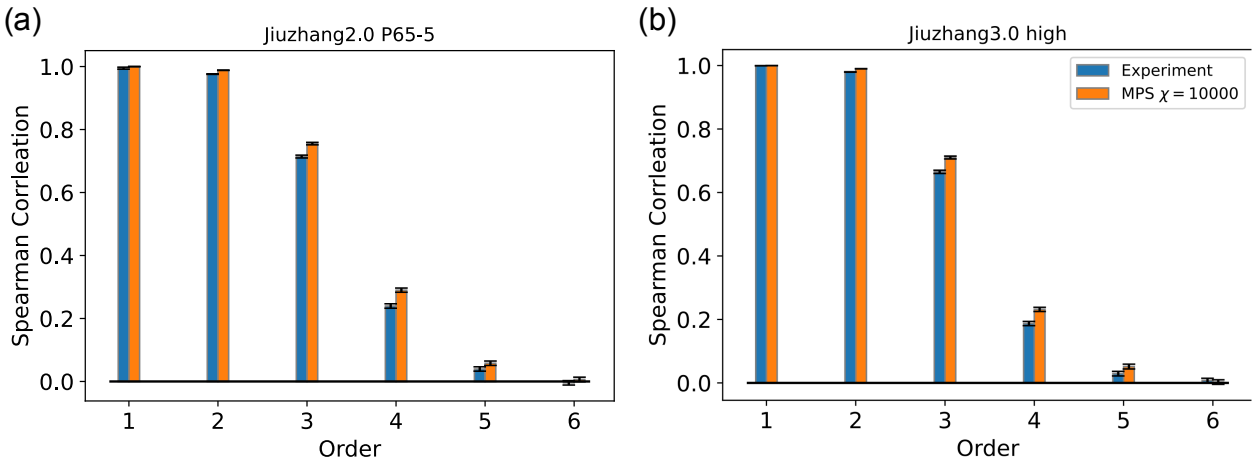

Figure 4.6: Spearman correlation of samples' higher-order correlations to the ground-truth correlations. We use 20 million samples for both samplers; and for each order, up to 20,000 randomly chosen subsets of modes out of $M = 144$ modes were considered. For the first and second orders, we used all subsets. The error bars are the standard deviation obtained by 1,000 bootstrapping resamples.

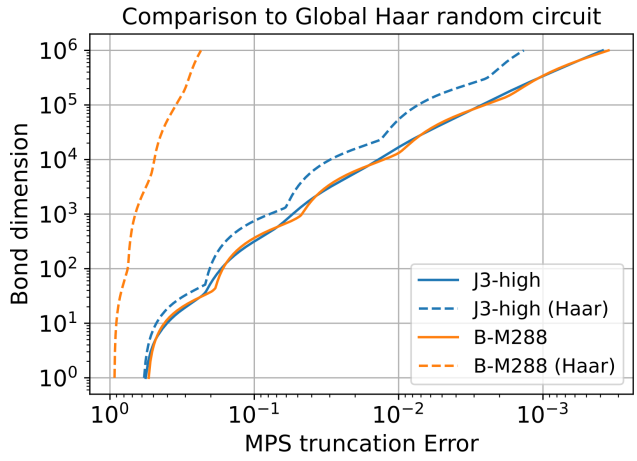

Figure 4.7: Comparison of required bond dimensions from the implemented experiments' circuits (solid curves) and when a global Haar-random circuit is implemented (dashed curves).

CHAPTER 5

COMPLEXITY OF QUANTUM CIRCUITS

This chapter is heavily based on [41]. Minzhao Liu is the primary author, responsible for conceptualizing and implementing the method of estimating the frame potential by Monte-Carlo sampling and contracting tensor networks, and results synthesis and interpretation.

The preceding chapter concluded by emphasizing the importance of sampling from a global Haar-random unitary when performing sampling based quantum supremacy experiment. This is true for both boson sampling and random circuit sampling. If only a small number of gates are applied or if the connectivity is poor, only a small fraction of the Hilbert space of the unitary can be sampled, meaning that the ensemble of circuits is far from Haar random. On the other hand, if the circuits are almost Haar random, it should be the case that synthesizing them generically requires many gates.

Given a random unitary, one may ask the following question: how many quantum gates does it take to construct this unitary? The precise answer depends on the target accuracy, the available gate set, as well as the connectivity of the qubits. However, researchers have been able to reveal some basic properties that are general to these specific considerations.

For an n-qubit system, the Haar measure is the unitary group $U(2^n)$. We know that a generic Haar random unitary requires exponentially many (in n) gates to synthesize. This is easy to see as an exponentially large number of parameters are needed to specify an arbitrary unitary. It is more interesting, therefore, to consider the number of gates required to approximate a random unitary sampled from other ensembles, which one considers as the 'complexity' of the unitary. For example, we can choose a set of rules of generating random quantum circuits, whose corresponding unitaries form the ensemble that we sample from. This set of rules is usually called an 'ansatze', with the gate set, parameters, and qubit connectivies specified. As concrete examples, we introduce three ansatzes that have one-dimensional connectivity: local random unitary ansätze, parallel random unitary ansätze

and hardware efficient ansätze, which are illustrated in Fig. 5.1.

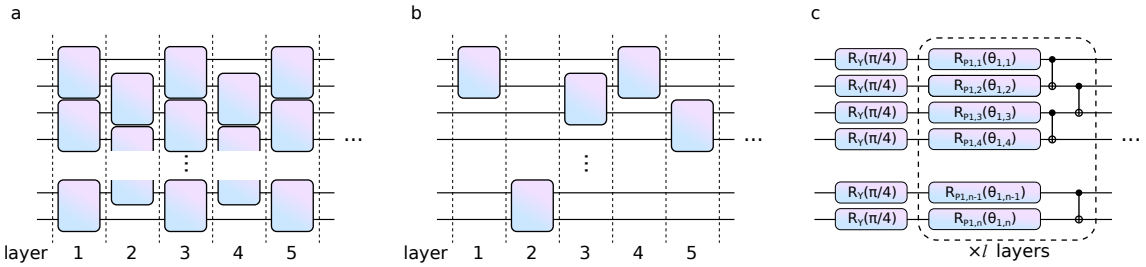

Figure 5.1: Illustration of ansätzes used in this work. All ansätzes assumes 1D nearest-neighbor connectivity. (a) Parallel random unitary ansätze. Each layer is a wall of two-qudit Haar random unitaries on neighboring qudits, and the next layer is offset by 1 qudit. This creates a brickwork motif, and the gate count scales as $O(ln)$. (b) Local random unitary ansätze. Each layer is a single two-qudit Haar random unitary between a pair of randomly chosen neighboring qudits. The gate count scales as $O(l)$. (c) Hardware-efficient ansätze. A wall of $R_Y(\pi/4)$ rotations is followed by alternating layers of random Pauli rotations and controlled-NOT gates, all independently parameterized.

The complexity of unitaries from an ensemble specified by an ansatze captures how chaotic the system is, as well as how much of the Hilbert space can be explored. The smaller the frame potential (which we will introduce shortly), the more chaotic and complicated the ensemble, and the easier it is to achieve quantum advantage [42, 43]. This is of tremendous interest in the area of black-hole physics, variational quantum algorithms, and quantum supremacy experiments.

In the context of holography in high-energy physics, the 'wormhole-growth paradox' apparently violates the anti-de-Sitter space/conformal field theory (AdS/CFT) correspondence, which can be resolved if most unitaries cannot be 'compressed' or represented with a shorter quantum circuit, motivating the 'Brown-Susskind conjecture' [44, 45], which states that the complexity of random quantum circuits grows linearly. In the context of variational quantum algorithms, a quantum state is parameterized by an ansatze whose parameters are optimized such that the quantum state becomes the target state. An ensemble that explores more of the Hilbert space can therefore approximate the target state with higher fidelity, potentially leading to better algorithm success.

In our work, we provide numerical evidence that the complexity grows linearly in the size of the random quantum circuit by computing the so called 'frame potential' [46, 47, 48], given by [49]

$$\mathcal{F}_{\mathcal{E}}^{(k)} = \int_{U,V\in\mathcal{E}} dUdV|\mathrm{Tr}(U^\dagger V)|^{2k}. \tag{5.1}$$

We have the following condition for the ensemble to be an ϵ-approximate k-design:

$$\sqrt{\mathcal{F}_{\mathcal{E}(l)}^{(k)} - \mathcal{F}_{\mathrm{Haar}}^{(k)}} \leq \frac{\epsilon}{q^{nk}}, \tag{5.2}$$

where the ensemble $\mathcal{E}(l)$ depends on the number of layers l. Since there is a linear relationship between k and complexity [50], a long-term open problem is to prove that depth $\mathcal{O}(nk)$ is required to approach approximate k-designs [51, 42, 52, 53, 54, 55, 56, 57, 58, 59, 50]. Assuming an exponentially decreasing frame potential approaching the Haar value, we have

$$\mathcal{F}_{\mathcal{E}(l)}^{(k)} - \mathcal{F}_{\mathrm{Haar}}^{(k)} \propto A^2 e^{-2l/C} \tag{5.3}$$

$$\Rightarrow Ae^{-l/C} \leq \frac{\epsilon}{q^{nk}} \tag{5.4}$$

$$\Rightarrow l \geq C(kn\log q + \log A + \log 1/\epsilon). \tag{5.5}$$

Under this assumption, A and C could still have n and k dependence. Therefore, for there to be linear scaling in n and k, A cannot be exponential, and C must be sublinear.

5.1 Algorithm

The unitary ensembles we are interested in are parameterized by a large number of parameters. Therefore, evaluating the integral is a high-dimensional integration problem, and a numerical Monte Carlo approach is suitable. We approximate the frame potential as the

mean value of the trace,

$$\mathcal{F}_{\mathcal{E}}^{(k)} \approx \frac{1}{N} \sum |\mathrm{Tr}(U^\dagger V)|^{2k},\ U, V \in \mathcal{E}. \tag{5.6}$$

Therefore, we need to evaluate the trace of the sampled unitaries on n target qudits.

A quantum circuit unitary $U = U_1 U_2 U_3 \ldots$ is a tensor $U_{ijk\ldots}^{\alpha\beta\gamma\cdots}$, where i, j, k are input qubit indices and α, β, γ are output qubit indices. The trace of the unitary is

$$\mathrm{Tr}(U) = \sum_{ijk\ldots\alpha\beta\gamma\ldots} U_{ijk\ldots}^{\alpha\beta\gamma\cdots} \delta_{i\alpha}\delta_{j\beta}\delta_{k\gamma}\cdots . \tag{5.7}$$

This is a tensor contraction operation that can be expressed as the tensor network in Fig. 5.2. The circuit shown here is a parallel random unitary circuit with 8 qubits. For efficient contraction, when the number of qubits is large, the contraction order is along the direction indicated in the Fig. 5.2 such that the maximum number of exposed indices in all intermediate tensors is minimum.

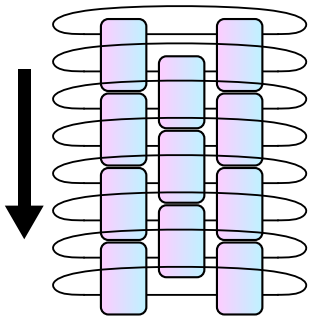

Figure 5.2: Graphical tensor network representation of the trace of a quantum circuit

5.2 Verifying the Brown–Susskind conjecture from frame potentials

Local and parallel random unitaries are commonly discussed in the context of quantum circuit complexity and the Brown-Susskin conjecture. For both ansätzes, the composing random unitaries are drawn from the Haar measure on $U(d^2)$.

5.2.1 Parallel random unitaries

The exponential decay of $\mathcal{F}^{(2)}$ for the parallel random unitary ansätze is given by [59]

$$\mathcal{F}^{(2)} < 2\left(1+\left(\frac{2q}{q^2+1}\right)^{2(l-1)}\right)^{n_g-1}, \tag{5.8}$$

where $n_g = \lfloor n/2 \rfloor$. This is plotted in Fig. 5.3. For fixed ϵ, this leads to a linear scaling of l in n, given by

$$l \geq C(2n\log q + \log n + \log 1/\epsilon), \tag{5.9}$$

where $C = \left(\log \frac{q^2+1}{2q}\right)^{-1}$ is independent of n. We emphasize that linear scaling in n is for fixed ϵ, not fixed $\mathcal{F}$.

