

Quantum Computing in a Statistical Context

Yazhen Wang¹ and Hongzhi Liu²

^{1,2} Department of Statistics, University of Wisconsin-Madison, Madison, WI 53706; email: yzwang@stat.wisc.edu

Xxxx. Xxx. Xxx. Xxx. YYYY. AA:1–28

[https://doi.org/10.1146/\(\(please add article doi\)\)](https://doi.org/10.1146/((please add article doi)))

Copyright © YYYY by Annual Reviews.
All rights reserved

Keywords

quantum annealing; quantum Boltzmann machine; quantum computation; quantum computational supremacy; quantum information; quantum statistics

Abstract

Quantum computing is widely considered as a frontier of interdisciplinary research involving fields ranging from computer science to physics and from chemistry to engineering. The stochastic essence of quantum physics renders the random nature of quantum computing, thus there is an important role for statistics to play in the development of quantum computing. On the other hand, quantum computing has great potential to revolutionize computational statistics and data science. This paper provides an overview on the statistical aspect of quantum computing. We review the basic concepts of quantum computing and introduce quantum research topics like quantum annealing and quantum machine learning where statistics is heavily demanded.

Contents

1. INTRODUCTION	2
2. QUANTUM BACKGROUND REVIEW	3
2.1. Mathematical Concepts and Notations	3
2.2. Quantum Physics	4
2.3. Quantum Probability	4
2.4. Quantum Statistics	5
3. QUANTUM COMPUTATION	6
3.1. Quantum Bit and Superposition	6
3.2. Quantum Entanglement	7
3.3. Quantum Algorithms	7
3.4. Quantum Machine Learning	7
4. Quantum Supremacy	8
4.1. RANDOM QUANTUM CIRCUITS	9
4.2. BOSON SAMPLING	13
5. QUANTUM ANNEALING	16
5.1. Classical Ising Model	16
5.2. Quantum Ising Model	17
5.3. Quantum Ising Model in the Transverse Field	18
5.4. Quantum Annealer	19
6. QUANTUM LEARNING WITH BOLTZMANN MACHINES	19
6.1. Boltzmann Machines	19
6.2. Restricted Boltzmann Machines (RBMs)	21
6.3. Quantum Training of Boltzmann Machines	23
6.4. Quantum Boltzmann Machines(QBMs)	23
7. Concluding Remarks	25

1. INTRODUCTION

Quantum information science investigates the preparation and control of the quantum states of physical systems for the purposes of information transmission and manipulation. It consists of quantum communication, quantum computation, and quantum information. There is a wide belief that quantum information science will likely lead to a new wave of technological innovations in communication, computation and information. See [Wang \(2012\)](#), [Wang et al. \(2016\)](#) and [Wang and Song \(2020\)](#) for details. As the crown jewel of quantum information science, quantum computing gains growing interests and tremendous attentions among fields ranging from computer science to physics and from chemistry to engineering. Theoretically it has been proven that quantum computational algorithms can be much faster than the best or optimal classical algorithms for solving certain tough computational problems. Experimentally the Google quantum AI group designed a hard sampling problem for its recently created quantum computer and successfully performed sampling computations on the quantum computer in a computational space of dimension $2^{53} \approx 10^{16}$, which is practically beyond the reach of the fastest classical supercomputers available at the present time (see Section 4.1 and [Arute et al. \(2019\)](#) [Zhong et al. \(2020\)](#) for details). It is featured by many media as a story that a calculation of 3 minutes 20 seconds by a quantum computer would take 10,000 years for the most powerful supercomputer in the

world. Such a task involving a combination of hardware construction, software design, and problem creation and implementation are often termed as quantum (computational) supremacy. As quantum computers of large scale are currently not available to implement faster quantum algorithms for accomplishing difficult computational tasks like breaking cryptosystems that are secure against any classical computer based attack, it is important to demonstrate quantum supremacy and provide experimental evidences to support the (theoretical) claim on the advantage of quantum computation over classical computation. Since quantum physics is essentially stochastic, quantum computation is random in nature. Consequently statistics can play an important role in quantum computation, which in turn has great potential to offer for computational statistics and data science. As our goal in this paper is to provide an overview on the statistical aspect of quantum computation, we introduce the basic concepts of quantum computation and present some selected relevant topics in quantum computing that encounter many statistical issues. Through the overview we illustrate the interplay between statistics and quantum computation. In particular our focus will be on the application of new quantum resources to accomplish statistical computational tasks that are either very slow or infeasible by classical techniques, and the use of quantum approaches that may lead to new theories, methodologies, and computational techniques for statistics and machine learning. We refer to Wang (2012) and Wang and Song (2020) for quantum cryptography topics such as quantum code-breaking algorithms and quantum crypto devices.

The rest of the paper proceeds as follows. Section 2 briefly introduces quantum mechanics and quantum probability and statistics. Section 3 reviews basic concepts in quantum computation and different architectures for quantum computation. Section 4 features two landmark projects on quantum computational supremacy that involve boson sampling and random quantum circuits. Section 5 illustrates quantum annealing and related statistical analysis. Section 6 presents quantum deep learning and describes both classical and quantum approaches with Boltzmann machines. Section 7 provides concluding remarks.

2. QUANTUM BACKGROUND REVIEW

2.1. Mathematical Concepts and Notations

Denote by \mathbb{N} and \mathbb{R} the sets of all integers and real numbers, respectively. We introduce finite-dimensional linear algebra and metric spaces. Denote by \mathbb{C} the set of all complex numbers, and by \mathbb{C}^k the vector space consisting of the set of all k -tuples of complex numbers (z_1, \dots, z_k) . In quantum mechanics and quantum computation we utilize Dirac notations $|\cdot\rangle$ (which is called ket) and $\langle\cdot|$ (which is called bra) to specify that the objects are column vectors or row vectors in the vector space, respectively. We use superscripts $*$, \prime and \dagger to denote the conjugate of a complex number, the transpose of a vector or matrix, and conjugate transpose operation, respectively. We denote by $\langle u|v\rangle$ the inner product of vectors $|u\rangle$ and $|v\rangle$ and adopt a natural inner product for \mathbb{C}^k : $\langle u|v\rangle = \sum_{j=1}^k u_j^* v_j = (u_1^*, \dots, u_k^*)(v_1, \dots, v_k)'$, where $\langle u| = (u_1, \dots, u_k)$ and $|v\rangle = (v_1, \dots, v_k)'$. The inner product induces a norm $\|u\| = \sqrt{\langle u|u\rangle}$, and a distance $\|u - v\|$ between $|u\rangle$ and $|v\rangle$. We say that \mathcal{H} is a finite dimensional Hilbert space if it is a vector space with an inner product.

We call a matrix \mathbf{A} Hermitian (or self-adjoint) if $\mathbf{A} = \mathbf{A}^\dagger$, and a matrix \mathbf{U} unitary if $\mathbf{U}\mathbf{U}^\dagger = \mathbf{U}^\dagger\mathbf{U} = \mathbf{I}$, where \mathbf{I} denotes the identity matrix. A matrix \mathbf{A} is called semi-positive (or positive) definite if $\langle u|\mathbf{A}|u\rangle \geq 0$ for all $|u\rangle \in \mathcal{H}$ (or $\langle u|\mathbf{A}|u\rangle \geq 0$ for all $|u\rangle \in \mathcal{H}$ with equality only for $|u\rangle = 0$). For the matrix $\mathbf{A} = (a_{j\ell})$, denote its trace by $tr(\mathbf{A}) = \sum_{j=1}^k a_{jj}$.

We use \otimes to denote the tensor product operation of vectors or matrices.

2.2. Quantum Physics

A quantum system is characterized by its state and the dynamic evolution of the state. We describe a quantum state by a unit complex vector and its dynamic time evolution by a unitary evolution, where the unitary evolution means that the quantum state changes over time, and these changes are linked by unitary matrices. Furthermore, the dynamic time evolution of the quantum state is governed by a differentiation equation called the Schrödinger equation. To be specific, let $|\psi(t)\rangle$ be the state of the quantum system at time t . The states $|\psi(t)\rangle$ and $|\psi(t+s)\rangle$ at times t and $t+s$, respectively, are linked by $|\psi(t+s)\rangle = \mathbf{U}(s)|\psi(t)\rangle$, where $\mathbf{U}(s) = \exp[-\sqrt{-1}\mathbf{H}s]$ is a unitary matrix, and \mathbf{H} is a Hermitian matrix on \mathbb{C}^d , which is known as the Hamiltonian of the quantum system. The Schrödinger equation governs the continuous-time evolution of $|\psi(t)\rangle$ as follows: $\sqrt{-1}\frac{\partial|\psi(t)\rangle}{\partial t} = \mathbf{H}|\psi(t)\rangle$. Alternatively we may depict a quantum system by a so-called density matrix. The quantum state of a d -dimensional quantum system can be described by a density matrix ρ on the d -dimensional complex space \mathbb{C}^d , where ρ is a positive semi-definite Hermitian matrix with unit trace. We often classify a quantum state as a pure state or an ensemble of pure states. A pure state is a unit vector $|\psi\rangle$ in \mathbb{C}^d with a corresponding density matrix $\rho = |\psi\rangle\langle\psi|$. An ensemble of pure states has a density matrix $\rho = \sum_{j=1}^J p_j |\psi_j\rangle\langle\psi_j|$, which corresponds to the scenario that the quantum system is in one of the states $|\psi_j\rangle$, $j = 1, \dots, J$, with probability p_j being in the state $|\psi_j\rangle$. We may describe the quantum evolution in the density matrix representation as follows: $\rho_{t+s} = \mathbf{U}(t)\rho_s\mathbf{U}^\dagger(t)$, where ρ_s and ρ_{t+s} stand for the density matrix of the state of the quantum system at times s and $t+s$ respectively, and the unitary matrix $\mathbf{U}(s)$ is introduced above. See [Sakurai and Napolitano \(2017\)](#) and [Shankar \(2012\)](#) for details.

2.3. Quantum Probability

Measurements on quantum systems are often through observables like position and momentum, where an *observable* \mathbf{M} is defined as a Hermitian matrix on \mathbb{C}^d . Assume the following eigen-decomposition for \mathbf{M} ,

$$\mathbf{M} = \sum_{a=1}^d x_a \mathbf{Q}_a, \quad 1.$$

where x_a are the real eigenvalues of \mathbf{M} , and \mathbf{Q}_a are the corresponding projections onto the eigen-spaces of \mathbf{M} . Performing a measurement on \mathbf{M} for a quantum system prepared in state ρ , we adopt a measure space (Ω, \mathcal{F}) to describe its possible measurement outcomes, and treat the measurement result X as a random variable on (Ω, \mathcal{F}) with probability distribution P_ρ as follows. The random variable X takes values in $\{x_1, x_2, \dots, x_d\}$, and the probability of obtaining measurement outcome x_a is given by

$$P_\rho(X = x_a) = \text{tr}(\mathbf{Q}_a\rho), \quad a = 1, 2, \dots, d.$$

With the defined probability we derive its expectation and variance,

$$E_\rho(X) = \text{tr}(\mathbf{M}\rho), \quad \text{Var}(X) = \text{tr}(\mathbf{M}^2\rho) - [\text{tr}(\mathbf{M}\rho)]^2.$$

A quantum probability is a noncommutative analog of the Kolmogorov probability. We give a simple illustration using the finite case considered above. The quantum counterparts

of sample space and sigma-field are $\mathcal{H} = \mathbb{C}^d$ and an algebra \mathcal{A} formed by subspaces of \mathcal{H} , respectively. Quantum events are subspaces of \mathcal{H} like eigen-spaces of \mathbf{M} , and observables are quantum analog of random variables. We present a simple quantum probability \mathbb{P} on $(\mathcal{H}, \mathcal{A})$ as follows. For a given subspace \mathbf{Q} of \mathbb{C}^d , define $\mathbb{P}(\mathbf{Q}) = \text{tr}(\mathbf{Q}\rho)$. Then there is a corresponding distributional relationship between \mathbf{M} under \mathbb{P} and random variable X under $P\rho$. In fact, identifying the projection matrix \mathbf{Q}_a with its corresponding eigen-space, we have $\mathbb{P}(\mathbf{Q}_a) = P\rho(X = a)$ with the quantum expectation of observable \mathbf{M} ,

$$\mathbb{E}(\mathbf{M}) = \sum_{a=1}^d a \mathbb{E}(\mathbf{Q}_a) = \sum_{a=1}^d a \mathbb{P}(\mathbf{Q}_a) = \sum_{a=1}^d a P\rho(X = a) = E\rho(X).$$

However, the defined quantum probability $(\mathcal{H}, \mathcal{A}, \mathbb{P})$ is noncommutative. For example, consider the case of $d = 2$ and two noncommutable observables $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Although each observable corresponds to a Bernoulli random variable, there is no quantum analog of the joint distribution for these two observables, which is related to Heisenberg's uncertainty principle and the fact that performing measurements on a quantum system changes its state and thus the resulting probability. The general definition of quantum probability takes \mathcal{H} as a complex Hilbert space, \mathcal{A} a C^* -algebra on \mathcal{H} , and \mathbb{P} a noncommutative probability on $(\mathcal{H}, \mathcal{A})$. For details see [Holevo \(2001\)](#), [Parthasarathy \(2012\)](#) and [Wang \(2012\)](#).

