A comprehensive review of Quantum Machine Learning: from NISQ to Fault Tolerance

Yunfei Wang

Martin A. Fisher School of Physics, Brandeis University, Waltham MA 02453, USA

E-mail: `yunfeiwang0214@gmail.com`

Junyu Liu

Chicago Quantum Exchange, Chicago, IL 60637, USA
Department of Computer Science, The University of Chicago, Chicago, IL 60637, USA
Pritzker School of Molecular Engineering, The University of Chicago, Chicago, IL 60637, USA
Kadanoff Center for Theoretical Physics, The University of Chicago, Chicago, IL 60637, USA

E-mail: `junyuliu@uchicago`

Abstract. Quantum machine learning, which involves running machine learning algorithms on quantum devices, has garnered significant attention in both academic and business circles. In this paper, we offer a comprehensive and unbiased review of the various concepts that have emerged in the field of quantum machine learning. This includes techniques used in Noisy Intermediate-Scale Quantum (NISQ) technologies and approaches for algorithms compatible with fault-tolerant quantum computing hardware. Our review covers fundamental concepts, algorithms, and the statistical learning theory pertinent to quantum machine learning.

Contents

1. Introduction

Quantum machine learning represents a highly promising realm in contemporary physics and computer science research, with far-reaching implications spanning quantum chemistry [108], artificial intelligence [89], and even high-energy physics [7]. Nevertheless, it remains in its nascent stages of development. This is evident from the absence of a precise definition for quantum machine learning. Some describe it as the convergence of quantum computing and machine learning, wherein machine learning algorithms are executed on quantum devices. In simpler terms, it can be thought of as the quantum counterpart to classical machine learning.

In recent times, artificial intelligence, exemplified by technologies like `ChatGPT`, has become an integral part of everyday life. It's entirely plausible that, in the future, we will harness artificial intelligence in an even wider array of applications, including medical diagnostics, education, and aiding scientific research. Much of artificial intellegence's success

hinges on machine learning, a departure from traditional computer programs that entail crafting explicit instructions to directly solve problems [112]. Machine learning models, on the other hand, are trained on real-world data stored in a dataset (often denoted as $\mathcal{D}$), acquiring the ability to tackle problems autonomously. This represents a departure, in a sense, from the conventional von Neumann model of digital computing.

Researchers might initially wonder why there's a need for quantum machine learning when its classical counterpart has demonstrated impressive performance. One of the primary rationales stems from the fact that classical machine learning relies heavily on linear algebra procedures [18]. For instance, classical machine learning can address problems like identifying the optimal hyperplane to separate two data clusters by employing matrix inversion techniques. Quantum mechanics, at its core, is inherently grounded in linear algebra, where we construct a Hilbert space ($\mathcal{H}$) of a system described by an Hermitian operator known as the Hamiltonian ($\hat{H}$) [114]. Over time, it has become evident that quantum computing has the potential to dramatically enhance a computer's problem-solving capabilities in certain specific scenarios. To illustrate, consider the task of matrix inversion again. Classical computers typically require computational efforts with a complexity of $\mathcal{O}(N \log N)$ to accomplish this, while quantum computers can leverage algorithms like the Harrow-Hassidim-Lloyd algorithm (HHL) algorithm [58], which has a complexity of $\mathcal{O}((\log N)^2)$, leading to significant speedup, in certain conditions like sparsity, fault-tolerance, small condition numbers, and fast interfaces between classical and quantum processors. Although there is no known explicit realizations of HHL algorithm at the large scale at the moment, it is still an exciting direction deserves further studies, especially when quantum devices are fast developing with more and more capabilities.

Due to its vague definition, quantum machine learning encompasses a broader spectrum of topics. For instance, quantum (shadow) tomography [2] has gained prominence, focusing on the characterization of a given quantum state by accumulating data from various measurements. This involves determining the minimum number of identical quantum state copies needed to extract sufficient information about the state's properties. Another facet involves machine learning for quantum physics, which entails employing (perhaps classical) machine learning tools to explore various aspects of quantum physics. Additionally, some developments in both quantum algorithms and quantum hardware are often encompassed within the broader umbrella of quantum machine learning.