Numerical simulation results from Monte Carlo integration of the trace for parallel random unitaries are presented in Figs. 5.4 and 5.5. In Fig. 5.4, The frame potential shows a super-exponential decay in the regime of a few layers and converges to exponential decay as the number of layers increases, just like the theoretical prediction in Fig. 5.3.

To obtain the layer scaling for reaching ϵ-approximate designs, we fit $\mathcal{F}_{\mathcal{E}}^{(k)} - \mathcal{F}_{\text{Haar}}^{(k)}$ to an exponential function according to Eq. 5.3, and l is estimated using Eq. 5.5. Note that our numerical results are in the regime of large ϵ but we are extrapolating to small ϵ values, the validity of which depends on a tightly exponentially decaying $\mathcal{F}$. The number of layers needed to reach $\epsilon < 0.1$ shown in Figs. 5.5. We observe a linear scaling of the number of

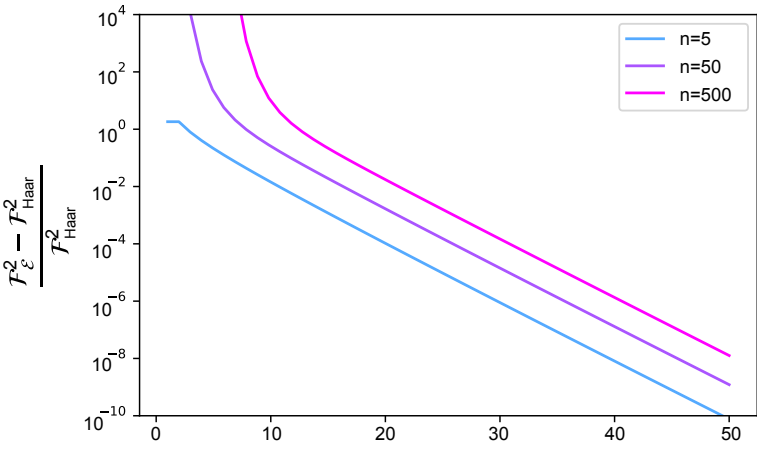

Figure 5.3: Theoretical fractional deviation of the $k = 2$ frame potential from the Haar value as a function of layers for the parallel random unitary ansätze. In this plot, the layer required to reach a fixed $\mathcal{F}$ does not scale linearly with n. The linear scaling is only for fixed ϵ.

needed layers in n, which agrees with the theoretical prediction.

Further, we compare the theoretical predictions in Eq. 5.9 against our numerical findings. Figure 5.5 shows the experimental and fitted k-design layer scaling as a function of the number of qubits. Specifically, we fit a linear curve, ignoring the $\log n$ and the constant $\log 1/\epsilon$ terms. We find a slope of 4.38 in the case of $k = 2$, which is lower than the theoretical value 6.2 as predicted by Brandao *et al.* [59]. We note, however, that the theoretical value gives an upper bound, which accounts for the discrepancy between the theoretical values and the experimental values.

In the inset of Fig. 5.5, we show the slopes of the scaling curves with different k values. It is predicted that there is a linear $O(nk)$ scaling in k for the number of layers l (or $O(n^2k)$ scaling for the circuit size T) needed to approach k-designs [59], and a linear relationship between k and complexity is established in [50]. Together, these findings imply that complexity grows linearly in the circuit size [50, 60]. Our results support the linear scaling of T in k, which predicts that the slope grows linearly in k.

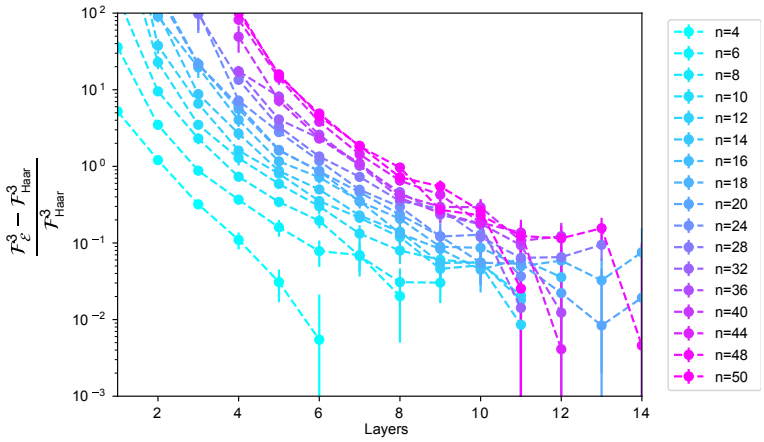

Figure 5.4: Fractional deviation of the $k = 3$ frame potential from the Haar value as a function of layers for the parallel random unitary ansätze. Error bars correspond to the standard errors. As shown in Fig. 5.3, we do not expect linear scaling of l in n with fixed $\mathcal{F}$.

5.2.2 Local random unitaries

Results for local random unitaries are presented in Figs. 5.6 and 5.7. Since each layer in the local random circuit has only one gate, we simulate layers proportional to the number of qubits and plot layers/qubits on the x-axis to maintain a linear scaling. We observe that this layer/qubits ratio scales linearly with the number of qubits. This is the same gate count scaling as the parallel random unitary ansätze, both quadratic in n. The scaling in k is close to linear, but the confidence is lower due to a lack of data points for $k = 4, 5$ at large n.

5.3 Hardware-efficient ansätze as approximate k-designs

Originally proposed as an ansätze for variational quantum eigensolvers [61], hardware-efficient ansätze utilize gates and connectivity readily available on the quantum hardware and are attractive because of their relaxed hardware requirements [62, 63, 64]. In addition, the ansätze are simulated in the context of the barren plateau problem [65], where the variance of gradients vanish exponentially with the number of qubits in sufficiently deep circuits. In fact, the proof of the barren plateau problem assumes that circuits before and after the gate

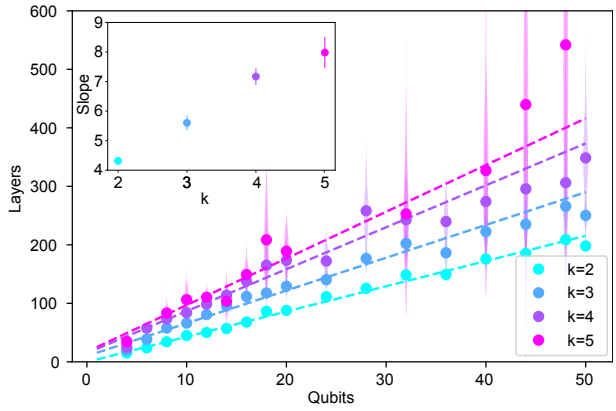

Figure 5.5: Layer scaling as a function of the number of qubits for the parallel random unitary ansätze on a violin plot. Solid points are medians of the bootstrap sample, and the vertical shadows represent the sample distribution where the width corresponds to the density. Dotted lines are linear fits. The inset shows the fitted slopes for different k values.

whose derivative we are computing are approximate 2-designs, which is especially suitable for the hardware efficient ansätze because they are believed to be efficient at scrambling. We simulated these circuits with controlled-phased gates and controlled-not gates as two-qubit gates, respectively. Figure 5.8 shows that the controlled-not gate based ansätzee approaches the Haar measure sooner, and therefore future analysis is conducted on CNOT based ansätze only. Figure 5.9 shows a linear dependence on the number of qubits, as well as a positive dependence on k.

We note that these ansätze reach lower frame potential values with much fewer layers, albeit having much fewer parameters per layer. This result is partially explainable through the observation that each layer in the hardware-efficient ansätze contains two layers of two-qubit gate walls, whereas each layer in the parallel random unitary ansätze contains only one wall. Further, random unitaries from $U(d^2)$ are not all maximally entangling. The hardware-efficient ansätze can therefore generate highly entangled stages much more efficiently, exploring a much larger space with fewer parameters.

Further, unlike the previously discussed ansätze where the frame potential decay rate is constant, the hardware-efficient ansätze decay rate increases with n as shown in the inset

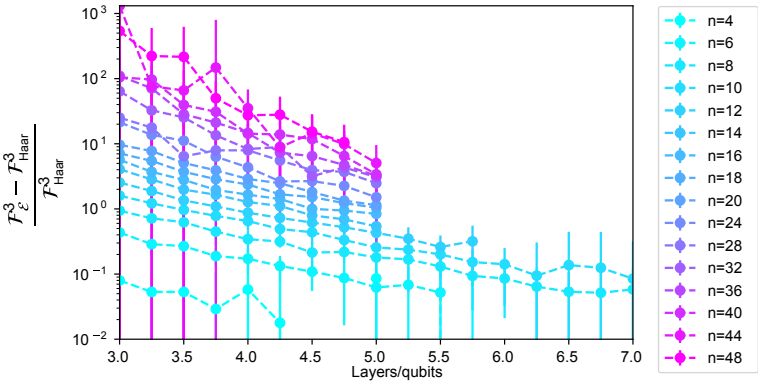

Figure 5.6: Fractional deviation of the $k = 3$ frame potential from the Haar value as a function of layers over the number of qubits for the local random unitary ansätze.

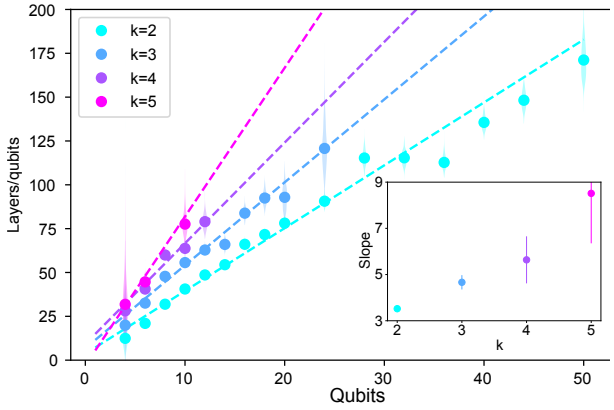

Figure 5.7: Layer/qubits scaling as a function of the number of qubits for the local random unitary ansätze. The inset shows the fitted slopes for different k values.

of Fig. 5.8. This does not contradict the observed linear scaling as long as the decay rate scaling is sublinear.

This observation confirms that hardware-efficient ansätze are highly expressive, a concept that is crucial to the utility of variational quantum algorithms. Ansätze with higher expressibility are able to better represent the Haar distribution and, therefore able to better approximate the target unitary or minimize the objective. This links the expressibility to the frame potential [66]. The high expressibility of hardware-efficient ansätze and their close relatives, and consequently the desirable noise properties due to their shallow depths,

are precisely the argument in favor of these ansätze over their deeper and more complex problem-aware counterparts [67]. With the recent discovery of the relation between expressibility and gradient variance [68], the analysis of frame potentials can play an important role in theoretically and empirically determining the usefulness of various ansätze for variational algorithms.

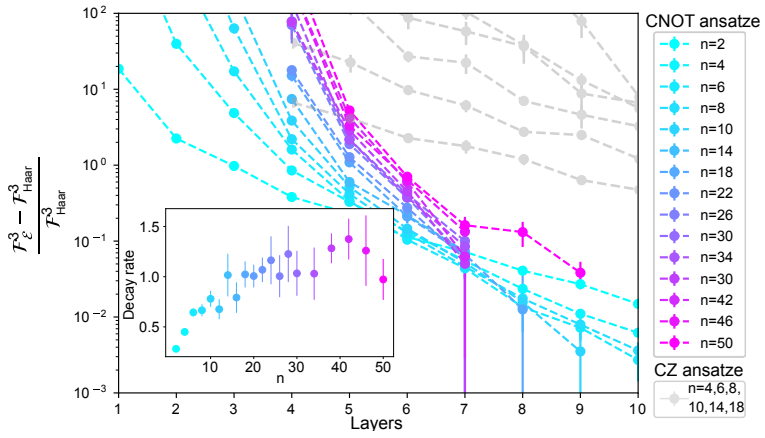

Figure 5.8: Fractional deviation of the $k = 3$ frame potential from the Haar value as a function of layers for the hardware-efficient ansätze. Colorful traces are for the CNOT gate-based ansätze, and gray traces are for the CZ gate-based ansätze. The inset shows the CNOT ansätze decay rate scaling of $\mathcal{F}$ in the number of qubits n.