2.4. Quantum Statistics

Statistics is heavily used in quantum theory and quantum experiments in particular quantum computation. For a quantum system we may make statistical inference about the measurement distribution $P\rho$ and thus indirectly about the quantum state ρ based on measurements X_1, \dots, X_n obtained from measuring some observable for the quantum system. That is, X_1, \dots, X_n are i.i.d. observations with distribution $P\rho$, and we can infer ρ based on X_1, \dots, X_n . Assume that ρ is known up to some unknown parameter θ . Then we use $P\rho$ to specify a quantum parametric model and draw statistical inference about θ . We may define quantum likelihood and quantum Fisher information and establish quantum statistical theory such as the quantum Cramér-Rao bound and asymptotic efficient estimation. Specifically, denote by ρ_θ a parametric quantum density matrix family indexed by parameter $\theta = (\theta_1, \dots, \theta_p)'$. Define its quantum score matrices $\ell_j(\theta)$, $j = 1, \dots, p$, to be symmetrized logarithmic derivatives of ρ_θ with respect to θ —that is, $\ell_j(\theta)$ are Hermitian matrices satisfying

$$\frac{\partial \rho_\theta}{\partial \theta_j} = \frac{1}{2} [\ell_j(\theta)\rho_\theta + \rho_\theta \ell_j(\theta)].$$

The quantum Fisher information matrix is given by $\mathbf{J}(\theta) = (J_{ij}(\theta))_{1 \leq i, j \leq p}$, where

$$J_{ij}(\theta) = \frac{1}{2} \text{tr} [\rho_\theta (\ell_i(\theta)\ell_j(\theta) + \ell_j(\theta)\ell_i(\theta))].$$

The quantum Cramér-Rao bound states that for an unbiased estimator of θ , its variance is bounded below by the inverse of $\mathbf{J}(\theta)$. Also we may model ρ nonparametrically and employ nonparametric methods to infer ρ . Quantum information science refers to the reconstruction of ρ as quantum state tomography. For example, consider testing the quantum hypothesis that a quantum state is in a given state ρ_1 against an alternative state ρ_2 . We can establish the quantum analog of the classical theory for the trade-off behavior between two types of

errors. As a case in point, denote by β_n the type II error of the optimal level α test for the quantum hypotheses based on n identical copies of the quantum system. Then we have the quantum Stein lemma: $\lim_{n \rightarrow \infty} [n^{-1} \log \beta_n] = -S(\rho_1 | \rho_2)$, where $S(\rho_1 | \rho_2) = \text{tr}(\rho_1 [\log \rho_1 - \log \rho_2])$ is the quantum relative entropy (or quantum Kullback-Leibler divergence) of ρ_1 and ρ_2 . That is, similar to the classical case, the quantum type II error exponentially decays to zero at a rate determined by the quantum relative entropy. For details see [Artiles et al. \(2005\)](#), [Barndorff-Nielsen et al. \(2003\)](#), [Cai et al. \(2016\)](#), [Holevo \(2001\)](#), [Petz \(2008\)](#) and [Wang and Xu \(2015\)](#).

3. QUANTUM COMPUTATION

3.1. Quantum Bit and Superposition

Bits are the most fundamental concept in classical information and computation science. We encode the information in a bit with two mutually exclusive states, 0 and 1, and may easily realize it by a mechanical switch. The quantum counterpart of the classical bit is the quantum bit, or called qubit for short. Similar to the two state values 0 and 1 for the classical bit, a qubit has states $|0\rangle$ and $|1\rangle$, where the customary notation $|\cdot\rangle$ is used to denote the qubit state. However, there is a key difference between a classical bit and a qubit. Besides states $|0\rangle$ and $|1\rangle$, a qubit may be in superposition states,

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle,$$

where α_0 and α_1 are complex numbers called amplitudes satisfying $|\alpha_0|^2 + |\alpha_1|^2 = 1$. The states of a qubit consist of unit vectors in \mathbb{C}^2 , with the states $|0\rangle$ and $|1\rangle$ forming an orthonormal basis, which are called computational basis states. Unlike the mutually exclusive states for classical bits, the superposition states allow qubits to be both one and zero at the same time.

We may realize qubits in various physical systems. For example, a qubit can be represented by the states of an electron orbiting a single atom. This atom model may treat $|0\rangle$ and $|1\rangle$ as the so-called ‘ground’ and ‘excited’ states of the electron, respectively; if the atom is exposed to light with appropriate energy and for a suitable amount of time, we may move the electron from the $|0\rangle$ state to the $|1\rangle$ state, and vice versa. Moreover, by changing the length of exposure, we can move the electron initially in the state $|0\rangle$ to ‘halfway’ between $|0\rangle$ and $|1\rangle$, say, into the state $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, or the state $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$, where $|+\rangle$ and $|-\rangle$ constitute another qubit basis.

A classical bit can be checked to decide its state being either 0 or 1, however, quantum physics indicates that we cannot examine a qubit to determine its state and find the values of α_0 and α_1 . Qubits have stochastic behaviors that may be described by quantum probability defined in Section 2.3. We can measure a qubit to yield a measurement outcome which is either 0 with probability $|\alpha_0|^2$, or 1 with probability $|\alpha_1|^2$. Moreover, measuring a qubit will change its state.

Similar to the classical case we may define multiple qubits. A b -qubit can be described by \mathbb{C}^{2^b} , and its states are unit vectors in \mathbb{C}^{2^b} with computational basis states $\langle x_1 \cdots x_b |$, $x_i = 0$ or 1 , $i = 1, \dots, b$. A given state can be expressed as a linear combination of the computational basis states with 2^b amplitudes. Quantum exponential complexity refers to the exponential growth in b of dimensionality of the vector space to describe the b -qubit and the number of amplitudes required to specify its superposition states. See [Nielsen and Chuang \(2010\)](#) and [Wang \(2012\)](#) for details.

3.2. Quantum Entanglement

Quantum entanglement refers to the phenomenon that two particles act in the same way as twins that are linked by an unobserved wave to share each other's properties. Consider an entangled 2-qubit system. The quantum entanglement leads to an intriguing feature of its entangled state: performing a measurement on one of the entangled qubits immediately casts the other one into the corresponding perfectly correlated state, which results perfect correlation between the two measurement outcomes for the qubits. For example, take a 2-qubit system in a Bell state

$$|\psi\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}. \quad 2.$$

Performing a measurement on the first qubit of the Bell state $|\psi\rangle$, we obtain a random measurement outcome 0 or 1 with probability 1/2 and 1/2 respectively. However, after the measurement on the first qubit being either 0 or 1, the result of measuring the second qubit in the Bell state $|\psi\rangle$ is always 1 or 0, respectively. That is, there is a perfect correlation between the measurement results of the two qubits in $|\psi\rangle$. We refer to quantum states like the Bell state in 2. as entangled states. The correlation phenomenon is called perfect anti-correlation in entanglement experiments. See [Nielsen and Chuang \(2010\)](#), [Wang \(2012\)](#) and [Wang and Song \(2020\)](#) for more details.

3.3. Quantum Algorithms

The goal of quantum computation is to harness the enormous information hidden in the quantum systems and utilize the exponential power of quantum particles for the purpose of computation. Classical computers are built by electrical circuits comprising wires for transferring information around the circuits and logic gates for accomplishing simple computational tasks. Similarly, quantum computers are created from quantum circuits with quantum gates to perform quantum computation and process quantum information. Despite the similarity, in contrast to classical computation where transistors are used to crunch the ones and zeroes individually, quantum superposition can allow quantum computation to manage both one and zero at the same time and do the trick of carrying out simultaneous calculations. Moreover, the new quantum resources such as quantum superposition and quantum entanglement make it possible for quantum computers to outperform classical computers for certain tough tasks. Already it has been theoretically proved that many quantum algorithms can speed up the best known classical algorithms with examples of quadratic speedup for Grover's search algorithm and exponential speedup for Shor's factoring algorithm. See [Nielsen and Chuang \(2010\)](#) and [Wang \(2012\)](#) for more details.

3.4. Quantum Machine Learning

Quantum learning extends classical machine learning and statistical learning to the quantum realm. It studies how quantum resources can enhance classical learning in terms of computational complexity and statistical efficiency. Quantum computers can be faster than classical computers for solving certain machine learning problems, and it is possible for quantum learners to achieve higher statistical efficiency for some particular learning tasks, albeit there are caveats regarding quantum state preparation in some quantum machine learning algorithms. Examples include support vector machines, principal component analysis, and Boltzmann machines.

A case in point is quantum reinforcement learning. Classical reinforcement learning studies the problem of learning in and from interactive task environments, where a task environment is specified by a Markov decision process (MDP) through its states that are observed by an agent. The agent takes actions to produce transitions from states to states, and transitions are rated with rewards. The agent needs to learn which actions to perform in order to maximize the rewards. In reinforcement learning, the environments are unknown in the sense of unknown transition rules of MDP, and its goal is to learn how to find optimal policies for achieving the maximum rewards, where a policy refers to a behavior rule to select actions based on states. Various procedures and algorithms are developed to estimate the so-called value functions and value-action functions and find optimal policies based on the estimated functions. See [Sutton and Barto \(2018\)](#) for more details.

We may consider a quantum approach to learning via interaction and establish a quantum framework for agents, environments and their interactions, namely, a quantum agent-environment paradigm for quantum reinforcement learning. The quantum paradigm has the potential to lead to enhancements in both computational complexity and statistical efficiency of classical reinforcement learning. Moreover, we may mix the classical and quantum approaches for reinforcement learning frameworks. Depending on whether the agent or the environment are classical or quantum, we may obtain four agent-environment settings: classical agent & classical environment (CC), classical agent & quantum environment (CQ), quantum agent & classical environment (QC), and quantum agent & quantum environment (QQ) frameworks. The classification loosely corresponds to placing (classical) machine learning in CC, applications of machine learning to control quantum systems in CQ, quantum speed-ups in machine learning algorithms (like quantum annealers with non-quantum data) in QC, and quantum machine learning/reinforcement learning with quantum agents, quantum environments, and quantum data in QQ.

Similar to quantum speed-ups in the case of supervised and unsupervised machine learning, quantum algorithms based on quantum walks and quantum Markov chains lead to provable quantum speed-ups in reinforcement learning. Like online learning, computational complexity and statistical efficiency may be closely connected in the context of reinforcement learning. Consider a reinforcement learning setting where an interaction is happening in respect to some external real time, and the environment alters regarding the real time. For a learner with slower processing time relative to the environment alteration, the agent recognizes only some time-average of the true environment, and the perceived ‘blurred’ environment makes the learner to lose some statistical efficiency or even unable to learn at all. On the other hand, a quantum learner can handle the environment change by facilitating the agent with enough time to learn before the environment changes and thus improve the statistical efficiency. See [Biamonte et al. \(2017\)](#), [Ciliberto et al. \(2018\)](#), [Dunjko and Briegel \(2018\)](#), [Wang and Song \(2020\)](#) and [Wittek \(2014\)](#) for more details.

4. Quantum Supremacy

Sections 3.3 and 3.4 indicate quantum algorithms with theoretically proven quantum speedups, such as Shor’s factoring algorithm, Grover’s search algorithm, and other machine learning algorithms. However, from the algorithmic implementation point of view, we need to build quantum computers with a huge number of qubits in order to actually run the fast quantum algorithms on these large-scale quantum computers and practically demonstrate their theoretically proven quantum advantages, which is not possible at the

present time, due to the significant difficulties to create quantum computers of large-scale with current technology. It is critical to acquire scalable architectures for constructing quantum computers with about 100 well-behaved qubits in the near-term future. We may employ such architectures to show the so-called quantum computational supremacy, where quantum supremacy refers to any practical major milestone achievement in the quest for outperforming classical computers on some tough computational tasks. Quantum computational supremacy is of great current interest in quantum computing and vigorously investigated by academic institutes, government labs, and private companies. We highlight below two landmark quantum supremacy projects (Arute et al. 2019; Zhong et al. 2020).