Although these topics may not directly align with the narrow definition of quantum machine learning outlined earlier, they hold substantial promise for its future development. To illustrate, within the narrow sensed field of quantum machine learning, a significant challenge is known as the input/output problem. In particular, the output problem pertains to the task of comprehending the solution generated by a quantum algorithm, and this aligns closely with the domain of shadow tomography. And by harnessing the capabilities of classical machine learning to gain deeper insights into quantum physics, it might lead to advancements in quantum computing too. In this review, we primarily focus on quantum machine learning in its narrower sense, which pertains to execute quantum algorithms designed for machine learning purposes.

Presently, we find ourselves in what's referred to as (perhaps the end of) the *noisy intermediate scale quantum* or NISQ era of quantum computing. Quantum computers are susceptible to background noise, which imposes limitations on our ability to construct quantum computers with sufficient depth for executing tasks demanding fast and precise computations. The quantum computers available today can only handle on the order of around 100 qubits, and they all exhibit noise, making it challenging to derive tangible benefits for our daily lives.

The solution to this predicament is known as quantum error correction (QEC) code [113]. Think of QEC as a safeguard for quantum information. Typically, quantum information is lost once it's measured, which becomes especially likely in noisy environments. However, information protected by QEC can persist if it remains undamaged within certain limits. It's worth noting that all error correction codes have their constraints, implying that information will inevitably be lost if it's severely damaged. Nevertheless, error correction provides a protective buffer zone against such losses.

Our objective is to implement QEC codes across all quantum devices, ushering in an era of *fault-tolerant quantum computing* (FTQC) in the future. This trajectory parallels the history of classical computing, where, before the invention of classical error correction, scaling up and running useful algorithms on classical computers was also a formidable challenge [29]. Today, we can reliably operate classical computers everyday. Given the promising advancements in quantum error correction in recent times, our optimism about the future of quantum computing remains steadfast.

However, when it comes to machine learning, classical machine learning doesn't inherently reject noise [88]. The widely recognized learning algorithm, known as stochastic gradient descent, explicitly incorporates noise, and surprisingly, this noise addition actually enhances its performance. To grasp this concept, consider that noise can effectively steer us away from saddle points, offering an automatic mechanism for avoiding them. In a way, one can interpret this as machine learning's ability to withstand and even benefit from noise. Consequently, this insight suggests that running certain machine learning algorithms on current (NISQ) quantum devices could have some significance. This prospect promises to enrich our present-day experiences significantly, especially considering the challenges associated with constructing QEC, which could take a few years to become fully integrated. Prior to entering the era of fault-tolerant quantum computing, we'll have the opportunity to experiment with quantum devices and apply machine learning techniques, adding a vibrant dimension to our current scientific research.

In addition to discussing the current applications of quantum capabilities for machine learning, we should also let our imaginations soar. The era of fault-tolerant quantum computing (FTQC) is a foreseeable future, and it's crucial to look ahead at what challenges we can tackle with quantum machine learning on FTQC devices. One of the most renowned and valuable algorithms in this context is the Harrow-Hassidim-Lloyd (HHL) algorithm [58]. Additionally, there are other algorithms that can be deployed to address a range of problems, such as principal component analysis [18].

Beside the optimistic future that quantum machine learning has, there are also a number

of controversial issues with the subject. For example, some might argue that the variational quantum algorithm will not work in some circumstance. People working on quantum landscapes theory observed the famous barren plateau phenomena which leads to a kind of no-go theorem [27, 86]. It amounts to situations that are hard to find minimum when we optimizes our objective function by gradient descent with randomized variational circuits.