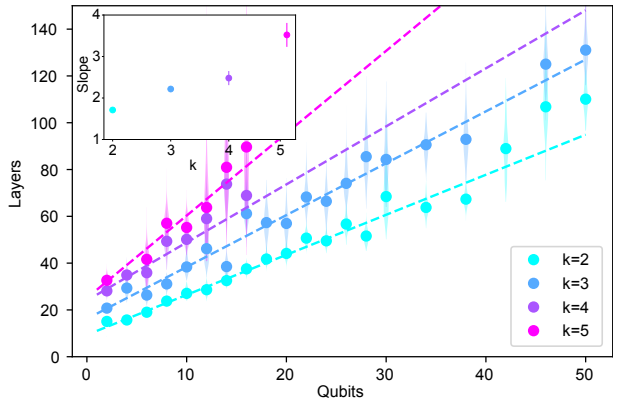

Figure 5.9: Layer scaling as a function of the number of qubits for the hardware-efficient ansätze. The inset shows the fitted slopes for different k values.

CHAPTER 6

CONCLUSIONS

We have given several examples of the ways in which tensor network algorithms can help advancing our understanding of quantum information science. For boson sampling, we show that previous quantum supremacy experiments can be simulated with classical supercomputers with better benchmarking results. This is specifically due to the high photon loss and low connectivity of the experiments, and future improvements should focus on these areas. This result also serves as a motivation to introduce our study on the randomness of random quantum circuit ensembles, which has implications on quantum computational advantage, black hole physics, and variational quantum algorithms. Notably, we find that the complexity of random quantum circuits grows linearly with the circuit depth as conjectured by many.

It is important for us to continue improving tensor networks. In the area of quantum supremacy, tensor networks are the dominant method for efficient simulation [69, 70**?**]. Since the initial 2019 result on quantum supremacy, numerous improvements have reduced the initially claimed classical simulation time [71, 72, 73, 74, 75, 76, 77, 78, 79]. In a most recent work, a six second simulation of the initial experiment is considered possible [69]. This constant tug-of-war between quantum experiments and classical simulations constantly modify the status of claims of quantum computational advantage, which is necessary for the healthy progress in our understanding of related important scientific questions in this area.

On the other hand, it calls for significant caution in making scientific claims as well. Recently, researchers have conducted a large simulation of quantum many-body physics on a quantum computer [80]. Although the authors are very careful in not explicitly claiming quantum computational advantage, this is the center of much of the discussion. Specifically, it is claimed that approximate tensor network methods cannot simulate the experiment on the largest supercomputers. Together with the fact that the paper is on the cover of Nature

and the plethora of media coverage including an article by the New York Times, this has led to a strong impression that quantum computational advantage is claimed. As a result, the subsequent classical simulations of the experiment on a single laptop and similarly small computational resources within merely two weeks of the publication of the experimental article cast serious doubt on the quantumness of the experiment [81, 82, 83].

The use of error mitigation, near-Clifford two-qubit gates, and the locally tree-like qubit connectivity, although perfectly valid and scientifically important to investigate, are not well-understood in terms of their effect on classical simulability. These are exactly the structures that the new classical algorithms exploit to make classical simulation tractable [81, 82, 83]. Further, unlike random circuit sampling where the fidelity is approximate the product of all two-qubit gate fidelities, this experiment has an observable fidelity orders of magnitude higher than the fidelity predicted for a random circuit, indicating that the correlation spread is slow and that the quantumness is low [83]. Overall, this example shows that it is crucial to improve our understanding of classical simulation techniques for us to be able to make accurate and careful statements regarding quantum computational advantage.

APPENDIX A

SUPERCOMPUTING $U(1)$ SYMMETRIC TENSOR NETWORK

An important class of quantum systems have global $U(1)$ symmetry, which arises when the system has some kind of conserved charge [84]. Examples of such systems include the hard-core Bose Hubbard model [85], the spin-1/2 XXZ quantum spin chain [86], boson sampling [25], quantum walk [87, 11, 88, 89, 90], and monitored quantum circuits [91]. A model is said to be $U(1)$ symmetric if the Hamiltonian commutes with the total charge operator [84]

$$[\hat{H}, \hat{N}] = 0. \tag{A.1}$$

As a result, evolution under such Hamiltonian must preserve the charge number operator. More generally, systems can preserve a global $U(1)$ symmetry if the applied unitaries preserve the global charge. In the case of boson sampling, the global charge is the total number of photons, which is preserved under lossless linear optic transformations.

Another example is the hardcore Bose Hubbard model, whose Hamiltonian is given by

$$\hat{H}_{\text{HCBH}} = \sum_{k=1}^{M} (\hat{a}_k^\dagger \hat{a}_{k+1} + \hat{a}_k \hat{a}_{k+1}^\dagger + \gamma \hat{n}_k \hat{n}_{k+1} - \mu \hat{n}_k), \tag{A.2}$$

where $\hat{a}^\dagger, \hat{a}$ are hardcore bosonic creation and annihilation operators, and $\hat{n} = \hat{a}^\dagger \hat{a}$. Since all terms have an equal number of creation and annihilation operators, the total number of particles in the system is preserved. Moreover, the hardcore Bose Hubbard model can be mapped to the spin-1/2 XXZ quantum spin chain by defining

$$\hat{n} = \frac{\mathbb{I} - \hat{\sigma}_z}{2}, \hat{a} = \frac{\hat{\sigma}_x + i\hat{\sigma}_y}{2}, \tag{A.3}$$

which yields the Hamiltonian that preserves the total up spins:

$$\hat{H}_{XXZ} = \sum_{k=1}^{M} (\hat{\sigma}_x^{(k)} \hat{\sigma}_x^{(k+1)} + \hat{\sigma}_y^{(k)} \hat{\sigma}_y^{(k+1)} + \Delta \hat{\sigma}_z^{(k)} \hat{\sigma}_z^{(k+1)}). \tag{A.4}$$

Another example of systems with $U(1)$ symmetry is quantum walk (QW), where particle number is naturally preserved. QW is most commonly discussed in the single-particle discrete time setting, where the system has position and spin degrees of freedom. At each time step, a unitary evolution $U(\theta) = TR(\theta)$ is applied to the system, where

$$T = \sum_x (|x+1\rangle\langle x| \otimes |\uparrow\rangle\langle\uparrow| + |x-1\rangle\langle x| \otimes |\downarrow\rangle\langle\downarrow|) \tag{A.5}$$

$$R(\theta) = \cos\theta(|\uparrow\rangle\langle\uparrow| + |\downarrow\rangle\langle\downarrow|) + \sin\theta(|\downarrow\rangle\langle\uparrow| - |\uparrow\rangle\langle\downarrow|).$$

Notably, the system is shown to be universal for quantum computation [88]. Further, the discrete-time evolution can be viewed as a stroboscopic view of continuous evolution under an effective Hamiltonian such that $U(\theta) = e^{-iH(\theta)\delta t}$ [87]. The system can be modified and generalized to realize numerous classes of topological phases in one and two dimensions. In the multiparticle and continuous time setting, QW has been proposed as a method for quantum sensing [89]. Further, multiparticle quantum walk can be used to explore interacting bosonic and fermionic systems.

Although MPS can efficiently represent many-body systems with controlled entanglement, it does not utilize any symmetry to further reduce the computational cost. To efficiently simulate $U(1)$ symmetric systems, we need to modify the MPS formalism [92, 93, 29].

We denote the total number of particles to the right of position k corresponding to bond

α_k as $c^{[k]}_{\alpha_k}$, then the probability amplitude tensor can be expressed as

$$c_{i_1,\ldots,i_M} = \sum_{\alpha_0,\ldots,\alpha_M=0}^{\chi-1} \Gamma^{[1]}_{\alpha_0\alpha_1}\lambda^{[1]}_{\alpha_1}\Gamma^{[2]}_{\alpha_1\alpha_2}\ldots\lambda^{[M-1]}_{\alpha_{M-1}}\Gamma^{[M]}_{\alpha_{M-1}\alpha_M}\prod_{k=1}^{M}\delta\left(c^{[k-1]}_{\alpha_{k-1}} - c^{[k]}_{\alpha_k} - i_k\right). \tag{A.6}$$

The δ function essentially determines the correct local particle number based on the charge value difference. Updating the wavefunction according to the unitary can be done with the following procedure. We first realize that a local two-site update does not change the charges at $k-1$ or $k+1$, and we can therefore compute the results for different resulting values of $c^{[k]}$. For each chosen value of $c^{[k]}$, $c^{[k-1]} \geq c^{[k]}$ and $c^{[k+1]} \leq c^{[k]}$, we can select a subset of bonds $\alpha_{k-1} \in \mathcal{A}_{k-1}, \alpha_k \in \mathcal{A}_k, \alpha_{k+1} \in \mathcal{A}_{k+1}$ that satisfy the conditions on the three charges. We can then obtain the Θ tensor similar to the normal MPS algorithm:

$$\begin{aligned}\Theta_{\alpha_{k-1}\alpha_{k+1}}(c^{[k]}) = \sum_{\substack{i_k,i_{k+1}=0\\ j_k,j_{k+1}=0}}^{d-1}\sum_{\alpha_k\in\mathcal{A}_k} & U^{i_k,i_{k+1}}_{j_k,j_{k+1}}\lambda^{[k-1]}_{\alpha_{k-1}}\Gamma^{[k]}_{\alpha_{k-1}\alpha_k}\lambda^{[k]}_{\alpha_k}\Gamma^{[k+1]}_{\alpha_k\alpha_{k+1}}\lambda^{[k+1]}_{\alpha_{k+1}} \\ & \times\delta\left(c^{[k-1]}_{\alpha_{k-1}} - c^{[k]}_{\alpha_k} - j_k\right)\delta\left(c^{[k]}_{\alpha_k} - c^{[k+1]}_{\alpha_{k+1}} - j_{k+1}\right) \\ & \times\delta\left(c^{[k-1]}_{\alpha_{k-1}} - c^{[k]} - i_k\right)\delta\left(c^{[k]} - c^{[k+1]}_{\alpha_{k+1}} - i_{k+1}\right),\end{aligned} \tag{A.7}$$

where $0 \leq c^{[k]} \leq N$ and the δ function determines which entry of the unitary matrix to look up. Additionally, examining the $U(1)$ symmetric MPS tells us that the Γ tensors lost their i indices corresponding to the physical degree of freedom (local particle number), reducing the memory complexity by a factor of d. This is instead captured by the size χ 1-d charge tensors c. Second, the size of the Θ matrices that we decompose with SVD is also reduced to at most $\chi \times \chi$ instead of $\chi d \times \chi d$.

We similarly need to compress the new two-site tensor and restore the MPS representation as in the regular algorithm. The full Θ matrix capturing the result of the contraction should be a block-diagonal matrix with $\Theta(c^{[k]})$'s as composing blocks. Therefore, performing SVD

on the full matrix can be achieved by decomposing individual $\Theta(c^{[k]})$, which is the source of the computational complexity reduction of the $U(1)$ symmetric algorithm. Our algorithm performs SVD on all $\Theta(c^{[k]})$'s and keep the χ largest singular values.

State-of-the-art simulations using tensor networks typically employ hardware acceleration, including the use of novel hardware platforms such as graphical processing units (GPUs) [94, 95, 96, 97]. However, symmetry-preserving tensor network algorithms [92, 84, 93] are highly specialized and require data-dependent array entry look up for the unitary matrix. This is an unusual requirement that is not commonly needed in normal tensor operations such as contraction, reshaping, index permutation, etc. As a result, no highly optimized hardware acceleration is readily available for our algorithm. In this work, we aim to bridge this gap in hardware acceleration for the algorithm by optimizing a subroutine on GPU, and also target our implementation to supercomputing resources.