4.1. RANDOM QUANTUM CIRCUITS

This subsection presents the Google quantum supremacy study in Arute et al. (2019) that is based on random quantum circuits for solving a hard sampling problem. A quantum circuit is a quantum computation model where a computation is a sequence of quantum gates, which are reversible transformations on a quantum mechanical analog of a classical n -bit register. Random quantum circuits are created as a quantum computation model with statistical sampling (in fact, a mixture of size-biased samplings) as its computation task. This model is proposed for the study of quantum computational supremacy. The computational task is to generate each random quantum circuit in a specific way so that we can sample from the output distribution corresponding to the generated quantum circuit. The goal is to construct random quantum circuits with ‘enough complexity’ so that even the most powerful classical supercomputer available at that time cannot directly simulate the constructed quantum circuits in practice.

4.1.1. Output Distribution. This part illustrates the way to generate random quantum circuits and describes their output distributions. A random quantum circuit refers to a sequence of clock cycles of one-qubit and two-qubit gates with gates applied to different qubits in the same cycle. The number of clock cycles, denoted by m , is called the depth of the circuit, and the number of qubits, denoted by n , is called the width of the circuit. When the gates to be put in use are randomly selected from the set of universal quantum gates, the unitary matrix \mathbf{U} of the resulted quantum circuit is a random matrix. As the depth of the circuit goes to infinity, the distribution of the random unitary matrix \mathbf{U} converges to the Haar measure on the unitary group of degree n .

Denote by $\mathcal{X} = \{|\mathbf{x}\rangle = |x_1x_2\dots x_n\rangle : x_i \in \{0, 1\}\}$ the set of quantum computational basis states. Then \mathcal{X} consists of $d = 2^n$ states. For a quantum circuit with a unitary matrix \mathbf{U} , let $|\psi_{\mathbf{U}}\rangle = \mathbf{U}|\psi_0\rangle$ be its output state, where ψ_0 is an input state. Given a computational basis state $|\mathbf{x}\rangle$, define measurement probability $p_{\mathbf{U}}(\mathbf{x}) = |\langle \mathbf{x} | \psi_{\mathbf{U}} \rangle|^2$, namely, the probability of obtaining measurement outcome \mathbf{x} . The quantum state of the random quantum circuit can be expressed as a linear combination of the computational basis with $d = 2^n$ amplitudes. As each amplitude has real and imaginary parts, there are total $2d = 2^{n+1}$ amplitude parameters. Because of the normalization constraint, the parameters lies in the unit sphere of a $2d$ -dimensional Euclidean space. If the unitary matrix of the random quantum circuit follows the Haar distribution, the distribution of the amplitude parameters will be uniform on the unit sphere. Thus, as the depth m of the random quantum circuit goes to infinity, the measurement probability $p_{\mathbf{U}}(\mathbf{x})$ approaches the Porter-Thomas distribution (Rinott et al. 2020).

4.1.2. Challenges in Output Distribution Sampling. Quantum supremacy can be demonstrated via random quantum circuits by checking quantum computers against state-of-the-art classical computers in the task of sampling the output distributions of random quantum circuits. A major part of the quantum supremacy study is a purely statistical endeavor. It includes estimating the noise level in the quantum circuits, assessing their fidelity, and validating that the simulated bitstrings data are actually generated from the claimed target distribution.

For a quantum computer, sampling the output distribution of a random quantum circuit means performing measurements on the qubits of the quantum circuit in the computational basis to generate a set of bitstrings, such as $\{0101101, 1001010, \dots\}$. Due to the noise in the circuit, the probability distribution of the observed bitstrings is different from the ideal output distribution $p_{\mathbf{U}}(\mathbf{x})$ as described in Section 4.1.1 and Sections 4.1.3 and 4.1.4 below. On the other hand, because the complexity of a random quantum circuit grows exponentially in its size, which is defined by its width (the number of qubits) and depth (the number of cycles), classical algorithms for simulating its output distribution suffer from an exponential scaling of runtime with circuit size, and classical simulation of the output distribution is practically prohibitive. Indeed, at the time of the quantum supremacy study, classical sampling the bitstrings distribution is intractable in the quantum supremacy regime of random quantum circuits with 53 qubits and 20 cycles. Furthermore, conventional tomographic estimation described in Section 2.4 scales exponentially in the circuit size, and an exponential number of bitstrings must be generated in order to statistically recover the circuit output distribution. These challenges motivate new statistical developments for the quantum supremacy study in the subsequent sections.

4.1.3. Cross-Entropy Benchmarking. Consider a sample $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, where \mathbf{x}_j are bitstrings obtained from measurements of every qubit in the computational basis. The joint distribution of \mathcal{S} is given by $Pr_{\mathbf{U}}(\mathcal{S}) = \prod_{i=1}^N p_{\mathbf{U}}(\mathbf{x}_i)$. An application of the central limit theorem leads to

$$\frac{1}{N} \log Pr_{\mathbf{U}}(\mathcal{S}) = \frac{1}{N} \sum_{i=1}^N \log p_{\mathbf{U}}(\mathbf{x}_i) = -\mathcal{H}(p_{\mathbf{U}}) + O(N^{-1/2}),$$

where $\mathcal{H}(p_{\mathbf{U}})$ is the Shannon entropy of the output distribution $p_{\mathbf{U}}$.

For comparison, let us consider a sample $\mathcal{S}_* = \{\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_N^*\}$ as outputs from a classical or quantum operation taking a specification of some random circuit \mathbf{U}_* , where the distribution of \mathbf{x}_j^* depends on the unitary matrix \mathbf{U}_* , and \mathbf{x}_j^* are uncorrelated with the output measurements \mathbf{x}_j of the quantum circuit \mathbf{U} . For example, we may take \mathbf{U}_* as a noisy version of \mathbf{U} in quantum computation, and \mathbf{x}_j^* are measured bitstrings obtained from a quantum circuit with noise (i.e. measurement outcomes of the quantum circuit \mathbf{U}_*), while \mathbf{x}_j are ideal bitstrings obtained from a quantum circuit without noise (i.e. measurement outcomes of the quantum circuit \mathbf{U}). Again applying the central limit theorem we obtain

$$\frac{1}{N} \log Pr_{\mathbf{U}}(\mathcal{S}_*) = \frac{1}{N} \sum_{i=1}^N \log p_{\mathbf{U}}(\mathbf{x}_i^*) = -\mathcal{H}(p_*, p_{\mathbf{U}}) + O(N^{-1/2}),$$

where $p_*(\mathbf{x}) = |\langle \mathbf{x} | \mathbf{U}_* | \psi_0 \rangle|^2$ is the output distribution of $\mathbf{x}_j^* \in \mathcal{S}_*$ associated with \mathbf{U}_* , and $\mathcal{H}(p_*, p_{\mathbf{U}})$ is the cross-entropy between the two distributions. In the quantum supremacy study described below, we use the observed bitstrings \mathbf{x}_i^* to estimate the cross-entropy

benchmarking by the average of the simulated probability distribution evaluated at the observed bitstrings \mathbf{x}_i^* , where the simulated probability distribution refers to the ideal bit-string probability $p_{\mathbf{U}}(\cdot)$ for the quantum circuit \mathbf{U} that is computed by classical simulations, and the average is over the bitstrings \mathbf{x}_i^* measured for the noisy quantum circuit \mathbf{U}_* .

4.1.4. Quantum Circuit Model and Fidelity Estimation. The Google quantum AI research group constructed a quantum processor (computer) named ‘Sycamore’ of 53 programmable superconducting qubits to implement random quantum circuits in a two dimensional lattice. The implementation creates quantum states on 53 qubits, corresponding to a computational state-space of dimension 2^{53} ($\approx 10^{16}$). An approach based on the described cross-entropy benchmarking is adopted to handle noisy random quantum circuits and statistical sampling from bitstrings probability distributions. Cross-entropy benchmarking (XEB), denoted by \mathcal{F}_{XEB} , is defined as the expectation of a function of the ideal output distribution $p_{\mathbf{U}}(\cdot)$ with respect to the noisy output distribution.

To be specific, denote by \mathcal{U} a set of r random quantum circuits $\mathbf{U}_1, \dots, \mathbf{U}_r$ with n qubits and m cycles. Each circuit $\mathbf{U} \in \mathcal{U}$ is executed N times on the quantum processor, and every execution of the circuit \mathbf{U} means that a quantum operation (corresponding to \mathbf{U}_* described in Section 4.1.3) as an imperfect realization of \mathbf{U} (due to the noise) is applied to the input state $|\psi_0\rangle$ (with density matrix $|\psi_0\rangle\langle\psi_0|$). We model the imperfect realization of \mathbf{U} by a noise model with the density matrix $\rho_{\mathbf{U}}^*$ of the noisy quantum operation as follows,

$$\rho_{\mathbf{U}}^* = \Upsilon|\psi_{\mathbf{U}}\rangle\langle\psi_{\mathbf{U}}| + (1 - \Upsilon)\chi_{\mathbf{U}}, \quad 3.$$

where $|\psi_{\mathbf{U}}\rangle = \mathbf{U}|\psi_0\rangle$ is the ideal output state, $\Upsilon = \langle\psi_{\mathbf{U}}|\rho_{\mathbf{U}}^*|\psi_{\mathbf{U}}\rangle$ is the fidelity, and $\chi_{\mathbf{U}}$ represents the density matrix of the noise which along with fidelity Υ describe the effect of the errors. The output probability distribution of $\rho_{\mathbf{U}}^*$ in 3. is given by

$$p_{\mathbf{U},\Upsilon}(\mathbf{x}) = \Upsilon\langle\mathbf{x}|\psi_{\mathbf{U}}\rangle\langle\psi_{\mathbf{U}}|\mathbf{x}\rangle + (1 - \Upsilon)\langle\mathbf{x}|\chi_{\mathbf{U}}|\mathbf{x}\rangle = \Upsilon p_{\mathbf{U}}(\mathbf{x}) + (1 - \Upsilon)\langle\mathbf{x}|\chi_{\mathbf{U}}|\mathbf{x}\rangle. \quad 4.$$

Then an expression for the cross-entropy benchmarking \mathcal{F}_{XEB} is established as follows,

$$\mathcal{F}_{\text{XEB}} = \Upsilon d \sum_{\mathbf{x} \in \{0,1\}^n} [p_{\mathbf{U}}(\mathbf{x})]^2 - \Upsilon = \Upsilon [d \langle p_{\mathbf{U}}(\cdot) \rangle_{p_{\mathbf{U}}} - 1], \quad 5.$$

where $d = 2^n$, and $\langle p_{\mathbf{U}}(\cdot) \rangle_{p_{\mathbf{U}}}$ on the right-hand side denotes the expectation of the ideal output distribution $p_{\mathbf{U}}(\mathbf{x})$ with respect to itself that can be computed analytically or obtained numerically by simulations. Consider two special cases: (i) the bitstrings are sampled from the uniform distribution, and (ii) the bitstrings are sampled from the theoretical Porter-Thomas output distribution. For case (i), we have $p_{\mathbf{U},\Upsilon}(\mathbf{x}) = 1/d$, and thus $\mathcal{F}_{\text{XEB}} = \Upsilon = 0$. For case (ii), $p_{\mathbf{U},\Upsilon}(\mathbf{x}) = p_{\mathbf{U}}(\mathbf{x})$ is equal to the Porter-Thomas distribution, and hence $\mathcal{F}_{\text{XEB}} = \Upsilon = 1$. Furthermore, for random quantum circuits with enough depth, their theoretical output distribution is essentially the Porter-Thomas distribution, and therefore from 5. we conclude $\mathcal{F}_{\text{XEB}} \doteq \Upsilon$ —that is, \mathcal{F}_{XEB} is essentially the same as fidelity Υ even if bitstrings are sampled from noisy quantum circuits.

Equation 5. naturally leads to the following estimator of the cross-entropy benchmarking \mathcal{F}_{XEB} based on the observed bitstrings \mathbf{x}_{ij} from random quantum circuit $\mathbf{U}_j \in \mathcal{U}$ with N bitstring samples, $i = 1, \dots, N, j = 1, \dots, r$,

$$\hat{\mathcal{F}}_{\text{XEB}} = \frac{1}{Nr} \sum_{i=1}^N \sum_{j=1}^r [d p_{\mathbf{U}}(\mathbf{x}_{ij}) - 1], \quad 6.$$

which has an asymptotic variance $(1 + 2\Upsilon - \Upsilon^2)/(Nr)$.