Debates indeed exist regarding whether quantum algorithms can consistently deliver exponential speedups. Some argue that quantum speedup is only guaranteed when dealing with quantum information. When it comes to classical information, it's conceivable to design classical machine learning models capable of achieving comparable average prediction accuracy [121, 66]. In such cases, the computational complexity difference between classical and quantum approaches may, at worst, be a modest polynomial factor. This debate underscores the importance of carefully assessing the specific problem and context when considering the potential advantages of quantum algorithms. However, the landscape shifts when the objective is to attain a low worst-case prediction error. In this scenario, it becomes feasible to achieve an exponential divergence in computational complexities between classical and quantum approaches. This underscores the potential superiority of quantum algorithms when stringent requirements for worst-case accuracy are in play. This phenomenon can be viewed as a manifestation of classical effects casting their "shadow" in the quantum realm. It should come as no surprise to individuals well-versed in quantum theory that quantum predictions tend to align with classical theory when dealing with systems possessing a high degree of freedom. This approach can also be applied to tackle a critical issue in variational quantum algorithms known as the "output problem," often referred to as the classical shadow.

Indeed, it is truly heartening and captivating to witness the evolution of quantum computing beyond celebrated algorithms like Shor's factoring algorithm [117] and Grover search [50]. Algorithms like HHL [58] offer a versatile blueprint, shedding light on how quantum computers hold the potential to deliver significant acceleration for fundamental tasks like clustering, pattern-matching, and principal component analysis. However, as Scott Aaronson aptly noted in his paper [1], the pursuit of these speedups will still demand significant effort and ingenuity, as nature continues to present challenges that compel us to work diligently for these advancements.

We have structured our review as follows. In section 2, we focus on the historical and ongoing developments in Quantum Machine Learning (QML) during the Noisy Intermediate-Scale Quantum (NISQ) era. Within this era, one of the central frameworks is the Variational Quantum Algorithm (VQA) [17, 44, 108], which we explore in detail in section 2.1. VQA comprises four key elements: the choice of an objective function (discussed in section 2.1.1), the employment of Parameterized Quantum Circuits (PQC) with adjustable parameters optimized by classical algorithms (covered in section 2.1.2), measurement strategies (explored in section 2.1.3), and the classical optimizer responsible for minimizing the objective function (explained in section 2.1.4). Subsequently, we provide insights into the construction of the Quantum Neural Tangent Kernel (QNTK) [87] in section 2.2, offering a theoretical foundation for quantum neural networks and an understanding of stochastic gradient descent dynamics from first principles. Finally, we address the issue of barren plateaus in section 2.3,

approaching this topic through the lens of quantum landscape theory [27]. Additionally, we present an alternative formulation of *laziness* [86], providing another perspective on the same problem.

In section 3, our focus shifts to quantum algorithms that have the potential for exponential speedup in the Fault-Tolerant Quantum Computing (FTQC) era. We begin by introducing Quantum Phase Estimation (QPE) [29] in section 3.1, where we also delve into the Quantum Principal Component Analysis (QPCA) program [91]. Subsequently, we present a counterpoint regarding the attainability of exponential speedup for programs such as the recommendation system in section 3.2. Moving on to section 3.3, we introduce the pivotal Harrow-Hassidim-Lloyd (HHL) algorithm [58], which opens up possibilities for various new quantum machine learning algorithms. In section 3.3.1, we illustrate a visionary perspective on the future of machine learning algorithms with the aid of Carleman linearization [90] and the HHL algorithm. Finally, in section 3.4, we introduce Quantum Random Access Memory (QRAM), focusing on theoretical designs [56], practical implementation efforts [54], and its necessity in certain algorithms [18].

In section 4, we delve into various topics that amalgamate quantum principles with statistical learning theory. Our primary focus is on shadow tomography [2] in section 4.1, where we explore the motivation behind shadow tomography and delve into the construction of the theorem. A pivotal subject inspired by shadow tomography, the classical shadow formalism [41], is thoroughly discussed in section 4.2. Moving on to section 4.2.1, we examine the application of classical shadow as an efficient quantum-to-classical information converter [66]. In section 4.3, we then shift our attention to the applications of Quantum Machine Learning (QML) in the study of quantum data and quantum simulators [18].