It is hard to improve SVD as it is a well-researched and optimized routine. The naive implementation of computing Θ also requires looping over all possible values of $c^{[k-1]}$ and $c^{[k+1]}$, which introduces an additional $O(d^2)$ complexity compared to SVD. Therefore, we focus our discussion on the Θ computation subroutine and how we optimize it.

A.1 CPU implementation

For a given center charge $c^{[k]}$, the CPU-based implementation loops through all possible left and right charge values $c^{[k-1]}$ and $c^{[k+1]}$ and selects a subset of left and right bonds α_{k-1} and α_{k+1} that satisfy the charge requirement. Since $c^{[k]}$ is fixed for each $\Theta(c^{[k]})$ submatrix, the only term that the delta function affects given the charges is U through $j_k, j_{k+1}, i_k, i_{k+1}$. With the correct unitary matrix value identified, the remaining computation is simply tensor contraction. Each iteration partially fills the $\Theta(c^{[k]})$ matrix at bonds $\alpha_{k-1}, \alpha_{k+1}$. Iterating over all possible left and right charges fills the entire matrix.

For large total particle number d, the $O(d^2)$ complexity due to the nested loop can

significantly increase the computational time. Tensor contraction calculations that would otherwise be parallel has to be broken down into pieces. Therefore, the ability to parallelize across different left and right charges and unitary matrix entries is highly desirable, which is exactly what our GPU algorithm accomplishes. The differences between the CPU and GPU implementations are illustrated in Fig. A.1.

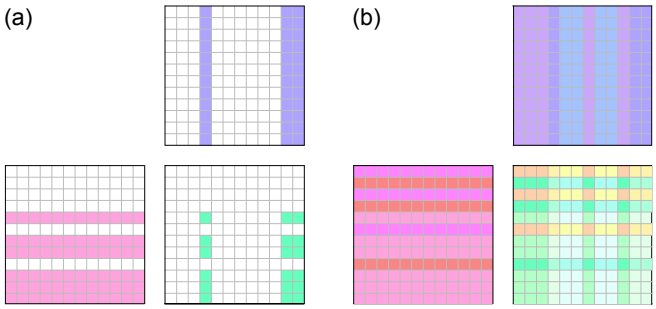

Figure A.1: Algorithms for computing Θ matrices. (a) CPU-based implementation. A subset of bonds are selected from $\Gamma^{[k]}, \Gamma^{[k+1]}$ that have the correct selected charge values $c^{[k-1]}, c^{[k+1]}$. A subset of Θ is computed. (b) GPU-based implementation. All bonds are used and the entire Θ matrix is computed at once.

A.2 Hierarchical GPU implementation

A naive parallel implementation of Θ matrix computation would assign the computation of a single array entry to a single thread. For example, the $\Theta_{i,j}$ can be calculated by a single thread that computes the inner product between the ith row of the first matrix and the jth column of the second matrix. However, this approach has several limitations, and a non-trivial hierarchical algorithm is used in reality for matrix multiplication. For a pedagogical introduction to the hierarchical approach in the context of matrix multiplication, see the work by Kerr *et. al.* [98]

Consider multiplication of $A \in \mathbb{C}^{M\times K}$ and $B \in \mathbb{C}^{K\times N}$. The two matrices are stored in the *global memory* of the GPU, which every thread can access at any time. During inner product calculation of a single thread, the thread needs to read the global memory

$2K$ times to complete the row and column vectors. Computing the whole matrix requires $2MNK$ reads of global memory, which turns out to be a limiting factor. Global memory is physically located far away from the compute cores of the GPU, and only a limited amount of memory can be fetched per second. The naive implementation would actually starve the compute cores due to a lack of data, leaving them idling most of the time.

Alternatively, we can replace element-wise inner products with the accumulation of outer products to reduce the memory read requirement, and Fig. A.2 illustrates the differences between the two approaches. If a whole row/column of A/B is saved in some memory that is closer to the compute cores but have less capacity, all the threads can accumulate $A_{i,k}B_{k,j}$ once with an outer product. This can be repeated K times to complete matrix multiplication. On a GPU, this closer memory is called *shared memory*, which is shared by threads in its thread block of at most 2048 threads. Each thread block has its owon shared memory. Each outer product requires transfer of data from global to shared memory with $M+N$ global reads, and the entire algorithm only needs $K(M+N)$ global reads and $2MNK$ shared memory reads. In reality, since a thread block has a limited number of threads and shared memory, we cannot fit everything in a single block and must compute the entire output matrix by sub-blocks.

The strategy of shifting the need for high memory access from large capacity broad access slow memory to small capacity local access fast memory can be repeated on lower levels. At the lowest level, a single thread actually computes multiple entries of the matrix, where data is stored in *registers* which are the fastest memory available and are private to each thread. Our GPU algorithm for the Θ computation subroutine only differs from matrix multiplication by U and λ value look up. Therefore, our implementation adopts all the techniques mentioned above to maximize performance.

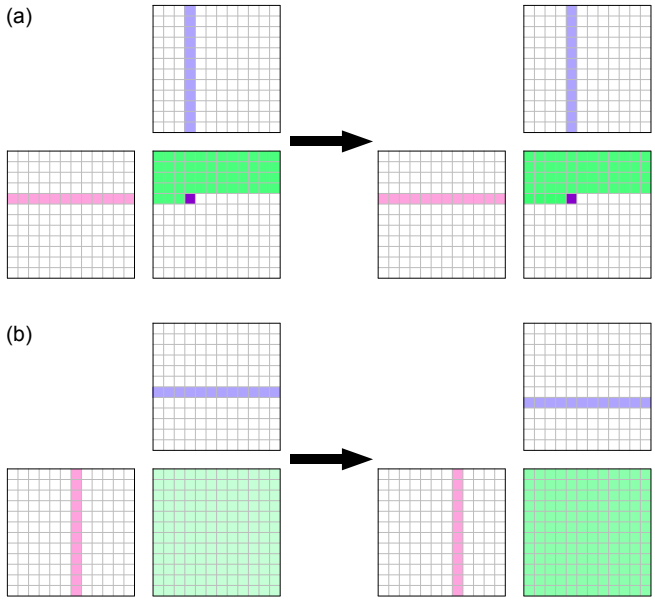

Figure A.2: Matrix multiplication with (a) inner products and (b) outer products.

A.3 Memory alignment in GPU implementation

The charge data-dependent access of U poses difficulties in efficient GPU parallelization. In optimized numerical routines, threads access memory in an *aligned* manner, where consecutive threads access consecutive memory addresses, which allows data to be sent in chunks. Sending data chunks allows multiple units of data to be sent in a single clock cycle, otherwise only one unit of data is sent in a given cycle. In a GPU, this can lead to a 32-fold memory bandwidth reduction. If the charge values are completely unpredictable, the memory address of U that needs to be accessed will not be aligned.

This issue can be easily addressed by sorting the bonds according to the charge values. This leads to aligned memory access as illustrated in Fig. A.3 and significantly improves performance. Additionally, since each thread calculates multiple entries, it might need to access multiple unitary values even after sorting. Due to the limited number of registers available to each thread, we cannot afford to store redundant unitary values. Therefore, we insert empty bonds to ensure that only one value of the unitary matrix corresponding to a

single charge c and physical state i value is stored per thread. This scheme is illustrated in Fig. A.4.

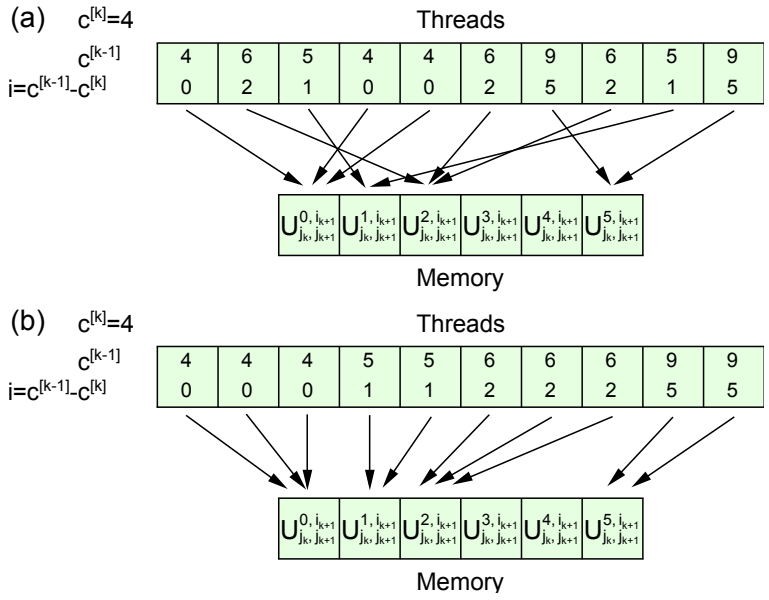

Figure A.3: Memory access pattern (a) without sorting and (b) with sorting.

Additionally, bond indices are sorted such that $c_{\alpha_k}^{[k]}$ only increases as the bond index increases. For small d, this means that $c_{\alpha_k}^{[k]}$ is the same for many consecutive indices. This eliminates the need for threads to look up new U elements, except at boundaries where $c_{\alpha_k}^{[k]}$ changes. This further reduces the need for memory access and reduces latency.

A.4 High-level parallelization

Besides the numerical parallelization of individual SVD and Θ matrix computations through the use of GPUs, additional parallelization is explicitly implemented on the algorithmic level. Further, for systems with large bond dimensions, storing the entire tensor network on a single-GPU or even a single node may become prohibitive. We distribute the storage of individual Γ tensors to different nodes.

First, we parallelize independent two-site unitary updates. A host node identifies all par-

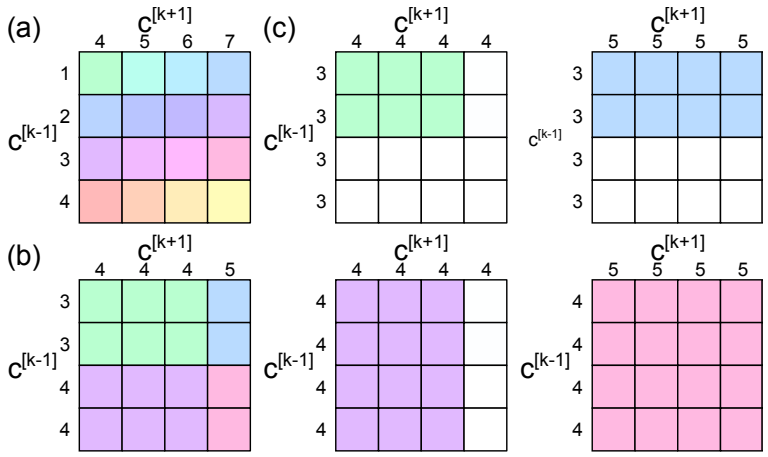

Figure A.4: Illustration of insertion of empty bonds. (a) Worst case scenario of charge value changes within a single fragment without empty bond insertion. The thread has to store 16 values of the unitary. (b) Generic case of a fragment at a charge change boundary. Less than 16 values need to be stored, but this is not known *a-priori* and 16 values of the unitary still needs to be stored. (c) With empty bond insertion, each thread only needs one unitary value.

allel local unitary updates and keeps track of a list of available and busy nodes. Local unitary updates are allocated as soon as a node is available. During allocation, the compute process of the computational node requests the needed Γ, λ, c tensors from the storage processes of the corresponding storage nodes. Similar communication takes place after the computation to update the stored tensors. Second, for a single beam splitter MPO update, the overall Θ matrix is broken up into $\Theta(c^{[k]})$'s, which we compute and decompose in parallel. After the computation node receives the data needed, the data needed for each $\Theta(c^{[k]})$ is distributed to individual GPUs.