Cross-entropy and fidelity measure the ‘closeness’ of two quantum states. For a quantum circuit with enough depth, $\mathcal{F}_{\text{XEB}} = 1$ when there is no noise in the quantum circuit and thus bitstrings are sampled from the ideal theoretical output distribution of the circuit; and $\mathcal{F}_{\text{XEB}} = 0$ when depolarizing errors are overwhelming in the quantum circuit and thus bitstrings are sampled from the uniform distribution. Intuitively \mathcal{F}_{XEB} calibrates how often high-probability bitstrings are sampled, and its value corresponds to the probability that bitstrings are sampled from the ideal quantum circuits (i.e. no error has occurred while running the circuit). From 5. and 6., we must obtain the ideal output probability $p_{\mathbf{U}}(\mathbf{x})$ by classically simulating the quantum circuit \mathbf{U} for evaluating \mathcal{F}_{XEB} and $\hat{\mathcal{F}}_{\text{XEB}}$, which is exponentially hard. Hence, it is intractable to compute \mathcal{F}_{XEB} and its estimates in the regime of quantum supremacy such as random quantum circuits with 53 qubits and 20 cycles. As we will present in Section 4.1.5 below, using quantum techniques to manipulate quantum circuits and statistical methodologies to model and analyze experimental data, a statistical extrapolation approach has been developed to statistically secure a high enough \mathcal{F}_{XEB} for random quantum circuits that are practically prohibitive for classical computers to simulate at the time. The approach utilizes classical numerical simulations to evaluate the likelihood of observed bitstrings, but does not require the reconstruction of the bitstring output probability distribution, which needs an exponential number of bitstrings for the increasing number of qubits.

4.1.5. Statistical Analysis for Quantum Supremacy. Classical computers were used to simulate random quantum circuits for confirming the quantum computer and calculating \mathcal{F}_{XEB} and $\hat{\mathcal{F}}_{\text{XEB}}$ as well as estimating the cost of classical sampling of bitstrings from the random quantum circuits. They were employed to verify that the quantum computer is working correctly by checking how often bitstrings are observed experimentally against their corresponding probabilities evaluated via classical simulations. The classical computers used include Google cloud cluster of 1000 machines and the Jülich supercomputer (with 100,000 cores and 250 terabytes) as well as the Summit supercomputer (the most powerful supercomputer in the world at the time).

We cannot evaluate \mathcal{F}_{XEB} and $\hat{\mathcal{F}}_{\text{XEB}}$ for random quantum circuits in the supremacy regime. However, we may modify the design of random quantum circuits to reduce their complexity so that they can be classically simulated to obtain output distributions and compute \mathcal{F}_{XEB} and/or $\hat{\mathcal{F}}_{\text{XEB}}$ for verifying physical models and validating statistical analysis. All the modified random quantum circuits closely mimics the full experiment with random quantum circuits in the supremacy regime while still remain classically simulatable, and their experimental data and associated statistical analysis provide models to track the cross-entropy benchmarking fidelity \mathcal{F}_{XEB} in the supremacy regime. Various experiments were conducted for r modified random quantum circuits with n qubits and m cycles to collect N bitstrings, where N ranges from half to five million, with r , n and m up to 10, 53 and 20, respectively. Experiments were carried out to collect $N = 3$ million bitstrings on each of $r = 10$ modified quantum circuits with $n = 53$ qubits and $m = 20$ cycles. The data were used to estimate $\mathcal{F}_{\text{XEB}} \doteq \Upsilon$ and assess the output distribution, and various statistical methods were employed to evaluate the cross-entropy benchmarking fidelity estimator $\hat{\mathcal{F}}_{\text{XEB}}$ and its asymptotic variance and validate the theoretical output distribution. Also the systematic uncertainty (Sinervo 2003) was quantified by a linear fitting to model how the fidelity varies over time, as the performance of the quantum system may fluctuate

and/or degrade with time. After the statistical checking and validation, with a total 30 million bitstring data, an estimated value of 2.24×10^{-3} was found for the mean cross-entropy benchmarking fidelity of $r = 10$ modified random quantum circuits with $n = 53$ qubits and $m = 20$ cycles, where the square root of its mean square error is estimated to be 0.21×10^{-3} . Based on these statistical results, [Arute et al. \(2019\)](#) concludes that the average fidelity for running the random quantum circuits on the Sycamore quantum computer is about 0.002.

The computing experiments show that it took 200 seconds for the Sycamore quantum computer to sample a million bitstrings from random quantum circuits with $n = 53$ qubits and $m = 20$ cycles at target fidelity $\mathcal{F}_{\text{XEB}} = 0.002$, while an extrapolation based on statistical fitting of the computing data by the mentioned classical computers indicates that an equal-fidelity classical sampling would take 10,000 years on a million cores, with further millions of years to confirm the fidelity using classical methods. Furthermore, the quantum computer consumed several orders of magnitude less energy to perform the sampling task than the Summit supercomputer. Therefore, quantum supremacy is demonstrated by the performed sampling task on the Sycamore quantum computer that is practically beyond the reach of the fastest classical supercomputers available at the time. See [Arute et al. \(2019\)](#), [Aaronson and Chen \(2016\)](#), [Boixo et al. \(2018\)](#), [Bouland et al. \(2018\)](#), [Neill et al. \(2018\)](#), and [Rinott et al. \(2020\)](#) for more details.

We like to point out again that the Google quantum supremacy study heavily relies on statistics. In spite of extensive statistical analysis conducted in [Arute et al. \(2019\)](#), there are many statistical issues that deserve further investigations. For example, the authors of this paper found that the noisy quantum circuit model does not fit to the generated bitstrings in [Arute et al. \(2019\)](#).

4.2. BOSON SAMPLING

This subsection describes the boson sampling quantum supremacy study reported by [Zhong et al. \(2020\)](#). Boson sampling is a special quantum computation model based on linear optics where the required physical devices are single-photon sources, beamsplitters, phaseshifters and photodetectors. The quantum computation model for boson sampling arranges n identical bosons to pass through a network of passive optical elements (beamsplitters and phase-shifters) and then detect the locations of the bosons, and its purpose is to sample from the output distribution for demonstrating quantum supremacy. We introduce two equivalent ways to define the boson sampling model, where one naturally leads to quantum computation, and another directly shows the difficulty in classical computation.

4.2.1. Physical Model Definition. Consider the quantum system involving n identical photons and m modes, where ‘mode’ can be loosely interpreted as the place that a photon can be in, and we are only interested in the case that m is greater than or equal to n . Note that we may write the computational basis states in the form of $|\mathbf{s}\rangle = |s_1, s_2, \dots, s_m\rangle$, where s_i is the number of photons detected in the i -th mode. Denote the set corresponding to all the computational basis states by

$$\Omega_{m,n} = \{\mathbf{s} = (s_1, s_2, \dots, s_m) : s_i \in \mathbb{N}, s_1 + s_2 + \dots + s_m = n\}.$$

The number of elements in the set $\Omega_{m,n}$ is $M = \binom{m+n-1}{n}$. Since a general state can be expressed as a linear combination of the computational basis states with complex coefficients whose squared norms sum up to 1, we may write a general computational state for the boson

computer with n photons and m modes in the following form

$$|\psi\rangle = \sum_{\mathbf{s} \in \Omega_{m,n}} \alpha_{\mathbf{s}} |\mathbf{s}\rangle, \quad \text{where} \quad \sum_{\mathbf{s} \in \Omega_{m,n}} |\alpha_{\mathbf{s}}|^2 = 1.$$

With no loss of generality we assume the initial state of a quantum computer to be $|\mathbb{1}_n\rangle \equiv |1, \dots, 1, 0, \dots, 0\rangle$, which means that each of the first n modes contains one photon and the remaining modes do not contain any photon. Consider how the linear optical elements work in a special case when there is only one photon in the quantum system. Each of phaseshifters and beamsplitters takes action on at most two modes, without any action on the other $m - 2$ modes. Assume that a phaseshifter takes action on the i -th mode, and $\mathbf{s} \in \Omega_{m,n}$ indicates the photon in the i -th mode. Then the phaseshifter changes only one amplitude $\alpha_{\mathbf{s}}$ by multiplying it with $e^{i\theta}$ for some specified θ , but does not alter any other amplitudes. Suppose that a beamsplitter takes action on the i -th and j -th modes, and denote the corresponding quantum states by $|\mathbf{s}\rangle$ and $|\mathbf{t}\rangle$ respectively. Then the action maps the two amplitudes $\alpha_{\mathbf{s}}$ and $\alpha_{\mathbf{t}}$ into $\check{\alpha}_{\mathbf{s}}$ and $\check{\alpha}_{\mathbf{t}}$, where their relationship can be specified by the following transformation,

$$\begin{bmatrix} \check{\alpha}_{\mathbf{s}} \\ \check{\alpha}_{\mathbf{t}} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \alpha_{\mathbf{s}} \\ \alpha_{\mathbf{t}} \end{bmatrix}.$$

As the quantum system in this case has only one photon, there are total m computational basis states. Consequently the unitary matrices representing the transformations of a beamsplitter and a phaseshifter are equal to the m dimensional identity matrices except for a 2×2 submatrix and a diagonal entry corresponding to the amplitude change, respectively. Then the product of such matrices corresponding to the optical elements in a linear optical network yields a $m \times m$ unitary matrix \mathbf{U} to represent the unitary transformation of the linear optical network. Conversely, any $m \times m$ unitary matrix \mathbf{U} can be mathematically decomposed as a product $\mathbf{U} = \mathbf{U}_T \dots \mathbf{U}_1$, where each \mathbf{U}_i corresponds to a unitary matrix of a beamsplitter or a phaseshifter, and $T = O(m^2)$. That is, we can use only linear optical elements to implement any $m \times m$ unitary transformation.

Given a quantum system with n photons, it can be shown that the unitary transformation corresponding to the optical network has a unitary matrix representation $\phi(\mathbf{U}) = \phi(\mathbf{U}_T) \dots \phi(\mathbf{U}_1)$, where ϕ is a homomorphism map. We just need to identify the specification of each $\phi(\mathbf{U}_i)$ to obtain an explicit expression for $\phi(\mathbf{U})$.

Assume that the i -th optical element is a phaseshifter. We can mathematically express its action as the following diagonal unitary transformation,

$$|s_1, s_2, \dots, s_m\rangle \rightarrow e^{i\theta s_i} |s_1, s_2, \dots, s_m\rangle,$$

and the corresponding unitary matrix representation yields an expression for $\phi(\mathbf{U}_i)$.

Assume that the j -th optical element is a beamsplitter. The action of the beamsplitter is hard to describe. It takes action on two modes only, without making any change for the other $m - 2$ modes. Suppose that the two modes the beamsplitter has acted on are the i -th and j -th modes. Then the corresponding unitary transformation can be expressed as follows,

$$\begin{aligned} & |s_1, s_2, \dots, s_{i-1}, u, s_{i+1}, \dots, s_{j-1}, v, s_{j+1}, \dots, s_m\rangle \\ & \rightarrow \sum_{s+t=u+v} \beta_{u,v,s,t} |s_1, s_2, \dots, s_{i-1}, s, s_{i+1}, \dots, s_{j-1}, t, s_{j+1}, \dots, s_m\rangle, \end{aligned}$$

where

$$\beta_{u,v,s,t} = \sqrt{\frac{u!v!}{s!t!}} \sum_{k+l=u, k \leq s, l \leq t} \binom{s}{k} \binom{t}{l} (-1)^{s-k} (\sin \theta)^{s+l-k} (\cos \theta)^{k+t-l}$$

for some specified angle θ . The unitary matrix corresponding to this transformation renders an expression for $\phi(\mathbf{U}_j)$.

Using the derived expression for each $\phi(\mathbf{U}_i)$ along with $\phi(\mathbf{U}) = \phi(\mathbf{U}_T) \dots \phi(\mathbf{U}_1)$, we can obtain the unitary matrix for representing the action of the whole optical network. Initiating in the state $|\mathbf{1}_n\rangle$, and then passing through the optical network, the photons will be in the quantum state $\phi(\mathbf{U})|\mathbf{1}_n\rangle$. We measure the state to obtain a measurement outcome corresponding to a computational basis state. The state we obtain is random, and the probability of obtaining a given computational basis state $|\mathbf{s}\rangle$ is equal to $|\langle \mathbf{1}_n | \phi(\mathbf{U}) | \mathbf{s} \rangle|^2$. Treating $|\langle \mathbf{1}_n | \phi(\mathbf{U}) | \mathbf{s} \rangle|^2$ as a mapping from $\mathbf{s} \in \Omega_{m,n}$ to $[0, 1]$, we obtain a probability distribution on $\Omega_{m,n}$ that assigns probability $Pr(\mathbf{s})$ to state $|\mathbf{s}\rangle$, where

$$Pr(\mathbf{s}) = |\langle \mathbf{1}_n | \phi(\mathbf{U}) | \mathbf{s} \rangle|^2, \quad \mathbf{s} \in \Omega_{m,n}. \quad 7.$$

4.2.2. Permanent Based Model Definition. Alternatively, we may define boson sampling by the permanents of the submatrices of the unitary matrix representing the optical network. Let $\mathbf{A} = (a_{ij})$ be a $n \times n$ matrix, and define its permanent as follows,

$$Per(\mathbf{A}) = \sum_{\pi \in \mathcal{S}_n} \prod_{i=1}^n a_{i\pi(i)},$$

where \mathcal{S}_n is the set of all permutations of $1, \dots, n$.