It's essential to note that our review doesn't encompass various topics, including quantum machine learning algorithms whose advantages stem from sampling-related statements, the studies of quantum error corrections, quantum memory, and designs of quantum data centers.

2. Noisy intermediate-scale quantum (NISQ) era

The concept of the NISQ era, which stands for Noisy Intermediate-Scale Quantum, was introduced by John Preskill in 2018 [110]. It's essential to understand that NISQ is primarily a hardware-oriented definition and doesn't inherently imply a specific time frame. In NISQ devices, quantum circuits are implemented, where all gates adhere to a predefined graph G, with qubits represented by nodes in this graph. These gates typically operate on one or two qubits. Due to the noise associated with each gate operation, NISQ algorithms are naturally constrained to shallow circuit depths [12].

On the other hand, "near-term" algorithms are those designed for quantum devices expected to be available in the next few years, and this term doesn't explicitly refer to the absence of Quantum Error Correction (QEC). The notion of what constitutes the "near-term" can be subjective because different researchers may have varying opinions on how many years can be considered "near-term." Predicting experimental progress in quantum computing is always challenging, and such predictions can be influenced by human biases. Algorithms

developed for near-term hardware may become unfeasible if hardware advancements do not align with the algorithm's experimental requirements [17].

The majority of existing NISQ algorithms leverage quantum computers' capabilities through a hybrid quantum-classical approach. These algorithms involve offloading the classically challenging aspects of certain computations to a quantum computer while performing the remaining tasks on a suitably powerful classical device. They operate by iteratively adjusting the parameters of a parametrized quantum circuit, making them known as Variational Quantum Algorithms (VQA) ([42, 26]).

2.1. Variational quantum algorithm (VQA)

The concept of Variational Quantum Algorithms (VQA) was initially introduced to address certain quantum chemistry challenges, notably through the well-known Variational Quantum Eigensolver (VQE) [97, 108, 131]. Additionally, VQA includes the Quantum Approximate Optimization Algorithm (QAOA) [44], which is designed for solving combinatorial optimization problems. Interestingly, reference [5] shows that VQE combined with filtering operators can be more efficient and accurate than standard VQE and QAOA in the context of combinatorial optimization.

It's crucial to emphasize that, as of now, there is no established quantum advantage for Variational Quantum Algorithms (VQA) employing NISQ devices. As previously outlined, VQA can be divided into four primary components: 1) the objective function; 2) the parameterized quantum circuit (PQC); 3) the measurement scheme; and 4) the classical optimizer. This breakdown is visually represented in FIG. 1.
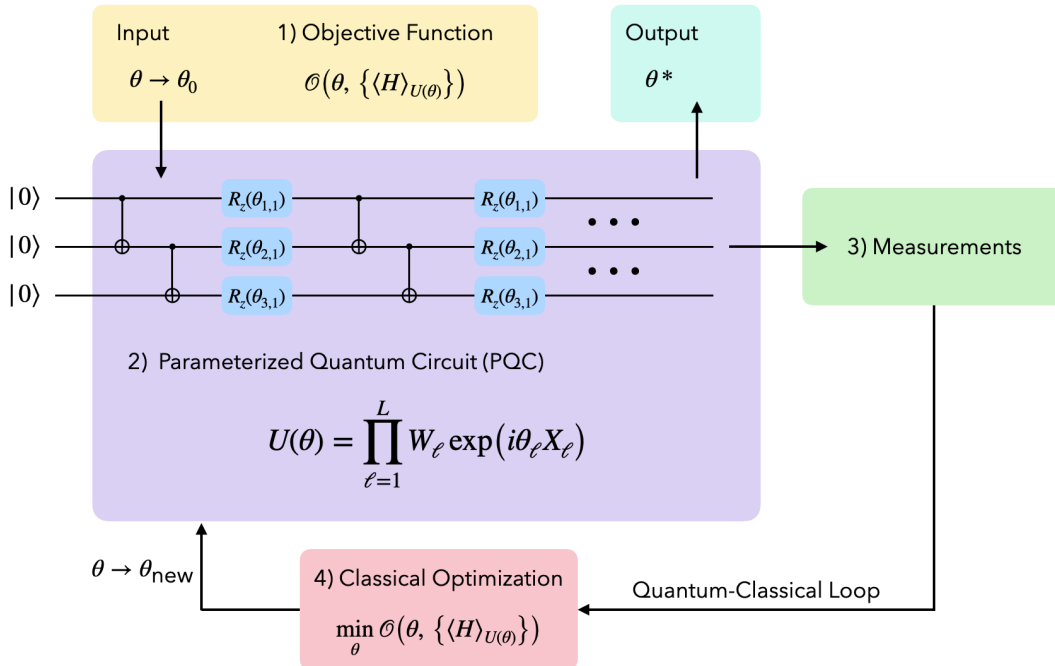