With the high-level parallelization discussed above and illustrated in Fig. A.5, the algorithm can be easily scaled to supercomputers with multiple nodes and GPUs, especially when the system size is large. However, there are smaller systems that do not require multi-node parallelization, and we provide implementations with intermediary parallelism as well to avoid the communication overhead of the fully parallel algorithm. On the lowest level, only one GPU is considered, and no distributed memory or computation is used. On the

second level, all memory is managed by a single node, and unitary updates are distributed to individual GPUs instead of nodes.

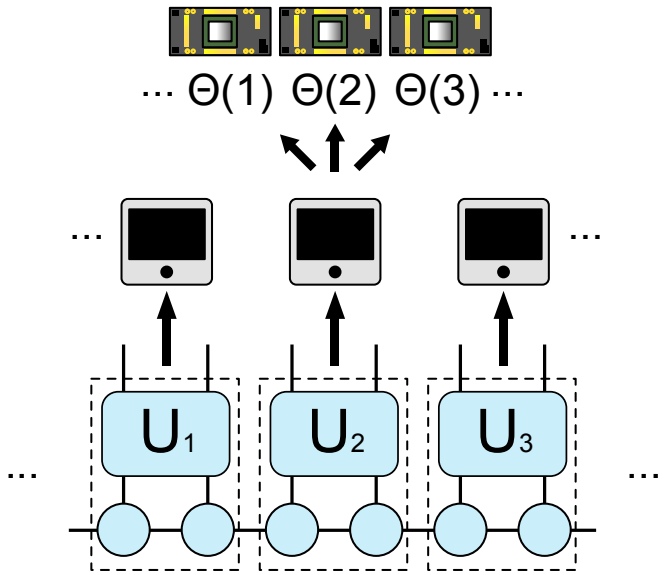

Figure A.5: High-level parallelization. Independent unitary gate updates are distributed to different nodes. Within each unitary update, $\Theta(c^{[k]})$'s are computed and decomposed with SVD independently on different GPUs.

A.5 Run time reduction

We evaluate the performance of our GPU supercomputing algorithm against the CPU-only implementation at the Argonne Leadership Computing Facility (ALCF). All CPU simulations are performed with a single node of the Bebop system with a 2.10 GHz Intel Xeon E5-2695v4 32-core CPU, and GPU simulations are performed on the Polaris system. A single node of the Polaris system has 4 Nvidia A100 GPUs. Table I shows the simulation time in seconds of different implementations for a lossy boson sampling experiment with 12 modes, 10 input squeezed modes, bond dimension 1024 and 8192, photon loss rate 0.55, and local Hilbert space dimension 15. Increasing the bond dimension χ increases the simulation accuracy and time. Moreover, lossy boson sampling requires the density matrix instead

	CPU	single-GPU	One node	Six nodes
$\chi = 1024$	7966	126	60	42
$\chi = 8192$	>259000	2066	1045	322

Table A.1: Simulation time in seconds.

of the state vector, and the generalized algorithm is described in detail in by Oh *et. al.* [29] The consequence of the density matrix generalization is that each charge can take on $15^2 = 225$ values instead of only 15, which means that the CPU-based algorithm needs to perform $225^2 = 50625$ iterations to fill the Θ matrix. On the other hand, our GPU algorithm computes all entries of Θ in parallel.

For the small χ experiment, we ar able to simulate using only CPUs in a reasonable amount of time for comparison. Encouragingly, the single-GPU algorithm achieves a dramatic 63-time speedup even with a single-GPU. We further test the unitary-level parallel algorithm on one node and observe a further two-fold speedup. Lastly, we use the fully parallel algorithm on 6 nodes, observing an additional 43% increase in the computational speed. We observe that the gain in computational speed by switching from less parallelized to highly parallelized implementation is less than the increase in computational resources. Higher-level parallelism incurs significant overhead, which indicates that there is still significant room for optimization.

Fortunately, this payoff in higher parallelism is more pronounced in the setting of larger system sizes. The CPU implementation failed to complete the simulation within the maximum allowed wall time of 72 hours. This means that our single-GPU implementation achieves at least a 125-fold speed up. The computational time is further reduced two-fold when going from the single-GPU implementation to the unitary-level parallel algorithm on one node, similar to the small bond dimension case. However, changing to the fully parallelized algorithm with 6 nodes further reduces the time more than three-fold compared to a fractional reduction in the small bond dimension case. Overall, the fully parallel implementation on six nodes is on the order of a 1000 times faster than the 32-core CPU implementation.

The exact speed up depends on the system size, so we show more experiments with different configurations. All the following experiments are performed with $N = 5, M = 32, \mu = 0.5, r = 0.88$ on a single 32-core CPU or a single A100 GPU. We show the CPU and GPU simulation time for various systems in Fig. A.6a. For system sizes that the CPU can reasonably complete, we observe over 10 times speed up on a single GPU. Further, Fig. A.6b shows that tensor contraction time dominates the overall run time due to the highly inefficient double nested loop. Fig. A.7 shows contributions to the GPU simulation time from subroutines. We see that the contribution from the tensor contraction step is minimal compared to the total simulation time thanks to the efficient custom kernel. Further, SVD takes up the majority of the simulation time, meaning that the overhead of tensor sorting, alignment, and storage is acceptable.

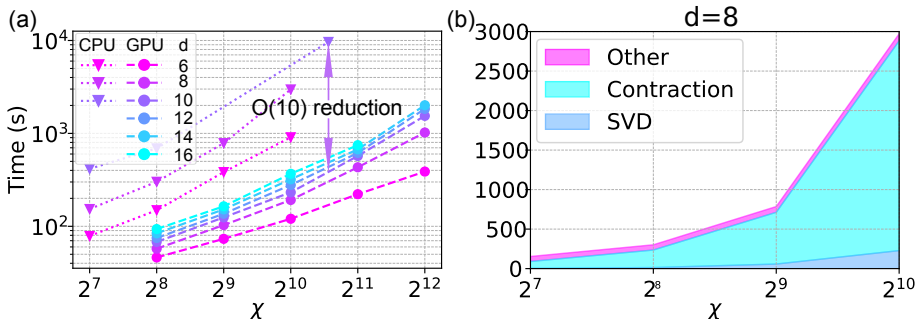

Figure A.6: (a) CPU and GPU simulation time. Traces have higher simulation time in ascending order in d. (b) Contribution to the CPU simulation time from subroutines.

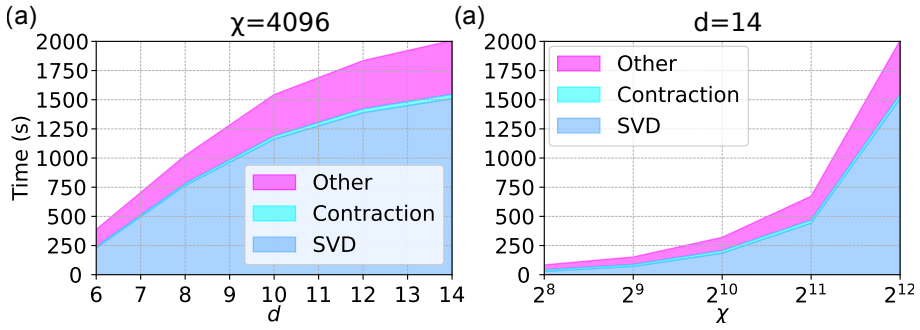

Figure A.7: Contribution to simulation time of the GPU algorithm from subroutines. (a) Bond dimension $\chi = 4096$. (b) Local Hilbert space size $d = 14$.

APPENDIX B

IMPLEMENTATION OF THE MPS ALGORITHM FOR GBS

B.1 MPS construction

We provide the method for MPS construction of Gaussian states based on Ref. [99]. While the method in that reference is typically inefficient, we employ the property of Gaussian states to make it efficient so that we can efficiently find the reduced density matrix of a bipartition $A:B$ and its spectral decomposition because the reduced density matrix is still a Gaussian state and Gaussian states can always be written as a product of thermal states $\hat{\rho}_T$ followed by a Gaussian unitary operation [100]:

$$\hat{\rho}_B = \mathrm{Tr}_A[|\psi\rangle\langle\psi|] = \hat{U}_B\hat{\rho}_T\hat{U}_B^\dagger = \sum_{\boldsymbol{n}=\boldsymbol{0}}^{\infty} p_T(\boldsymbol{n})\hat{U}_B|\boldsymbol{n}\rangle\langle\boldsymbol{n}|\hat{U}_B^\dagger, \tag{B.1}$$

where $p_T(\boldsymbol{n}) = \prod_{i\in B}\bar{n}_i^{n_i}/(\bar{n}_i+1)^{n_i+1}$ and $\bar{n}_i$ is the mean photon number of the ith mode's thermal state. One can easily find $\hat{U}_B$ and $\{\bar{n}_i\}_{i\in B}$ by Williamson decomposition of the covariance matrix of the state in the B part [100]. Hence, the eigenstates of the reduced density matrix are always a number state followed by a Gaussian unitary operation.

Here we recall the method of constructing an MPS proposed in Ref. [99] with adapting Gaussian states' properties. First, we apply the singular value decomposition along the first mode and the rest of the modes with a prechosen bond dimension χ:

$$|\psi\rangle \approx \sum_{\alpha_1=0}^{\chi-1} \lambda_{\alpha_1}^{[1]}|\Phi_{\alpha_1}^{[1]}\rangle|\Phi_{\alpha_1}^{[2\cdots M]}\rangle \tag{B.2}$$

$$= \sum_{n_1=0}^{d-1}\sum_{\alpha_1=0}^{\chi-1} \Gamma_{\alpha_1}^{[1]n_1}\lambda_{\alpha_1}^{[1]}|n_1\rangle|\Phi_{\alpha_1}^{[2\cdots M]}\rangle \tag{B.3}$$

$$= \sum_{n_1=0}^{d-1}\sum_{\alpha_1=0}^{\chi-1} A_{\alpha_1}^{[1]n_1}|n_1\rangle|\Phi_{\alpha_1}^{[2\cdots M]}\rangle, \tag{B.4}$$

where

$$A^{[1]n_1}_{\alpha_1} = \langle n^{[1]}_1 | \langle \Phi^{[2\cdots M]}_{\alpha_1} | \psi \rangle \tag{B.5}$$

$$= \langle n^{[1]}_1 | \langle \boldsymbol{n}^{[2\cdots M]}_{\alpha_1} | \hat{U}^{[2\cdots M]\dagger} \hat{U} | \boldsymbol{n} = \boldsymbol{0} \rangle. \tag{B.6}$$