Given a $m \times m$ unitary matrix \mathbf{U} and $\mathbf{s} = (s_1, \dots, s_m) \in \Omega_{m,n}$, we construct a $n \times n$ matrix $\mathbf{U}_\mathbf{s}$ from \mathbf{U} by retaining its first n columns and then copying s_i times its i -th row. Define a discrete probability distribution on $\Omega_{m,n}$ as follows,

$$Pr(\mathbf{s}) = \frac{|Per(\mathbf{U}_\mathbf{s})|^2}{s_1! \dots s_m!}. \quad 8.$$

It can be proved that the probability distribution 8. is equal to the probability distribution 7. in the way described in Section 4.2.1 through the system with n photons, m modes and an optical network whose action is represented by the unitary matrix \mathbf{U} .

4.2.3. Quantum Supremacy with Boson Sampling. The Boson sampling problem is defined as sampling from the distribution $Pr(\mathbf{s})$ defined in 7. or 8.. Gaussian Boson sampling makes use of Gaussian states as probability sources of photons, and the resulted output distribution can be further expressed as matrix functions called Hafnian and Torontonian. Since Permanent, Hafnian and Torontonian are in the #P-complete complexity class, it is intractable for classical computers to evaluate the matrix functions and thus handle Boson sampling. On the other hand, the physical definition inherently shows that it is possible to carry out a successful quantum computing experiment on an optical network with appropriate size and thus render quantum supremacy.

The quantum supremacy study reported in Zhong et al. (2020) built a photonic quantum computer (processor) called Jiuzhang to perform Gaussian Boson sampling. The photonic quantum computer Jiuzhang can enable up to 76 qubits to successfully accomplish Gaussian

Boson sampling tasks that are beyond the capacity of the fastest classical supercomputers at the time. [Zhong et al. \(2020\)](#) documented physical experiments performed and statistical analysis undertaken to verify quantum states and validate output distributions based on the generated samples in the easy regime where the full output distributions can be obtained. It provided circumstantial evidence to support the results in the quantum supremacy regime where a full verification is not possible due to the intractable nature of Boson sampling. Also time costs to run Gaussian Boson sampling on supercomputers in the prohibitive regime were estimated based on statistical fitting of classical computational cost data. In a nutshell, [Zhong et al. \(2020\)](#) announced that a 200-second job of Gaussian Boson sampling on Jiuzhang would require 0.6 billion years for the fastest supercomputer available at the time to finish. Hence, quantum supremacy is demonstrated by the performed Gaussian Boson sampling on the photonic quantum computer that is practically beyond the reach of the fastest classical supercomputers available at the time. More details can be found in [Aaronson and Arkhipov \(2011\)](#), [Hamilton et al. \(2017\)](#), [Harrow and Montanaro \(2017\)](#), [Lund et al. \(2017\)](#), [Markov et al. \(2018\)](#), [Quesada et al. \(2018\)](#) and [Zhong et al. \(2020\)](#).

5. QUANTUM ANNEALING

Quantum annealing is the quantum analog of classical annealing, with thermodynamics replaced by quantum dynamics. Both classical annealing and quantum annealing are employed to solve optimization problems whose objective function can be represented by the energies of physical systems. Quantum annealing may be considered as a special-purpose quantum computer that is designed to effectively solve specific optimization problems, and quantum annealers are physical hardware devices to implement quantum annealing. See [McGeoch \(2014\)](#) and [Wang et al. \(2016\)](#).

5.1. Classical Ising Model

Consider a classical Ising model described by a graph $\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$, where $\mathcal{V}(\mathcal{G})$ and $\mathcal{E}(\mathcal{G})$ are the vertex and edge sets of \mathcal{G} , respectively. We assign each vertex a random variable with value in $\{+1, -1\}$, and let each edge represent the interaction between two vertex variables connected by the edge. A configuration $\mathbf{s} = \{s_j, j \in \mathcal{V}(\mathcal{G})\}$ is a set of values assigned to all vertex variables $s_j, j \in \mathcal{V}(\mathcal{G})$. In physics vertices and vertex variables are called sites and spins, respectively, and we refer to the spin values $+1$ and -1 as spin up and spin down, respectively. Define the following Hamiltonian for the classical Ising model,

$$\mathbf{H}_I^c \equiv \mathbf{H}_I^c(\mathbf{s}) = - \sum_{(i,j) \in \mathcal{E}(\mathcal{G})} \delta_{ij} s_i s_j - \sum_{j \in \mathcal{V}(\mathcal{G})} \gamma_j s_j, \quad 9.$$

where (i, j) represents the edge between the sites i and j , the first summation is over all $(i, j) \in \mathcal{E}(\mathcal{G})$, δ_{ij} denotes the interaction between sites i and j associated with edge $(i, j) \in \mathcal{E}(\mathcal{G})$, and γ_j stands for an external magnetic field on vertex $j \in \mathcal{V}(\mathcal{G})$. We refer to a set of fixed values $\{\delta_{ij}, \gamma_j\}$ as an instance of the Ising model, and $\mathbf{H}_I^c(\mathbf{s})$ as the energy of the Ising model at configuration \mathbf{s} . Then the probability of a specific configuration \mathbf{s} is described by the Boltzmann distribution of the Ising model as follows,

$$P_T(\mathbf{s}) = \frac{e^{-\mathbf{H}_I^c(\mathbf{s})/T}}{Z_T}, \quad Z_T = \sum_{\mathbf{s}} e^{-\mathbf{H}_I^c(\mathbf{s})/T}, \quad 10.$$

where T is the fundamental temperature of the system in units of energy.

We illustrate the classical annealing by the Ising model as follows. The annealing is used to solve a combinatorial minimization problem whose objective function is represented by the Ising energy function $\mathbf{H}_I^c(\mathbf{s})$. Denote by b the total number of sites in the Ising model. As $\mathbf{H}_I^c(\mathbf{s})$ is defined over $\mathbf{s} \in \{-1, +1\}^b$, it is prohibitive to search over the exponential large configuration space for a minimizer of $\mathbf{H}_I^c(\mathbf{s})$ by deterministic exhaustive search algorithms. We resort annealing methods to explore the huge search space probabilistically and search for a configuration with the minimal energy. For example, we implement simulated annealing by using Markov chain Monte Carlo (MCMC) to generate configurations from the Boltzmann distribution $P_T(\mathbf{s})$ with slowly decreasing temperature T . The lowest energy state is often called a ground state in physics. See [Bertsimas and Tsitsiklis \(1993\)](#), [Kirkpatrick et al. \(1983\)](#), and [Wang et al. \(2016\)](#).

5.2. Quantum Ising Model

The same graph \mathcal{G} is used to describe the quantum Ising model, where the vertex set $\mathcal{V}(\mathcal{G})$ stands for the quantum spins, with the edge set $\mathcal{E}(\mathcal{G})$ for the interactions between two quantum spins. Each vertex has a qubit that is realized by its quantum spin. As \mathcal{G} has b vertices, the vector space for the described quantum Ising system is \mathbb{C}^d ($d = 2^b$). We characterize its quantum state by a unit vector in \mathbb{C}^d , and its dynamic evolution by a Hermitian matrix of size d , which is called a quantum Hamiltonian for the quantum system. The energies of the quantum system are defined to be the eigenvalues of the quantum Hamiltonian, and ground states refer to the eigenvectors corresponding to the smallest eigenvalue. Set

$$\mathbf{I}_j = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_j^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_j^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad j = 1, \dots, b,$$

where σ_j^x and σ_j^z are called Pauli matrices in x and z axes, respectively. We replace s_j in the classical Ising Hamiltonian $\mathbf{H}_I^c(\mathbf{s})$ by σ_j^z to define the quantum Hamiltonian as follows,

$$\mathbf{H}_I^q = - \sum_{(i,j) \in \mathcal{E}(\mathcal{G})} \delta_{ij} \sigma_i^z \sigma_j^z - \sum_{j \in \mathcal{V}(\mathcal{G})} \gamma_j \sigma_j^z, \quad 11.$$

where γ_j and δ_{ij} stand for the local field on the vertex $j \in \mathcal{V}(\mathcal{G})$ and the Ising interaction along the edge $(i, j) \in \mathcal{E}(\mathcal{G})$, respectively, and we use the quantum convention that σ_j^z and $\sigma_i^z \sigma_j^z$ in 11. denote their tensor products with identity matrices as follows,

$$\sigma_j^z \equiv \mathbf{I}_1 \otimes \cdots \otimes \mathbf{I}_{j-1} \underbrace{\otimes \sigma_j^z}_{\text{vertex } j} \otimes \mathbf{I}_{j+1} \otimes \cdots \otimes \mathbf{I}_b, \quad 12.$$

$$\sigma_i^z \sigma_j^z \equiv \mathbf{I}_1 \otimes \cdots \otimes \mathbf{I}_{i-1} \underbrace{\otimes \sigma_i^z \otimes \mathbf{I}_{i+1} \otimes \cdots \otimes \mathbf{I}_{j-1} \otimes \sigma_j^z}_{\text{vertices } i \text{ and } j} \otimes \mathbf{I}_{j+1} \otimes \cdots \otimes \mathbf{I}_b. \quad 13.$$

Observe that the quantum Hamiltonian \mathbf{H}_I^q in 11. is a diagonal matrix of size 2^b whose diagonal elements (eigenvalues) are the same as the classical Hamiltonian $\mathbf{H}_I^c(\mathbf{s})$ in 9. corresponding to all 2^b binary states \mathbf{s} ordered lexicographically. Therefore finding the minimal energy of the classical Ising Hamiltonian \mathbf{H}_I^c is equivalent to finding the minimal energy of the quantum Ising Hamiltonian \mathbf{H}_I^q . The quantum formulation of the original optimization

problem is to facilitate the design of quantum annealing in the next subsection, but so far the computational task for solving the optimization problem is still the same as in the classical case.

5.3. Quantum Ising Model in the Transverse Field

Quantum annealing requires an introduction of a transverse magnetic field to yield a quantum Hamiltonian in the transverse field. Define a quantum Hamiltonian to govern the transverse magnetic field as follows,

$$\mathbf{H}_X = - \sum_{j \in \mathcal{V}(\mathcal{G})} \sigma_j^x, \quad 14.$$

where again we adopt the quantum convention to denote by σ_j^x the tensor products of b matrices of size 2,

$$\sigma_j^x \equiv \mathbf{I}_1 \otimes \cdots \otimes \mathbf{I}_{j-1} \underbrace{\otimes \sigma_j^x \otimes}_{\text{vertex } j} \mathbf{I}_{j+1} \otimes \cdots \otimes \mathbf{I}_b. \quad 15.$$

Observe that the $2^b \times 2^b$ non-diagonal matrix \mathbf{H}_X in 14. does not commute with diagonal matrix \mathbf{H}_I^q in 11. and thus introducing \mathbf{H}_X changes the system behavior from classical to quantum. Due to the simple symmetric structure of \mathbf{H}_X , we can derive explicit expressions for its eigenvalues and eigenvectors. In particular $(1, \dots, 1)'$ is its ground state, the eigenvector corresponding to the smallest eigenvalue of \mathbf{H}_X .

Quantum annealing proceeds as follows. The quantum annealing system is initially driven by the transverse magnetic field \mathbf{H}_X prepared in its ground state $(1, \dots, 1)'$, and then we slowly drive the system from the initial Hamiltonian \mathbf{H}_X to its final target Hamiltonian \mathbf{H}_I^q . In specific, we engineer the quantum annealing process through the instantaneous Hamiltonian for the Ising model in the transverse field as follows,

$$\mathbf{H}_D(t) = A(t)\mathbf{H}_X + B(t)\mathbf{H}_I^q, \quad t \in [0, t_f], \quad 16.$$

where t_f denotes the total annealing time, and time varying smooth functions $A(t)$ and $B(t)$ are called the annealing schedules which satisfy $A(t_f) = B(0) = 0$, $A(t)$ is decreasing, and $B(t)$ is increasing. It is evident that at $t = 0$, $\mathbf{H}_D(0) = A(0)\mathbf{H}_X$, and at $t = t_f$, $\mathbf{H}_D(t_f) = B(t_f)\mathbf{H}_I^q$. As $A(0)$ and $B(t_f)$ are known constants, $\mathbf{H}_D(t)$ has identical eigenvectors as \mathbf{H}_X at the initial time $t = 0$ and as \mathbf{H}_I^q at the final time $t = t_f$, with the corresponding eigenvalues differing by factors $A(0)$ and $B(t_f)$, respectively. Consequently through $\mathbf{H}_D(t)$ we can engineer the system to move from \mathbf{H}_X initialized in its ground state to the final target \mathbf{H}_I^q .