Here the approximation is due to truncation from the predetermined bond dimension, and the state $|\psi\rangle$ can always be written as $|\psi\rangle = \hat{U}|\boldsymbol{n} = \boldsymbol{0}\rangle$ by Williamson decomposition of a pure Gaussian state; and, as emphasized before, the singular values $\lambda^{[1]}_{\alpha_1}$ can be easily found by performing the Williamson decomposition for the marginal covariance matrix over the bipartition between the first mode and the rest of the modes as in Eq (B.1). Also, the eigenstates $\{|\Phi^{[1]}_{\alpha_1}\rangle\}$, $\{|\Phi^{[2\cdots M]}_{\alpha_1}\rangle\}$ are always photon number states followed by Gaussian unitary operations. Thus, we can characterize the eigenstate $|\Phi^{[2\ldots M]}_{\alpha_1}\rangle = \hat{U}^{[2\cdots M]}|\boldsymbol{n}^{[2\cdots M]}_{\alpha_1}\rangle$ as $\{\hat{U}^{[2\cdots M]}, \boldsymbol{n}^{[2\cdots M]}_{\alpha_1}\}$. We then rewrite it in number basis $|n^{[2]}_2\rangle$ as

$$|\Phi^{[2\cdots M]}_{\alpha_1}\rangle = \sum_{n_2=0}^{d-1} |n^{[2]}_2\rangle |\tau^{[3\cdots M]}_{\alpha_1 n_2}\rangle, \tag{B.7}$$

where

$$|\tau^{[3\cdots M]}_{\alpha_1 n_2}\rangle = \langle n^{[2]}_2 | \Phi^{[2\cdots M]}_{\alpha_1}\rangle, \tag{B.8}$$

where we expand by the eigenstates of the reduced density matrix $\{|\Phi^{[3\cdots M]}_{\alpha_2}\rangle\}^{\chi-1}_{\alpha_2=0}$

$$|\tau^{[3\cdots M]}_{\alpha_1 n_2}\rangle \approx \sum_{\alpha_2=0}^{\chi-1} A^{[2]n_2}_{\alpha_1\alpha_2} |\Phi^{[3\cdots M]}_{\alpha_2}\rangle = \sum_{\alpha_2=0}^{\chi-1} \Gamma^{[2]i_2}_{\alpha_1\alpha_2} \lambda^{[2]}_{\alpha_2} |\Phi^{[3\cdots M]}_{\alpha_2}\rangle \tag{B.9}$$

$$A^{[2]n_2}_{\alpha_1\alpha_2} = \langle n^{[2]}_2 | \langle \Phi^{[3\cdots M]}_{\alpha_2} | \Phi^{[2\cdots M]}_{\alpha_1}\rangle, \tag{B.10}$$

where $A^{[2]n_2}_{\alpha_1\alpha_2} = \Gamma^{[2]n_2}_{\alpha_1\alpha_2}\lambda^{[2]}_{\alpha_2}$, and $|\Phi^{[3\cdots M]}_{\alpha_2}\rangle$ is the eigenstate of the reduced density matrix of

the $[3\cdots M]$ part and $\lambda_{\alpha_2}^{[2]}$ are the singular values, which can be easily identified. Practically, we need only to compute matrices A by

$$A_{\alpha_1\alpha_2}^{[2]n_2} = \langle n_2^{[2]}|\langle \Phi_{\alpha_2}^{[3\cdots M]}|\Phi_{\alpha_1}^{[2\cdots M]}\rangle \tag{B.11}$$

$$= \langle n_2^{[2]}|\langle \boldsymbol{n}_{\alpha_2}^{[3\cdots M]}|\hat{U}^{[3\cdots M]\dagger}\hat{U}^{[2\cdots M]}|\boldsymbol{n}_{\alpha_1}^{[2\cdots M]}\rangle \tag{B.12}$$

$$= \langle n_2^{[2]}|\langle \boldsymbol{n}_{\alpha_2}^{[3\cdots M]}|\hat{V}^{[2\cdots M]}|\boldsymbol{n}_{\alpha_1}^{[2\cdots M]}\rangle, \tag{B.13}$$

where $\hat{V}^{[2\cdots M]} \equiv \hat{U}^{[3\cdots M]\dagger}\hat{U}^{[2\cdots M]}$. By iterating this procedure, we obtain all the matrix elements, which is summarized as

$$A_{\alpha_1}^{[1]n_1} = \langle n_1^{[1]}|\langle \boldsymbol{n}_{\alpha_1}^{[2\cdots M]}|\hat{U}^{[2\cdots M]\dagger}\hat{U}^{[1\cdots M]}|\boldsymbol{n}=\boldsymbol{0}\rangle, \tag{B.14}$$

$$A_{\alpha_{k-1}\alpha_k}^{[k]n_k} = \langle n_k^{[k]}|\langle \boldsymbol{n}_{\alpha_k}^{[(k+1)\cdots M]}|\hat{U}^{[(k+1)\cdots M]\dagger}\hat{U}^{[k\cdots M]}|\boldsymbol{n}_{\alpha_{k-1}}^{[k\cdots M]}\rangle$$

$$\text{for } \ 1<k<M, \tag{B.15}$$

$$A_{\alpha_{M-1}}^{[M]n_M} = \langle n_M^{[M]}|\hat{U}^{[M]}|\boldsymbol{n}_{\alpha_{M-1}}^{[M]}\rangle, \tag{B.16}$$

and

$$\Gamma_{\alpha_1}^{[1]n_1} = A_{\alpha_1}^{[1]n_1}/\lambda_{\alpha_1}^{[1]}, \tag{B.17}$$

$$\Gamma_{\alpha_{k-1}\alpha_k}^{[k]n_k} = A_{\alpha_{k-1}\alpha_k}^{[k]n_k}/\lambda_{\alpha_k}^{[k]} \ \text{ for } \ 1<k<M, \tag{B.18}$$

$$\Gamma_{\alpha_{M-1}}^{[M]n_M} = A_{\alpha_{M-1}}^{[M]n_M}/\lambda_{\alpha_{M-1}}^{[M-1]}. \tag{B.19}$$

Therefore, the remaining calculation to obtain all the matrix elements is $\langle \boldsymbol{n}_1|\hat{V}|\boldsymbol{n}_2\rangle$, for number states $|\boldsymbol{n}_1\rangle$ and $|\boldsymbol{n}_2\rangle$ and a Gaussian unitary operator $\hat{V}$. This quantity has already been studied in Refs. [101, 102] by noting that any Gaussian unitary operation can be decomposed as $\hat{V} = \hat{U}_2\hat{S}(\boldsymbol{r})\hat{U}_1$, where passive unitary operations $\hat{U}_1$ and $\hat{U}_2$ and single-mode

squeezers $\hat{S}(\boldsymbol{r}) = \otimes_i \hat{S}(r_i)$:

$$\langle \boldsymbol{n}_1 | \hat{U}_2 \hat{S}(\boldsymbol{r}) \hat{U}_1 | \boldsymbol{n}_2 \rangle = \frac{\mathrm{haf}(\Sigma_{\boldsymbol{n}_1,\boldsymbol{n}_2})}{\sqrt{\boldsymbol{n}_1! \boldsymbol{n}_2! \prod_i \cosh r_i}}, \tag{B.20}$$

where U_2 and U_1 correspond to the unitary matrix that characterize the unitary operators $\hat{U}_2$ and $\hat{U}_1$, and Σ is a matrix obtained by

$$\Sigma = \begin{pmatrix} U_2 & 0 \\ 0 & U_1^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \tanh \boldsymbol{r} & \mathrm{sech}\ \boldsymbol{r} \\ \mathrm{sech}\ \boldsymbol{r} & -\tanh \boldsymbol{r} \end{pmatrix} \begin{pmatrix} U_2^{\mathrm{T}} & 0 \\ 0 & U_1 \end{pmatrix}, \tag{B.21}$$

where $\Sigma_{\boldsymbol{n}_1,\boldsymbol{n}_2}$ is obtained by repeating Σ's block matrices by $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ times. Hence, the complexity of obtaining all the matrix elements of A, or equivalently Γ and λ, is $O(Md\chi^2 \times (\text{hafnian of } \Sigma_{\boldsymbol{n}_1,\boldsymbol{n}_2}))$, and the complexity of computing the hafnian of $\Sigma_{\boldsymbol{n}_1,\boldsymbol{n}_2}$ is $\tilde{O}(2^{(\boldsymbol{n}_1+\boldsymbol{n}_2)/2})$ [103, 104]. Therefore, two crucial factors determine the complexity: the bond dimension χ and the maximum of $|\boldsymbol{n}_1 + \boldsymbol{n}_2| \equiv \sum_i ((\boldsymbol{n}_1)_i + (\boldsymbol{n}_2)_i)$. Both the bond dimension and the maximum $|\boldsymbol{n}_1 + \boldsymbol{n}_2|$ are affected by the amount of entanglement. The former is evident, and the latter is because for a pure multimode Gaussian state, the reduced state on part B over a bipartition $A : B$ becomes more thermalized when the parties A and B are highly entangled. For example, if they are a product state, the reduced state is still a pure state. Also, we emphasize that the matrix size of $\Sigma_{\boldsymbol{n}_1,\boldsymbol{n}_2}$ is much smaller than the matrix size for computing the output probability of the actual output state, which includes the random displacement. Hence, our MPS construction is, in general, much more efficient than directly sampling from the output state using the best-known classical algorithm [105, 106] when the loss rate is large.

Here, marginal probabilities can be computed as

$$p(m_1,\ldots,m_k|\boldsymbol{\beta}) = \mathrm{Tr}[\hat{D}(\boldsymbol{\beta})|\psi\rangle\langle\psi|\hat{D}^\dagger(\boldsymbol{\beta})|m_1,\ldots,m_k\rangle\langle m_1,\ldots,m_k| \otimes \mathbb{1}] \tag{B.22}$$

$$= \sum_{\alpha_1,\ldots,\alpha_{M-1}=0}^{\chi-1} \sum_{\beta_1,\ldots,\beta_{k-1}=0}^{\chi-1} (\tilde{\Gamma}^{[1]m_1}_{\alpha_1}\lambda^{[1]}_{\alpha_1}\tilde{\Gamma}^{[2]m_2}_{\alpha_1\alpha_2}\cdots\tilde{\Gamma}^{[k]m_k}_{\alpha_{k-1}\alpha_k}) \tag{B.23}$$

$$(\tilde{\Gamma}^{[1]m_1}_{\beta_1}\lambda^{[1]}_{\beta_1}\tilde{\Gamma}^{[2]m_2}_{\beta_1\beta_2}\cdots\tilde{\Gamma}^{[k]m_k}_{\beta_{k-1}\alpha_k})^*(\lambda^{[k]}_{\alpha_k})^2\cdots(\lambda^{[M-1]}_{\alpha_{M-1}})^2$$

$$= \sum_{\alpha_k,\ldots,\alpha_{M-1}=0}^{\chi-1} \left| \sum_{\alpha_1,\ldots,\alpha_{k-1}=0}^{\chi-1} \tilde{\Gamma}^{[1]m_1}_{\alpha_1}\lambda^{[1]}_{\alpha_1}\tilde{\Gamma}^{[2]m_2}_{\alpha_1\alpha_2}\cdots\tilde{\Gamma}^{[k]m_k}_{\alpha_{k-1}\alpha_k} \right|^2 (\lambda^{[k]}_{\alpha_k})^2\cdots(\lambda^{[M-1]}_{\alpha_{M-1}})^2. \tag{B.24}$$

And we iterate this, for example,

$$|\Phi^{[(k+2)\cdots n]}_{\alpha_k}\rangle = \sum_{i_{k+1}=0}^{d-1} |i_{k+1}\rangle|\tau^{[(k+2)\cdots n]}_{\alpha_k i_{k+1}}\rangle, \tag{B.25}$$

where

$$|\tau^{[(k+2)\cdots n]}_{\alpha_k i_{k+1}}\rangle = \sum_{\alpha_{k+1}} A^{[k+1]i_{k+1}}_{\alpha_k\alpha_{k+1}}|\Phi^{[(k+2)\cdots n]}_{\alpha_{k+1}}\rangle \tag{B.26}$$

$$\langle i_n|\hat{U}^{[n]}|\boldsymbol{n}^{[n]}_{\alpha_{n-1}}\rangle. \tag{B.27}$$

To construct the output state's MPS description, we employ the time evolution method. More specifically, we first prepare squeezed vacuum states as input and then apply beam splitters. To do that, for a given unitary $M \times M$ matrix, we decompose it by one-dimensional local beam splitters by using the technique introduced in Ref. [107]. It is worth emphasizing that although we consider one-dimensional systems to employ an MPS, it does not lose generality because M-mode linear-optical systems can always be decomposed into one-dimensional beam splitter networks with $O(M^2)$ depth. Thus, our algorithm is applicable to any $M \times M$

linear-optical circuits. When we apply beam splitters on the MPS, we update the corresponding tensors. During the update, we truncate the singular values by a predetermined bond dimension χ. Hence, depending on the bond dimension, the MPS accumulates a different amount of simulation error. Here, the complexity of the MPS depends on the bond dimension and the local Hilbert space dimension $T = \text{poly}(\chi, d)$.

B.2 Implementation

We implement our simulation algorithm using Python. Specifically, GPU computations are optimized by using the CuPy library, and distributed computation is achieved by using the Message Passing Interface (MPI) using the MPI for Python library. We also develop a custom CUDA kernel for a minor subroutine in CUDA C++, which interfaces with Python through CuPy. Part of the algorithm is taken or modified from the source code of the Strawberry Fields library [108] as well as The Walrus library [109].