The adiabatic quantum theorem shows that during the evolution of quantum annealing, the system tends to stay in the ground states of the instantaneous Hamiltonian via quantum tunneling (Farhi et al. 2000, 2001, 2002; McGeoch 2014). At the end of the annealing procedure, we measure the quantum system. With some probability the quantum system stays in a ground state of the final Hamiltonian \mathbf{H}_I^q , and thus the measurement outcome renders a solution to the optimization problem. That is, we can utilize the quantum annealing procedure driven by 16. to find the global minimum of $\mathbf{H}_I^q(\mathbf{s})$ and solve the original minimization problem. See Brooke et al. (1999), Hu and Wang (2021), Isakov et al. (2016), Jörg et al. (2010), Wang et al. (2016), and Wang and Song (2020) for details.

5.4. Quantum Annealer

Quantum annealers, quantum devices to implement quantum annealing, are currently investigated by a number of academic labs and companies, with uncertain quantum speedup. In particular, the D-Wave machine is a commercially available hardware device that is designed and built to physically implement quantum annealing. It is an analog computing device based on superconducting qubits to process quantum annealing and solve certain combinatorial optimization problems. Many experiments have been conducted to test the D-Wave machines, and computational studies, such as Monte Carlo simulations, have been carried out to assess the performance of the D-Wave machines and compare with classical and quantum models through sophisticated statistical analysis. It has been demonstrated that the D-Wave machines are useful in designing quantum algorithms and solving application problems and can be faster than classical algorithms like classical annealing, yet there is no quantum supremacy or quantum speedup over classical computation found in the D-Wave machines. See [Albash et al. \(2015\)](#), [Boixo et al. \(2014, 2016, 2018\)](#), [Brady and van Dam \(2016\)](#), [Rønnow et al. \(2014\)](#), and [Wang et al. \(2016\)](#) for more discussions.

6. QUANTUM LEARNING WITH BOLTZMANN MACHINES

Boltzmann machines (BMs) have been introduced as probabilistic generative models that contain bidirectionally connected networks of stochastic binary units, and can be interpreted as neural network models. They can be regarded as particular graphical models, more precisely undirected graphical models known as Markov random fields. BMs provide a model for deep learning architectures such as deep belief networks. They have the potential of learning internal representations for complex unsupervised learning tasks such as object and speech recognition problems. A BM model is mathematically equivalent to the Ising model in physics. This renders a new way to sample from a Boltzmann machine: (i) map the BM to the corresponding Ising model; (ii) engineer a physical system to realize the target problem to be solved; (iii) run the physics until the Ising system establishes some equilibrium with a state corresponding to a possible solution; (iv) measure the physical system to obtain a realization of states of the Ising model; and (v) map the Ising measurement outputs into these corresponding to the BM to render a possible solution to the original problem. The procedure requires relevant physical implementations or computer simulations, which have been employed in both classical and quantum domains with examples including classical and quantum annealing. We will review classical BMs and describe their training by quantum means as well as introduce quantum BMs for deep learning, where quantum deep learning here refers to both quantum BMs and the training of classical BMs by quantum resources. The scenarios may bear some resemblance to quantum reinforcement learning considered in Section 3.4 where mixtures of classical and quantum approaches are used in the quantum reinforcement learning framework.

6.1. Boltzmann Machines

A BM is a network of symmetrically coupled stochastic binary units, which consists of a set of visible units $\mathbf{v} \in \{0, 1\}^n$ associated with observations, and a set of hidden units $\mathbf{h} \in \{0, 1\}^m$ used to capture dependencies between observed variables. Every two nodes are connected so the model creates a fully connected undirected graph as illustrated in Figure 1. The model has a joint distribution

$$p(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z(\theta)} e^{-E(\mathbf{v}, \mathbf{h}; \theta)}, \quad Z(\theta) = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \theta)),$$

where $Z(\theta)$ is called the partition function, the energy function $E(\mathbf{v}, \mathbf{h}; \theta)$ is defined as

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\frac{1}{2} \mathbf{v}^T \mathbf{L} \mathbf{v} - \frac{1}{2} \mathbf{h}^T \mathbf{J} \mathbf{h} - \mathbf{v}^T \mathbf{W} \mathbf{h},$$

$\theta = \{\mathbf{W}, \mathbf{L}, \mathbf{J}\}$ is the model parameter, and matrices $\mathbf{W} = (W_{ij})$, $\mathbf{L} = (L_{ij})$, and $\mathbf{J} = (J_{ij})$ represent visible-to-hidden, visible-to-visible, and hidden-to-hidden symmetric interaction terms. The diagonal elements of \mathbf{L} and \mathbf{J} are set to 0. The conditional distributions of hidden and visible units are given by

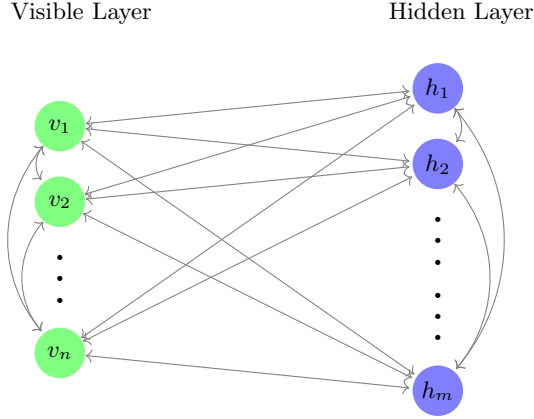


Figure 1. General Boltzmann Machine.

$$p(h_j = 1 | \mathbf{v}, \mathbf{h}_{-j}) = \sigma \left(\sum_i W_{ij} v_i + \sum_{l \neq j} J_{jl} h_l \right), \quad p(v_i = 1 | \mathbf{h}, \mathbf{v}_{-i}) = \sigma \left(\sum_j W_{ij} h_j + \sum_{k \neq i} L_{ik} v_k \right),$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function. The marginal distribution of the visible units is

$$p(\mathbf{v}; \theta) = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \theta)).$$

Given the observed data, the training of a BM corresponds to finding the parameter θ that maximizes the log-likelihood function $\log p(\mathbf{v}; \theta)$. Because for a general BM it is not possible to analytically find the maximizer of the likelihood, the usual approach is to apply gradient ascent, iteratively updating $\theta^{(t)}$ to $\theta^{(t+1)}$ by the gradient of the log-likelihood

$$\theta^{(t+1)} = \theta^{(t)} + \eta \frac{\partial}{\partial \theta^{(t)}} \left(\log p(\mathbf{v}; \theta^{(t)}) \right),$$

where η is the learning rate. It can be shown that the parameter increment during the iteration has the following expressions,

$$\begin{aligned} \Delta \mathbf{W} &= \eta (E_{P_{data}}[\mathbf{v} \mathbf{h}'] - E_{P_{model}}[\mathbf{v} \mathbf{h}']), & \Delta \mathbf{L} &= \eta (E_{P_{data}}[\mathbf{v} \mathbf{v}'] - E_{P_{model}}[\mathbf{v} \mathbf{v}']), \\ \Delta \mathbf{J} &= \eta (E_{P_{data}}[\mathbf{h} \mathbf{h}'] - E_{P_{model}}[\mathbf{h} \mathbf{h}']), \end{aligned}$$

where $E_{P_{model}}$ and $E_{P_{data}}$ denote, respectively, the expectations with respect to the model distribution $p(\mathbf{v}, \mathbf{h}; \theta)$ and the data distribution $P_{data}(\mathbf{h}, \mathbf{v}; \theta) = p(\mathbf{h} | \mathbf{v}; \theta) P_{data}(\mathbf{v})$, and

$P_{data}(\mathbf{v})$ is the empirical distribution of the observed data. We call $E_{P_{data}}$ the data-dependent expectation, and since $E_{P_{model}}$ represents an expectation taken with respect to the joint distribution $p(\mathbf{v}, \mathbf{h}; \theta)$ of the visible and hidden variables, we call it the data-independent expectation or model expectation. For general BMs, it is usually very difficult to directly compute the model expectation, due to its exponential growth in the units.

We employ MCMC approaches to estimate the data-dependent expectation and the data-independent expectation. Especially, for the computation of model expectation, the conditional distribution of every node given the other nodes is known, so standard MCMC simulation methods are often employed to compute the model expectation, although MCMC can be very costly or even impossible for large BMs. See [Salakhutdinov and Hinton \(2009, 2012\)](#), [Hinton and Salakhutdinov \(2012\)](#), and [Salakhutdinov \(2015\)](#) for more details.

6.2. Restricted Boltzmann Machines (RBMs)

Because the learning process for general BMs is time-consuming, we may impose some restrictions on the network topology to simplify the learning problem. A RBM model is a special variant of BM where every visible node is connected to every hidden node, but there is no connection between two variables of the same layer as shown in Figure 2. One major advantage of the RBM is that its model expectations are easy to calculate. Again it contains n visible units $\mathbf{v} = (v_1, \dots, v_n)$ associated with observations and m hidden units $\mathbf{h} = (h_1, \dots, h_m)$ to capture dependencies between observed variables. The joint distribution of the RBM model is given by $p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v}, \mathbf{h})}$ with the energy function

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{i=1}^m \sum_{j=1}^n w_{ij} h_i v_j - \sum_{j=1}^n b_j v_j - \sum_{i=1}^m c_i h_i.$$

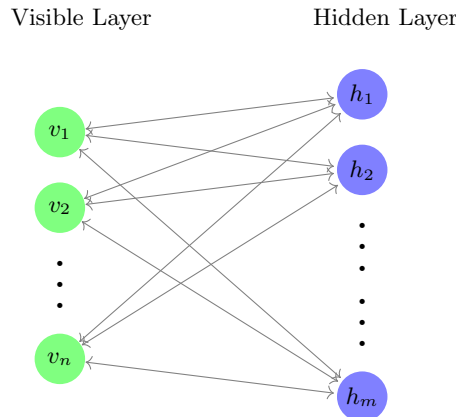


Figure 2. Restricted Boltzmann Machine.

Thus, we have

$$p(\mathbf{h}|\mathbf{v}) = \prod_{i=1}^m p(h_i|\mathbf{v}) \text{ and } p(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^n p(v_i|\mathbf{h}).$$

Since the connections between hidden variables are absent, the marginal distribution of the visible variables has a simplified expression,

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})} = \frac{1}{Z} \prod_{j=1}^n \left(e^{b_j v_j} \right) \prod_{i=1}^m \left(1 + e^{c_i + \sum_{j=1}^n w_{ij} v_j} \right).$$

The conditional distribution of h_i given \mathbf{v} and the conditional distribution of v_j given \mathbf{h} are, respectively, given by,

$$p(h_i = 1 | \mathbf{v}) = \sigma \left(\sum_{j=1}^n w_{ij} v_j + c_i \right), \quad p(v_j = 1 | \mathbf{h}) = \sigma \left(\sum_{i=1}^m w_{ij} h_i + b_j \right),$$

where $\sigma(x)$ is the sigmoid function. For the RBM model with $\theta = (w_{ij}, b_j, c_i)$, we can obtain an explicit form for the gradient of the log-likelihood function, giving

$$\theta^{(t+1)} = \theta^{(t)} + \eta \frac{\partial}{\partial \theta^{(t)}} \left[\sum_{i=1}^n \ln L(\theta^{(t)} | v_i) \right] - \lambda \theta^{(t)} + \epsilon \Delta \theta^{(t-1)},$$

where η , λ , and ϵ are positive constants, representing the learning rate, moment weight, and momentum coefficient, respectively, and $L(\theta | \mathbf{v}) = \prod_{i=1}^n L(\theta | v_i)$ is the likelihood function. Given one single training example $\bar{\mathbf{v}}$, the log-likelihood of the RBM model with the parameter θ is given by

$$\ln L(\theta | \bar{\mathbf{v}}) = \ln p(\bar{\mathbf{v}} | \theta) = \ln \left[\frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\bar{\mathbf{v}}, \mathbf{h})} \right] = \ln \sum_{\mathbf{h}} e^{-E(\bar{\mathbf{v}}, \mathbf{h})} - \ln \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})},$$

and gradient

$$\frac{\partial \ln L(\theta | \bar{\mathbf{v}})}{\partial \theta} = - \sum_{\mathbf{h}} p(\mathbf{h} | \bar{\mathbf{v}}) \frac{\partial E(\bar{\mathbf{v}}, \mathbf{h})}{\partial \theta} + \sum_{\mathbf{v}, \mathbf{h}} p(\mathbf{v}, \mathbf{h}) \frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \theta}.$$

That is,
$$\frac{\partial \ln L(\theta | \bar{\mathbf{v}})}{\partial w_{ij}} = p(h_i = 1 | \bar{\mathbf{v}}) \bar{v}_j - \sum_{\mathbf{v}} p(\mathbf{v}) p(h_i = 1 | \mathbf{v}) v_j,$$

$$\frac{\partial \ln L(\theta | \bar{\mathbf{v}})}{\partial b_j} = \bar{v}_j - \sum_{\mathbf{v}} p(\mathbf{v}) v_j, \quad \frac{\partial \ln L(\theta | \bar{\mathbf{v}})}{\partial c_i} = p(h_i = 1 | \bar{\mathbf{v}}) - \sum_{\mathbf{v}} p(\mathbf{v}) p(h_i = 1 | \mathbf{v}).$$

In above gradient expressions, the second terms correspond to the expectations of the gradient of the energy function under the model distribution, and their exact computation calls for a summation over all the visible variables, which is computationally intractable. To get around this difficulty, one way is to estimate the second term in the log-likelihood gradient by MCMC simulations, though the MCMC approach can be computationally very expensive. See [Fischer and Igel \(2012, 2014\)](#) for more details.