The simulation algorithm uses a single GPU for the computation and storage of a single-mode MPS tensor. During the MPS calculation, all tensors are computed independently on different GPUs, which fills tensor entries with appropriately calculated hafnian values. Hafnians of equal-sized square matrices are computed in parallel for numerical efficiency, and this is possible because the hafnian calculation algorithm is data independent. To avoid impractical memory costs, we limit the maximum number of parallel hafnians to a value dependent on the size of the input matrices, and we loop over subbatches to complete all hafnians. After a tensor is completed, the tensor is saved to the local SSD for later use during sampling.

Additionally, since the computational cost for different modes varies dramatically, GPUs that completed the designated tensor calculations are used to accelerate the computation of more costly tensors. Specifically, when challenging tensors have too many parallel hafnians to compute and other GPUs are available, subbatches are sent via the communication fabric

for computation, and the results are later collected.

For the sampling algorithm, the computed single mode MPS tensors are loaded to each GPU. During the sampling procedure, a vector is passed from one mode to the next via the communication fabric after some operations with local tensors for sampling. After the vector is sent to the next GPU, a new tensor is received from the previous GPU for new samples, resulting in a stream of samples that propagate through the chain. This process is also performed in a parallel manner via batch parallel random displacement generation, matrix multiplications, and weighted random choices. For the largest simulations, 100 samples are processed in parallel on a single GPU for numerical efficiency and reasonable memory costs. Therefore, $100 \times M$ samples are processed in parallel on M GPUs.

REFERENCES

[1] Omid Daei, Keivan Navi, and Mariam Zomorodi-Moghadam. Optimized quantum circuit partitioning. *Int. J. Theor. Phys.*, 59:3804–3820, 2020.

[2] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020.

[3] Han-Sen Zhong, Yu-Hao Deng, Jian Qin, Hui Wang, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Dian Wu, Si-Qiu Gong, Hao Su, et al. Phase-programmable Gaussian boson sampling using stimulated squeezed light. *Phys. Rev. Lett.*, 127(18):180502, 2021.

[4] L. S. Madsen et al. Quantum computational advantage with a programmable photonic processor. *Nature*, 606:75–81, June 2022.

[5] Yu-Hao Deng, Yi-Chao Gu, Hua-Liang Liu, Si-Qiu Gong, Hao Su, Zhi-Jiong Zhang, Hao-Yang Tang, Meng-Hao Jia, Jia-Min Xu, Ming-Cheng Chen, et al. Gaussian boson sampling with pseudo-photon-number-resolving detectors and quantum computational advantage. *Phys. Rev. Lett.*, 131:150601, Oct 2023.

[6] Richard P Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, 21(6/7):467–488, 1982.

[7] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. IEEE, 1994.

[8] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery.

[9] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders. Efficient quantum algorithms for simulating sparse hamiltonians. *Commun. Math. Phys.*, 270(270):359–371, 2007.

[10] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse hamiltonians. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, page 283–292, New York, NY, USA, 2014. Association for Computing Machinery.

[11] A.M. Childs. On the relationship between continuous- and discrete-time quantum walk. *Commun. Math. Phys.*, 294(294):581–603, 2010.

[12] Guang Hao Low and Isaac L. Chuang. Optimal hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.*, 118:010501, Jan 2017.

[13] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009.

[14] Yuri Alexeev, Dave Bacon, Kenneth R. Brown, Robert Calderbank, Lincoln D. Carr, Frederic T. Chong, Brian DeMarco, Dirk Englund, Edward Farhi, Bill Fefferman, Alexey V. Gorshkov, Andrew Houck, Jungsang Kim, Shelby Kimmel, Michael Lange, Seth Lloyd, Mikhail D. Lukin, Dmitri Maslov, Peter Maunz, Christopher Monroe, John Preskill, Martin Roetteler, Martin J. Savage, and Jeff Thompson. Quantum computer systems for scientific discovery. *PRX Quantum*, 2:017001, Feb 2021.

[15] Alessandro Ferraro, Stefano Olivares, and Matteo G. A. Paris. Gaussian states in continuous variable quantum information. *Preprint at* *https://arxiv.org/abs/quant-ph/0503237*, 2005.

[16] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.*, 5:4213, 2014.

[17] I. Cong, S. Choi, and M.D. Lukin. Quantum convolutional neural networks. *Nat. Phys.*, 15:1273–1278, 2019.

[18] V. Havlíček, A.D. Córcoles, and K. Temme et al. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567:209–212, 2019.

[19] Minzhao Liu, Junyu Liu, Rui Liu, Henry Makhanov, Danylo Lykov, Anuj Apte, and Yuri Alexeev. Embedding learning in hybrid quantum-classical neural networks. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 79–86, 2022.

[20] Minzhao Liu, Ge Dong, Kyle Gerard Felker, Matthew Otten, Prasanna Balaprakash, William Tang, and Yuri Alexeev. Exploration of quantum machine learning and ai accelerators for fusion science, 2022. ANL/CPS-21/3 172142; TRN: US2302685. USDOE. doi:10.2172/1840522.

[21] Junyu Liu, Minzhao Liu, Jin-Peng Liu, Ziyu Ye, Yunfei Wang, Yuri Alexeev, Jens Eisert, and Liang Jiang. Towards provably efficient quantum algorithms for large-scale machine-learning models. *Nat. Commun.*, 15:434, Jan 2024.

[22] Leonardo Banchi, Mark Fingerhuth, Tomas Babej, Christopher Ing, and Juan Miguel Arrazola. Molecular docking with gaussian boson sampling. *Sci. Adv.*, 6(23):eaax1950, 2020.

[23] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457–3467, Nov 1995.

[24] A Yu Kitaev. Quantum computations: algorithms and error correction. *Russ. Math. Surv.*, 52(6):1191, dec 1997.

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

[26] Craig S Hamilton, Regina Kruse, Linda Sansoni, Sonja Barkhofen, Christine Silberhorn, and Igor Jex. Gaussian boson sampling. *Phys. Rev. Lett.*, 119(17):170501, 2017.

[27] Abhinav Deshpande, Arthur Mehta, Trevor Vincent, Nicolás Quesada, Marcel Hinsche, Marios Ioannou, Lars Madsen, Jonathan Lavoie, Haoyu Qi, Jens Eisert, Dominik Hangleiter, Bill Fefferman, and Ish Dhand. Quantum computational advantage via high-dimensional Gaussian boson sampling. *Sci. Adv.*, 8(1):eabi7894, 2022.

[28] Minzhao Liu, Changhun Oh, Junyu Liu, Liang Jiang, and Yuri Alexeev. Simulating lossy gaussian boson sampling with matrix-product operators. *Phys. Rev. A*, 108:052604, Nov 2023.

[29] Changhun Oh, Kyungjoo Noh, Bill Fefferman, and Liang Jiang. Classical simulation of lossy boson sampling using matrix product operators. *Phys. Rev. A*, 104:022407, Aug 2021.

[30] Roman Orus. A practical itroduction to tensor networks: matrix product states and projected entangled pair states. *Preprint at https://arxiv.org/abs/1306.2164*, 2013.

[31] Tao Xiang, Jizhong Lou, and Zhaobin Su. Two-dimensional algorithm of the density-matrix renormalization group. *Phys. Rev. B*, 64:104414, Aug 2001.

[32] J. Huh, G. G. Guerreschi, B. Peropadre, J. R. McClean, and A. Aspuru-Guzik. Boson sampling for molecular vibronic spectra. *Nat. Photonics*, 9(9):615–620, 2015.

[33] C. S. Wang, J. C. Curtis, B. J. Lester, Y. Zhang, Y. Y. Gao, J. Freeze, V. S. Batista, P. H. Vaccaro, I. L. Chuang, L. Frunzio, et al. Efficient multiphoton sampling of molecular vibronic spectra on a superconducting bosonic processor. *Phys. Rev. X*, 10(2):021060, 2020.

[34] Kamil Brádler, Pierre-Luc Dallaire-Demers, Patrick Rebentrost, Daiqin Su, and Christian Weedbrook. Gaussian boson sampling for perfect matchings of arbitrary graphs. *Phys. Rev. A*, 98:032310, Sep 2018.

[35] Changhun Oh, Minzhao Liu, Yuri Alexeev, Bill Fefferman, and Liang Jiang. Classical algorithm for simulating experimental gaussian boson sampling. *Preprint at https://arxiv.org/abs/2306.03709, Forthcoming in Nat. Phys.*, 2023.

[36] M. Oszmaniec and D. J. Brod. Classical simulation of photonic linear optics with lost particles. *New J. Phys.*, 20(9):092002, 2018.

[37] R. García-Patrón, J. J. Renema, and V. Shchesnovich. Simulating boson sampling in lossy architectures. *Quantum*, 3:169, 2019.

[38] Haoyu Qi, Daniel J. Brod, Nicolás Quesada, and Raúl García-Patrón. Regimes of classical simulability for noisy Gaussian boson sampling. *Phys. Rev. Lett.*, 124:100502, Mar 2020.

[39] Javier Martínez-Cifuentes, K. M. Fonseca-Romero, and Nicolás Quesada. Classical models may be a better explanation of the jiuzhang 1.0 gaussian boson sampler than its targeted squeezed light model. *Quantum*, 7:1076, 2023.

[40] Meng Zhang, Chao Wang, Shaojun Dong, Hao Zhang, Yongjian Han, and Lixin He. Entanglement entropy scaling of noisy random quantum circuits in two dimensions. *Phys. Rev. A*, 106:052430, Nov 2022.

[41] Minzhao Liu, Junyu Liu, Yuri Alexeev, and Liang Jiang. Estimating the randomness of quantum circuit ensembles up to 50 qubits. *npj Quantum Inf.*, 8(137), Nov 2022.

[42] Fernando G. S. L. Brandão and Michal Horodecki. Exponential quantum speed-ups are generic. *Quantum Info. Comput.*, 13(11–12):901–924, nov 2013.

[43] Daniel Harlow and Patrick Hayden. Quantum computation vs. firewalls. *J. High Energy Phys.*, 2013(6):1–56, 2013.

[44] Adam R. Brown and Leonard Susskind. Second law of quantum complexity. *Phys. Rev. D*, 97(8):086015, 2018.

[45] Leonard Susskind. Black holes and complexity classes. *Preprint at* `https://arxiv.org/abs/1802.02175`, 2 2018.

[46] Daniel A. Roberts and Beni Yoshida. Chaos and complexity by design. *J. High Energy Phys.*, 04:121, 2017.

[47] Jordan Cotler, Nicholas Hunter-Jones, Junyu Liu, and Beni Yoshida. Chaos, Complexity, and Random Matrices. *J. High Energy Phys.*, 11:048, 2017.

[48] Junyu Liu. Spectral form factors and late time quantum chaos. *Phys. Rev. D*, 98(8):086026, 2018.

[49] D. Gross, K. Audenaert, and J. Eisert. Evenly distributed unitaries: On the structure of unitary designs. *J. Math. Phys.*, 48(5):052104, 2007.

[50] Fernando G. S. L. Brandão, Wissam Chemissany, Nicholas Hunter-Jones, Richard Kueng, and John Preskill. Models of quantum complexity growth. *PRX Quantum*, 2(3):030316, 2021.

[51] Aram W Harrow and Richard A Low. Random quantum circuits are approximate 2-designs. *Commun. Math. Phys.*, 291(1):257–302, 2009.

[52] Igor Tuche Diniz and Daniel Jonathan. Comment on "random quantum circuits are approximate 2-designs" by AW Harrow and RA Low (Commun. Math. Phys. 291, 257–302 (2009)). *Commun. Math. Phys.*, 304(1):281–293, 2011.

[53] Fernando GSL Brandao, Aram W Harrow, and Michał Horodecki. Local random quantum circuits are approximate polynomial-designs. *Commun. Math. Phys.*, 346(2):397–434, 2016.