We may utilize RBMs to build deep Boltzmann Machines (DBMs), where a DBM model consists of multi-hidden layers, and only between-layer connections exist. For example, a DBM with two hidden layers can be built by a stack of two RBMs with tied weights. The learning procedure for BMs and RBMs described in this and previous sections can be applied to DBMs. See [Hinton and Salakhutdinov \(2012\)](#), [Salakhutdinov \(2015\)](#) and [Salakhutdinov and Hinton \(2009, 2012\)](#) for more details.

6.3. Quantum Training of Boltzmann Machines

As the typical training of BMs relies on MCMC and thus is computationally very expensive or even impossible, we may find that training with quantum resources can be very helpful in reducing the training cost. In fact the quantum approach in learning with BMs can be more feasible or preferable than the classical approach. There are quantum training techniques developed to train the classical Boltzmann machines. Examples include special-purpose quantum computers such as quantum annealing and programmable photonic circuits. In particular quantum annealing is very suitable for training BMs, and the D-Wave machine, a quantum annealer with thousands of qubits, has been explored for training BMs with deep quantum learning protocols. See [Adachi and Henderson \(2015\)](#), [Benedetti et al. \(2016\)](#) and [Wiebe et al. \(2014\)](#) for details.

6.4. Quantum Boltzmann Machines(QBMs)

As in the classical BM case, we adopt a probabilistic graphical model that consists of a set of visible units $\mathbf{v} \in \{-1, 1\}^n$ associated with observations, and a set of hidden units $\mathbf{h} \in \{-1, 1\}^m$ used to capture dependencies between observed variables. We use the notation $\mathbf{z} = (\mathbf{v}, \mathbf{h})$ to denote the combined units. A quantum Ising model is defined through its quantum Hamiltonian given by

$$\mathbf{H}_q = \sum_i b_i \sigma_i^z + \sum_{i,j} w_{ij} \sigma_i^z \sigma_j^z,$$

where as in Section 5 we write

$$\sigma_i^z \equiv \underbrace{\mathbf{I} \otimes \dots \otimes \mathbf{I}}_{i-1} \otimes \sigma_z \otimes \underbrace{\mathbf{I} \otimes \dots \otimes \mathbf{I}}_{b-i}, \quad \mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad b = m + n.$$

Define the density matrix as $\rho = Z^{-1} e^{-\mathbf{H}_q}$, where $Z = \text{tr}[e^{-\mathbf{H}_q}]$ is the partition function. The i -th diagonal element of the density matrix gives the probability associated with the i -th state of the QBM. We can derive the marginal distribution of the visible units from the density matrix. For a given visible vector $\mathbf{v} = (v_1, \dots, v_n) \in \{1, -1\}^n$, define a matrix $\Lambda_{\mathbf{v}} = |\mathbf{v}\rangle\langle\mathbf{v}| \otimes \mathbf{I}_{\mathbf{h}}$, where $\mathbf{I}_{\mathbf{h}}$ is the identical matrix of dimension equal to the number of hidden units. Then the marginal distribution corresponding to \mathbf{v} of the QBM model is given by $P_{\mathbf{v}} = \text{tr}[\Lambda_{\mathbf{v}} \rho]$.

Now we consider adding a transverse field to the Ising Hamiltonian to introduce a transverse field quantum Ising model. Define non-diagonal matrices

$$\sigma_i^x \equiv \underbrace{\mathbf{I} \otimes \dots \otimes \mathbf{I}}_{i-1} \otimes \sigma_x \otimes \underbrace{\mathbf{I} \otimes \dots \otimes \mathbf{I}}_{b-i}, \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The transverse Ising Hamiltonian is then given by,

$$\mathbf{H} = \sum_i \Gamma_i \sigma_i^x + \sum_i b_i \sigma_i^z + \sum_{i,j} w_{ij} \sigma_i^z \sigma_j^z,$$

where Γ_i , b_i , and w_{ij} are model parameters. Again we define the partition function $Z = \text{tr}[e^{-\mathbf{H}}]$, the density matrix $\rho = Z^{-1} e^{-\mathbf{H}}$, and the marginal distribution of the visible units

$P_{\mathbf{v}} = \text{tr}[\Lambda_{\mathbf{v}}\rho]$. To train the QBM, we employ gradient descent to update the parameters $\theta = (\Gamma, \mathbf{b}, \mathbf{w})$ and minimize the negative log-likelihood function

$$\mathcal{L} = - \sum_{\mathbf{v}} P_{\mathbf{v}}^{data} \log(P_{\mathbf{v}}),$$

where $P_{\mathbf{v}}^{data}$ denotes the empirical distribution of the training set. The gradient of \mathcal{L} is given by

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{\mathbf{v}} P_{\mathbf{v}}^{data} \left(\frac{\text{tr}[\Lambda_{\mathbf{v}} \frac{\partial e^{-\mathbf{H}}}{\partial \theta}]}{\text{tr}[\Lambda_{\mathbf{v}} e^{-\mathbf{H}}]} - \frac{\text{tr}[\frac{\partial e^{-\mathbf{H}}}{\partial \theta}]}{\text{tr}[e^{-\mathbf{H}}]} \right).$$

As $\text{tr}[\frac{\partial e^{-\mathbf{H}}}{\partial \theta}] = -\text{tr}[\frac{\partial \mathbf{H}}{\partial \theta} e^{-\mathbf{H}}]$, we have

$$\frac{\text{tr}[\frac{\partial e^{-\mathbf{H}}}{\partial \theta}]}{\text{tr}[e^{-\mathbf{H}}]} = - \left\langle \frac{\partial \mathbf{H}}{\partial \theta} \right\rangle,$$

where $\langle \mathbf{A} \rangle \equiv \text{tr}[\rho \mathbf{A}]$ denotes the Boltzmann average of a given matrix \mathbf{A} . We may estimate $\langle \frac{\partial \mathbf{H}}{\partial \theta} \rangle$ by sampling from the model. However, the term $\frac{\text{tr}[\Lambda_{\mathbf{v}} \frac{\partial e^{-\mathbf{H}}}{\partial \theta}]}{\text{tr}[\Lambda_{\mathbf{v}} e^{-\mathbf{H}}]}$ cannot be estimated using sampling. Thus, it is almost impossible to find the model parameters by minimizing the negative log-likelihood. We use the variational approach to solve this problem by minimizing an upper bound of the negative log-likelihood.

Let $\mathbf{H}_{\mathbf{v}} \equiv \langle \mathbf{v} | \mathbf{H} | \mathbf{v} \rangle$. We call $\mathbf{H}_{\mathbf{v}}$ the clamped Hamiltonian, because every visible qubit σ_i^z is clamped to its corresponding classical data value v_i . An application of the Golden-Thompson inequality leads us to $P_{\mathbf{v}} \geq \frac{\text{tr}[e^{-\mathbf{H}_{\mathbf{v}}}]}{\text{tr}[e^{-\mathbf{H}}]}$, and we then can conclude that

$$\tilde{\mathcal{L}} = - \sum_{\mathbf{v}} P_{\mathbf{v}}^{data} \log \left(\frac{\text{tr}[e^{-\mathbf{H}_{\mathbf{v}}}]}{\text{tr}[e^{-\mathbf{H}}]} \right)$$

is an upper bound of the negative log-likelihood function \mathcal{L} . Instead of minimizing \mathcal{L} , we now minimize the upper bound $\tilde{\mathcal{L}}$ using its gradient

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \theta} = \sum_{\mathbf{v}} P_{\mathbf{v}}^{data} \left(\frac{\text{tr}[\Lambda_{\mathbf{v}} \frac{\partial e^{-\mathbf{H}_{\mathbf{v}}}}{\partial \theta}]}{\text{tr}[\Lambda_{\mathbf{v}} e^{-\mathbf{H}_{\mathbf{v}}}] } - \frac{\text{tr}[\frac{\partial e^{-\mathbf{H}}}{\partial \theta}]}{\text{tr}[e^{-\mathbf{H}}]} \right) = \overline{\left\langle \frac{\partial \mathbf{H}_{\mathbf{v}}}{\partial \theta} \right\rangle_{\mathbf{v}}} - \left\langle \frac{\partial \mathbf{H}}{\partial \theta} \right\rangle,$$

where for a matrix \mathbf{A} we define

$$\overline{\langle \mathbf{A} \rangle_{\mathbf{v}}} = \sum_{\mathbf{v}} P_{\mathbf{v}}^{data} \langle \mathbf{A} \rangle_{\mathbf{v}} = \sum_{\mathbf{v}} P_{\mathbf{v}}^{data} \frac{\text{tr}[e^{-\mathbf{H}_{\mathbf{v}}} \mathbf{A}]}{\text{tr}[e^{-\mathbf{H}_{\mathbf{v}}}] }.$$

Therefore, the updating rules for b_i , w_{ij} , Γ_i are given as follows,

$$\Delta b_i = \eta \left(\overline{\langle \sigma_i^z \rangle_{\mathbf{v}}} - \langle \sigma_i^z \rangle \right), \quad \Delta w_{ij} = \eta \left(\overline{\langle \sigma_i^z \sigma_j^z \rangle_{\mathbf{v}}} - \langle \sigma_i^z \sigma_j^z \rangle \right), \quad \Delta \Gamma_i = \eta \left(\overline{\langle \sigma_i^x \rangle_{\mathbf{v}}} - \langle \sigma_i^x \rangle \right),$$

where η is the learning rate of the gradient descent algorithm (Wang and Wu 2020). We can estimate the unclamped terms $\langle \sigma_i^z \rangle$ and $\langle \sigma_i^z \sigma_j^z \rangle$ by sampling from a Boltzmann distribution with Hamiltonian H , and the clamped terms $\overline{\langle \sigma_i^z \rangle_{\mathbf{v}}}$ and $\overline{\langle \sigma_i^z \sigma_j^z \rangle_{\mathbf{v}}}$ by sampling from a Boltzmann distribution with Hamiltonian $H_{\mathbf{v}}$. However, there is a serious issue regarding estimating $\overline{\langle \sigma_i^x \rangle_{\mathbf{v}}}$ and $\langle \sigma_i^x \rangle$. In fact, we need measurements in the σ_i^x basis to estimate $\langle \sigma_i^x \rangle$, but we cannot estimate $\langle \sigma_i^x \rangle$ by sampling in the σ_i^z basis, as non-diagonal matrix σ_i^x does

not commute with σ_i^z . As a matter of fact, for all visible variables \mathbf{v} , we have $\overline{\langle \sigma_i^x \rangle_{\mathbf{v}}} = 0$. Note that $\langle \sigma_i^x \rangle > 0$ for positive Γ_i and $\langle \sigma_i^x \rangle < 0$ for negative Γ_i ; thus we conclude that $\Delta\Gamma_i < 0$ if $\Gamma_i > 0$ and $\Delta\Gamma_i > 0$ if $\Gamma_i < 0$. This renders an invalid training of Γ_i using the updating rule. One possible ad hoc fix is to treat the b_i 's and w_{ij} 's as trainable parameters and regard the Γ_i 's as super-parameters, or even set all the Γ_i 's to be equal to a fixed value.

As we have described in Section 5, quantum annealing driven by the same quantum Hamiltonian \mathbf{H} can be implemented by quantum annealers like the D-Wave machine. It turns out that a quantum annealer can provide sample from the Boltzmann distribution of the Hamiltonian and train a QBM to tune the model parameters $\theta = (\Gamma_i, b_i, w_{ij})$. More details can be found in [Amin et al. \(2018\)](#) and [Kieferova and Wiebe \(2016\)](#).

7. Concluding Remarks

Quantum computation has attracted enormous attention in frontiers of sciences. While the predominance of quantum computation is the invention of faster quantum algorithms and the creation of quantum computers to demonstrate quantum advantages and implement the quantum algorithms for accomplishing hard computational or communication tasks, this paper provides an overview on the statistical aspect of quantum computation to illustrate the interaction between statistics and quantum computation. This stands in contrast to classical computation where there is little role for statistics to play in its deterministic platform. We introduce important quantum concepts and key quantum properties for quantum computation. We review quantum annealing, quantum machine learning with Boltzmann machines and quantum supremacy via boson sampling and random quantum circuits. Our feature of the selected topics focuses on the use of quantum computation in statistical machine learning and applications of statistical analysis to resolve issues encountered in quantum computation as well as their interplay that may demonstrate quantum advantage and/or lead to new theories, methodologies and computational techniques for statistics and machine learning. In particular we present quantum computation and illustrate its interface with statistics and data science; and we highlight the advantages of quantum computation and quantum learning for statistics and machine learning in terms of computational complexity and learning efficiency. There is a great demand for the certification of quantum devices such as testing and assessing their quantum performances, and such certification needs sound and scalable statistical methods for calibrating and validating quantum properties. In fact, a quantum computation endeavor such as quantum supremacy calls for an integration of new experimental techniques, better mathematical and statistical modeling, and improved computational tools where statistics and data science can play a major role ([Wang 2012](#); [Wang et al. 2016](#); [Wang and Song 2020](#); [Hu and Wang 2021](#); [Wang 2021](#)). For example, for the study of quantum supremacy, we need to repeat computing experiments, reanalyze observed data, and address or close potential loopholes. Classical algorithms and computer power continue to be improved, and what is impractical for classical computers today may become tractable in the future. At the same time the computational power of quantum computers will keep growing. Hence the benchmark of the classical computational cost is a moving target, and the quantum supremacy frontier will be moving towards larger and larger computational problems to herald a much anticipated computing paradigm that will ultimately offer the large-scale computational platform to run well known quantum algorithms, such as the Shor and Grover algorithms. As shown in Section 4, we expect a high demand for statistics in the continuing study on quantum supremacy.

ACKNOWLEDGMENTS

The research of Yazhen Wang was supported in part by NSF grants DMS-1528735, DMS-1707605, and DMS-1913149. The authors thank the Editorial Committee for the comments and suggestions that led to improvements of the paper.

LITERATURE CITED

- Aaronson, S. and Arkhipov, A. (2011). The computational complexity of linear optics. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, volume 10 of *STOC '11*, pages 333–342, New York, NY, USA. Association for Computing Machinery.
- Aaronson, S. and Chen, L. (2016). Complexity-theoretic foundations of quantum supremacy experiments. *arXiv preprint arXiv:1612.05903*.
- Adachi, S. H. and Henderson, M. P. (2015). Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*.
- Albash, T., Rønnow, T. F., Troyer, M., and Lidar, D. A. (2015). Reexamining classical and quantum models for the d-wave one processor. *The European Physical Journal Special Topics*, 224(1):111–129.
- Amin, M. H., Andriyash, E., Rolfe, J., Kulchitsky, B., and Melko, R. (2018). Quantum Boltzmann machine. *Physical Review X*, 8(2):021050.
- Artiles, L. M., Gill, R. D., and Guță, M. I. (2005). An invitation to quantum tomography. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):109–134.
- Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G. S. L., Buell, D. A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., Fowler, A., Gidney, C., Giustina, M., Graff, R., Guerin, K., Habegger, S., Harrigan, M. P., Hartmann, M. J., Ho, A., Hoffmann, M., Huang, T., Humble, T. S., Isakov, S. V., Jeffrey, E., Jiang, Z., Kafri, D., Kechedzhi, K., Kelly, J., Klimov, P. V., Knysh, S., Korotkov, A., Kostritsa, F., Landhuis, D., Lindmark, M., Lucero, E., Lyakh, D., Mandr, S., McClean, J. R., McEwen, M., Megrant, A., Mi, X., Michielsen, K., Mohseni, M., Mutus, J., Naaman, O., Neeley, M., Neill, C., Niu, M. Y., Ostby, E., Petukhov, A., Platt, J. C., Quintana, C., Rieffel, E. G., Roushan, P., Rubin, N. C., Sank, D., Satzinger, K. J., Smelyanskiy, V., Sung, K. J., Trevithick, M. D., Vainsencher, A., Villalonga, B., White, T., Yao, Z. J., Yeh, P., Zalcman, A., Neven, H., and Martinis, J. M. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510.
- Barndorff-Nielsen, O. E., Gill, R. D., and Jupp, P. E. (2003). On quantum statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(4):775–804.
- Benedetti, M., Realpe-Gómez, J., Biswas, R., and Perdomo-Ortiz, A. (2016). Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning. *Physical Review A*, 94(2):022308.
- Bertsimas, D. and Tsitsiklis, J. (1993). Simulated annealing. *Statistical science*, 8(1):10–15.
- Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., and Lloyd, S. (2017). Quantum machine learning. *Nature*, 549(7671):195.
- Boixo, S., Isakov, S. V., Smelyanskiy, V. N., Babbush, R., Ding, N., Jiang, Z., Bremner, M. J., Martinis, J. M., and Neven, H. (2018). Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595.
- Boixo, S., Rønnow, T. F., Isakov, S. V., Wang, Z., Wecker, D., Lidar, D. A., Martinis, J. M., and Troyer, M. (2014). Evidence for quantum annealing with more than one hundred qubits. *Nature Physics*, 10(3):218.
- Boixo, S., Smelyanskiy, V. N., Shabani, A., Isakov, S. V., Dykman, M., Denchev, V. S., Amin, M. H., Smirnov, A. Y., Mohseni, M., and Neven, H. (2016). Computational multiqubit tunnelling in programmable quantum annealers. *Nature communications*, 7:10327.

- Bouland, A., Fefferman, B., Nirkhe, C., and Vazirani, U. (2018). Quantum supremacy and the complexity of random circuit sampling. *arXiv preprint arXiv:1803.04402*.
- Brady, L. T. and van Dam, W. (2016). Quantum Monte Carlo simulations of tunneling in quantum adiabatic optimization. *Physical Review A*, 93(3):032304.
- Brooke, J., Bitko, D., and Aeppli, G. (1999). Quantum annealing of a disordered magnet. *Science*, 284(5415):779–781.
- Cai, T., Kim, D., Wang, Y., Yuan, M., and Zhou, H. H. (2016). Optimal large-scale quantum state tomography with pauli measurements. *The Annals of Statistics*, 44(2):682–712.
- Ciliberto, C., Herbster, M., Ialongo, A. D., Pontil, M., Rocchetto, A., Severini, S., and Wossnig, L. (2018). Quantum machine learning: A classical perspective. *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2209):20170551.
- Dunjko, V. and Briegel, H. J. (2018). Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001.
- Farhi, E., Goldstone, J., and Gutmann, S. (2002). Quantum adiabatic evolution algorithms versus simulated annealing. *arXiv preprint quant-ph/0201031*.
- Farhi, E., Goldstone, J., Gutmann, S., Lapan, J., Lundgren, A., and Preda, D. (2001). A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science*, 292(5516):472–475.
- Farhi, E., Goldstone, J., Gutmann, S., and Sipser, M. (2000). Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*.
- Fischer, A. and Igel, C. (2012). An introduction to restricted Boltzmann machines. In *CIARP*, pages 14–36.
- Fischer, A. and Igel, C. (2014). Training restricted Boltzmann machines: An introduction. *Pattern Recognition*, 47:25–39.
- Hamilton, C. S., Kruse, R., Sansoni, L., Barkhofen, S., Silberhorn, C., and Jex, I. (2017). Gaussian boson sampling. *Phys. Rev. Lett.*, 119:170501.
- Harrow, A. W. and Montanaro, A. (2017). Quantum computational supremacy. *Nature*, 549(7671):203.
- Hinton, G. and Salakhutdinov, R. (2012). A better way to pretrain deep Boltzmann machines. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 2447–2455. Curran Associates, Inc.
- Holevo, A. S. (2001). *Statistical structure of quantum theory*. Springer.
- Hu, J. and Wang, Y. (2021). Quantum annealing via path-integral Monte Carlo with data augmentation. *Journal of Computational and Graphical Statistics*.
- Isakov, S. V., Mazzola, G., Smelyanskiy, V. N., Jiang, Z., Boixo, S., Neven, H., and Troyer, M. (2016). Understanding quantum tunneling through quantum Monte Carlo simulations. *Physical review letters*, 117(18):180402.
- Jörg, T., Krzakala, F., Kurchan, J., and Maggs, A. C. (2010). Quantum annealing of hard problems. *Progress of Theoretical Physics Supplement*, 184:290–303.
- Kieferova, M. and Wiebe, N. (2016). Tomography and generative data modeling via quantum Boltzmann training. *arXiv preprint arXiv:1612.05204*.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.
- Lund, A., Bremner, M. J., and Ralph, T. (2017). Quantum sampling problems, bosonsampling and quantum supremacy. *npj Quantum Information*, 3(1):15.
- Markov, I. L., Fatima, A., Isakov, S. V., and Boixo, S. (2018). Quantum supremacy is both closer and farther than it appears. *arXiv preprint arXiv:1807.10749*.
- McGeoch, C. C. (2014). Adiabatic quantum computation and quantum annealing: Theory and practice. *Synthesis Lectures on Quantum Computing*, 5(2):1–93.
- Neill, C., Roushan, P., Kechedzhi, K., Boixo, S., Isakov, S. V., Smelyanskiy, V., Megrant, A., Chiaro, B., Dunsworth, A., and Arya, K. (2018). A blueprint for demonstrating quantum supremacy with

- superconducting qubits. *Science*, 360(6385):195–199.
- Nielsen, M. A. and Chuang, I. L. (2010). *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition.
- Parthasarathy, K. R. (2012). *An introduction to quantum stochastic calculus*, volume 85. Birkhäuser.
- Petz, D. (2008). *Quantum information theory and quantum statistics*. Springer.
- Quesada, N., Arrazola, J. M., and Killoran, N. (2018). Gaussian boson sampling using threshold detectors. *Phys. Rev. A*, 98:062322.
- Rinott, Y., Shoham, T., and Kalai, G. (2020). Statistical aspects of the quantum supremacy demonstration. *arXiv preprint arXiv:2008.05177*.
- Rønnow, T. F., Wang, Z., Job, J., Boixo, S., Isakov, S. V., Wecker, D., Martinis, J. M., Lidar, D. A., and Troyer, M. (2014). Defining and detecting quantum speedup. *Science*, 345(6195):420–424.
- Sakurai, J. and Napolitano, J. (2017). Modern quantum mechanics. *Modern Quantum Mechanics, by JJ Sakurai, Jim Napolitano, Cambridge, UK: Cambridge University Press, 2017*.
- Salakhutdinov, R. (2015). Learning deep generative models. *Annual Review of Statistics and Its Application*, 2(1):361–385.
- Salakhutdinov, R. and Hinton, G. (2009). Deep Boltzmann machines. In van Dyk, D. and Welling, M., editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 448–455, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA. PMLR.
- Salakhutdinov, R. and Hinton, G. (2012). An efficient learning procedure for deep Boltzmann machines. *Neural Computation*, 24(8):1967–2006.
- Shankar, R. (2012). *Principles of quantum mechanics*. Springer Science & Business Media.
- Sinervo, P. K. (2003). Definition and treatment of systematic uncertainties in high energy physics and astrophysics. In Lyons, L., Mount, R., and Reitmeyer, R., editors, *Proceedings of the Conference on Statistical Problems in Particle Physics, Astrophysics and Cosmology (PHYSTAT2003)*, PHYSTAT2003, pages 122–129, Stanford, California, USA. Stanford Linear Accelerator Center.
- Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, second edition.
- Wang, Y. (2012). Quantum computation and quantum information. *Statistical Science*, 27(3):373–394.
- Wang, Y. (2021). When quantum computation meets data science: Making data science quantum. *Harvard Data Science Review*.
- Wang, Y. and Song, X. (2020). Quantum science and quantum technology. *Statistical Science*, 35(1):51–74.
- Wang, Y. and Wu, S. (2020). Asymptotic analysis via stochastic differential equations of gradient descent algorithms in statistical and computational paradigms. *Journal of Machine Learning Research*, 21(199):1–103.
- Wang, Y., Wu, S., and Zou, J. (2016). Quantum annealing with Markov chain Monte Carlo simulations and D-Wave quantum computers. *Statistical Science*, 31(3):362–398.
- Wang, Y. and Xu, C. (2015). Density matrix estimation in quantum homodyne tomography. *Statistica Sinica*, pages 953–973.
- Wiebe, N., Kapoor, A., and Svore, K. M. (2014). Quantum deep learning. *arXiv:1412.3489*.
- Wittek, P. (2014). *Quantum Machine Learning: What Quantum Computing Means to Data Mining*. Academic Press.
- Zhong, H.-S., Wang, H., Deng, Y.-H., Chen, M.-C., Peng, L.-C., Luo, Y.-H., Qin, J., Wu, D., Ding, X., Hu, Y., Hu, P., Yang, X.-Y., Zhang, W.-J., Li, H., Li, Y., Jiang, X., Gan, L., Yang, G., You, L., Wang, Z., Li, L., Liu, N.-L., Lu, C.-Y., and Pan, J.-W. (2020). Quantum computational advantage using photons. *Science*.