[54] Fernando GSL Brandao, Aram W Harrow, and Michał Horodecki. Efficient quantum pseudorandomness. *Phys. Rev. Lett.*, 116(17):170502, 2016.

[55] Aram Harrow and Saeed Mehraban. Approximate unitary t-designs by short random quantum circuits using nearest-neighbor and long-range gates. *Commun. Math. Phys.*, 401:1531–1626, 2023.

[56] Yoshifumi Nakata, Christoph Hirche, Masato Koashi, and Andreas Winter. Efficient quantum pseudorandomness with nearly time-independent hamiltonian dynamics. *Phys. Rev. X*, 7:021006, Apr 2017.

[57] Emilio Onorati, Oliver Buerschaper, Martin Kliesch, Winton Brown, Albert H Werner, and Jens Eisert. Mixing properties of stochastic quantum Hamiltonians. *Commun. Math. Phys.*, 355(3):905–947, 2017.

[58] Nima Lashkari, Douglas Stanford, Matthew Hastings, Tobias Osborne, and Patrick Hayden. Towards the Fast Scrambling Conjecture. *J. High Energy Phys.*, 04:022, 2013.

[59] Nicholas Hunter-Jones. Unitary designs from statistical mechanics in random quantum circuits. *Preprint at* `https://arxiv.org/abs/1905.12053`, 2019.

[60] Jonas Haferkamp, Philippe Faist, Naga BT Kothakonda, Jens Eisert, and Nicole Yunger Halpern. Linear growth of quantum circuit complexity. *Nat. Phys.*, pages 1–5, 2022.

[61] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, 2017.

[62] Michele Grossi, Oriel Kiss, Francesco De Luca, Carlo Zollo, Ian Gremese, and Antonio Mandarino. Finite-size criticality in fully connected spin models on superconducting quantum hardware. *Phys. Rev. E*, 107:024113, Feb 2023.

[63] Kouhei Nakaji and Naoki Yamamoto. Expressibility of the alternating layered ansatz for quantum computation. *Quantum*, 5:434, 2021.

[64] Yuxuan Du, Tao Huang, Shan You, Min-Hsiu Hsieh, and Dacheng Tao. Quantum circuit architecture search for variational quantum algorithms. *npj Quantum Inf.*, 8:62, 2022.

[65] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.*, 9(1):1–6, 2018.

[66] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Adv. Quantum Technol.*, 2(12):1900070, 2019.

[67] Xiaoyuan Liu, Anthony Angone, Ruslan Shaydulin, Ilya Safro, Yuri Alexeev, and Lukasz Cincio. Layer VQE: A variational approach for combinatorial optimization on noisy quantum computers. *IEEE Trans. Quantum Eng.*, 3:1–20, 2022.

[68] Zoë Holmes, Kunal Sharma, M. Cerezo, and Patrick J. Coles. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *PRX Quantum*, 3:010313, Jan 2022.

[69] A. Morvan, B. Villalonga, X. Mi, S. Mandrà, A. Bengtsson, P. V. Klimov, Z. Chen, S. Hong, C. Erickson, I. K. Drozdov, et al. Phase transition in random circuit sampling. *https://arxiv.org/abs/2304.11119*, 2023.

[70] Johnnie Gray and Stefanos Kourtis. Hyper-optimized tensor network contraction. *Quantum*, 5:410, 2021.

[71] Gleb Kalachev, Pavel Panteleev, PengFei Zhou, and Man-Hong Yung. Classical sampling of random quantum circuits with bounded fidelity. *https://arxiv.org/abs/2112.15083*, 2021.

[72] Feng Pan and Pan Zhang. Simulation of quantum circuits using the big-batch tensor network method. *Phys. Rev. Lett.*, 128:030501, Jan 2022.

[73] Feng Pan, Keyang Chen, and Pan Zhang. Solving the sampling problem of the sycamore quantum circuits. *Phys. Rev. Lett.*, 129:090502, Aug 2022.

[74] 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. *https://arxiv.org/abs/2005.06787*, 2020.

[75] Benjamin Villalonga, Sergio Boixo, Bron Nelson, Christopher Henze, Eleanor Rieffel, Rupak Biswas, and Salvatore Mandrà. A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware. *npj Quantum Inf.*, 5(86), 2019.

[76] Gleb Kalachev, Pavel Panteleev, and Man-Hong Yung. Multi-tensor contraction for xeb verification of quantum circuits. *https://arxiv.org/abs/2108.05665*, 2021.

[77] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, and Hartmut Neven. Simulation of low-depth quantum circuits as complex undirected graphical models. *https://arxiv.org/abs/1712.05384*, 2018.

[78] Igor L. Markov, Aneeqa Fatima, Sergei V. Isakov, and Sergio Boixo. Quantum supremacy is both closer and farther than it appears. *https://arxiv.org/abs/1807.10749*, 2018.

[79] Trevor Vincent, Lee J. O'Riordan, Mikhail Andrenkov, Jack Brown, Nathan Killoran, Haoyu Qi, and Ish Dhand. Jet: Fast quantum circuit simulations with parallel task-based tensor-network contraction. *Quantum*, 6:709, 2022.

[80] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout van den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, and Abhinav Kandala. Evidence for the utility of quantum computing before fault tolerance. *Nature*, 601:500–505, 2023.

[81] Tomislav Begušić, Johnnie Gray, and Garnet Kin-Lic Chan. Fast and converged classical simulations of evidence for the utility of quantum computing before fault tolerance. *Science Advances*, 10(3):eadk4321, 2024.

[82] Joseph Tindall, Matthew Fishman, E. Miles Stoudenmire, and Dries Sels. Efficient tensor network simulation of ibm's eagle kicked ising experiment. *PRX Quantum*, 5:010308, Jan 2024.

[83] K. Kechedzhi, S.V. Isakov, S. Mandrà, B. Villalonga, X. Mi, S. Boixo, and V. Smelyanskiy. Effective quantum volume, fidelity and computational cost of noisy quantum processing experiments. *Future Gener. Comput. Syst.*, 153:431–441, Apr 2024.

[84] Sukhwinder Singh, Robert N. C. Pfeifer, and Guifre Vidal. Tensor network states and algorithms in the presence of a global u(1) symmetry. *Phys. Rev. B*, 83:115125, Mar 2011.

[85] Michael Aizenman, Elliott H. Lieb, Robert Seiringer, Jan Philip Solovej, and Jakob Yngvason. Bose-einstein quantum phase transition in an optical lattice model. *Phys. Rev. A*, 70:023612, Aug 2004.

[86] F.C. Alcaraz, U. Grimm, and V. Rittenberg. The xxz heisenberg chain, conformal invariance and the operator content of c $<$ 1 systems. *Nuclear Physics B*, 316(3):735–768, 1989.

[87] Takuya Kitagawa, Mark S. Rudner, Erez Berg, and Eugene Demler. Exploring topological phases with quantum walks. *Phys. Rev. A*, 82:033429, Sep 2010.

[88] Andrew M. Childs, David Gosset, and Zak Webb. Universal computation by multiparticle quantum walk. *Science*, 339(6121):791–794, 2013.

[89] Xiaoming Cai, Hongting Yang, Hai-Long Shi, Chaohong Lee, Natan Andrei, and Xi-Wen Guan. Multiparticle quantum walks and fisher information in one-dimensional lattices. *Phys. Rev. Lett.*, 127:100406, Sep 2021.

[90] Andreas Schreiber, Aurél Gábris, Peter P. Rohde, Kaisa Laiho, Martin Štefaňák, Václav Potoček, Craig Hamilton, Igor Jex, and Christine Silberhorn. A 2d quantum walk simulation of two-particle dynamics. *Science*, 336(6077):55–58, 2012.

[91] Utkarsh Agrawal, Aidan Zabalo, Kun Chen, Justin H. Wilson, Andrew C. Potter, J. H. Pixley, Sarang Gopalakrishnan, and Romain Vasseur. Entanglement and charge-sharpening transitions in u(1) symmetric monitored quantum circuits. *Phys. Rev. X*, 12:041002, Oct 2022.

[92] H.-L. Huang, W.-S. Bao, and C. Guo. Simulating the dynamics of single photons in boson sampling devices with matrix product states. *Phys. Rev. A*, 100(3):032305, 2019.

[93] C. Guo and D. Poletti. Matrix product states with adaptive global symmetries. *Phys. Rev. B*, 100(13):134304, 2019.

[94] Thien Nguyen, Dmitry Lyakh, Eugene Dumitrescu, David Clark, Jeff Larkin, and Alexander McCaskey. Tensor network quantum virtual machine for simulating quantum circuits at exascale. *ACM Transactions on Quantum Computing*, 4(1), oct 2022.

[95] Dmitry I. Lyakh, Thien Nguyen, Daniel Claudino, Eugene Dumitrescu, and Alexander J. McCaskey. Exatn: Scalable gpu-accelerated high-performance processing of general tensor networks at exascale. *Appl. Math. Stat.*, 8, 2022.

[96] Danylo Lykov, Angela Chen, Huaxuan Chen, Kristopher Keipert, Zheng Zhang, Tom Gibbs, and Yuri Alexeev. Performance evaluation and acceleration of the QTensor quantum circuit simulator on GPUs. In *2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS)*, pages 27–34. IEEE, 2021.

[97] Danylo Lykov, Roman Schutski, Alexey Galda, Valeri Vinokur, and Yuri Alexeev. Tensor network quantum simulator with step-dependent parallelization. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 582–593, 2022.

[98] Andrew Kerr, Duane Merrill, Julien Demouth, and John Tran. Cutlass: Fast linear algebra in cuda c++, 2017. `https://developer.nvidia.com/blog/cutlass-linear-algebra-cuda/`. Accessed 19 March 2023.

[99] G. Vidal. Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.*, 91(14):147902, 2003.

[100] Alessio Serafini. *Quantum continuous variables: a primer of theoretical methods*. CRC press, 2017.

[101] Nicolás Quesada. Franck–Condon factors by counting perfect matchings of graphs with loops. *The Journal of Chemical Physics*, 150(16):164113, 2019.

[102] Changhun Oh, Youngrong Lim, Yat Wong, Bill Fefferman, and Liang Jiang. Quantum-inspired classical algorithm for molecular vibronic spectra. *arXiv preprint arXiv:2202.01861*, 2022.

[103] Andreas Björklund. Counting perfect matchings as fast as Ryser. In *Proceedings of the twenty-third annual acm-siam symposium on discrete algorithms*, pages 914–921. SIAM, 2012.

[104] Andreas Björklund, Brajesh Gupt, and Nicolás Quesada. A faster hafnian formula for complex matrices and its benchmarking on a supercomputer. *Journal of Experimental Algorithmics (JEA)*, 24:1–17, 2019.

[105] Jacob FF Bulmer, Bryn A Bell, Rachel S Chadwick, Alex E Jones, Diana Moise, Alessandro Rigazzi, Jan Thorbecke, Utz-Uwe Haus, Thomas Van Vaerenbergh, Raj B Patel, et al. The boundary for quantum advantage in Gaussian boson sampling. *Science Advances*, 8(4):eabl9236, 2022.

[106] Nicolás Quesada, Rachel S Chadwick, Bryn A Bell, Juan Miguel Arrazola, Trevor Vincent, Haoyu Qi, Raúl García, et al. Quadratic speed-up for simulating Gaussian boson sampling. *PRX Quantum*, 3(1):010306, 2022.

[107] William R Clements, Peter C Humphreys, Benjamin J Metcalf, W Steven Kolthammer, and Ian A Walmsley. Optimal design for universal multiport interferometers. *Optica*, 3(12):1460–1465, 2016.

[108] Nathan Killoran, Josh Izaac, Nicolás Quesada, Ville Bergholm, Matthew Amy, and Christian Weedbrook. Strawberry fields: A software platform for photonic quantum computing. *Quantum*, 3:129, 2019.

[109] Brajesh Gupt, Josh Izaac, and Nicolás Quesada. The Walrus: A library for the calculation of hafnians, Hermite polynomials and Gaussian boson sampling. *Journal of Open Source Software*, 4(44):1705, 2019.