Figure 1. VQA in 4 parts.

2.1.1. Objective function The Hamiltonian, a Hermitian operator, plays a pivotal role in governing the time evolution of a physical system through Schrödinger equation. Its expectation value provides the energy associated with a quantum state, making it a common target for minimization in VQA, often used to find the ground state of a system. However, when we aim to solve problems unrelated to real physical systems, we can construct a Hamiltonian that encodes the information we seek to extract.

In the context of VQA, the specific choice of the Hamiltonian is not particularly significant. One can select any Hermitian operator and designate it as the Hamiltonian, then attempt to find the ground state. However, an essential property that the operator must possess is the ability to decompose into several operators that can be measured by the quantum processor. We will explore this concept in more detail in section 2.1.3.

Constructing the objective function $\mathcal{O}$, it is natural to employ the expectation value of the Hamiltonian:

$$\langle H \rangle_{U(\theta)} \equiv \langle 0 | U^\dagger(\theta) H U(\theta) | 0 \rangle \ . \tag{1}$$

It's important to note that we initialize our qubits in the state $|0\rangle$ and subsequently evolve them using the parameterized quantum circuit (PQC). As shown in Eqn. 1, the objective function depends on the unitary time evolution operator, making it a function of classical tunable parameters denoted as θ. In some situations, it can be advantageous to utilize objective functions other than the Hamiltonian expectation value. Consequently, we have some flexibility in the specific form of the objective function, but we require that the objective function $\mathcal{O}$ be a function of θ and $\langle H \rangle_{U(\theta)}$, expressed as $\mathcal{O}\big(\theta\,, \{\langle H \rangle_{U(\theta)}\}\big)$.

In fact, the selection of the objective function is a critical factor in achieving the desired convergence in VQA. Barren plateaus, where gradients vanish, are closely tied to the choice of the cost function. Therefore, a poor choice can result in a failure to locate a minimum of the objective function through gradient descent.

(i) **Pauli strings**

   The most natural approach to representing our Hamiltonian involves expressing it as a linear combination of basic tensor products of Pauli matrices: σ_i, with σ_i taking values from the set $\{I, \sigma_x, \sigma_y, \sigma_z\}$. These tensor products are commonly referred to as Pauli strings, defined as $\hat{P} = \otimes_{j=1}^n \sigma_i^{(j)}$, where the index j corresponds to different qubits. This representation is well-suited because the Pauli matrices σ_i collectively form a basis for all Hermitian 2 by 2 matrices and serve as the generators of the special unitary group SU(2). The Hamiltonian, then, can be decomposed as $H = \sum_{k=1}^M c_k \hat{P}_k$. In this decomposition, the complex coefficient c_k pertains to the k-th Pauli string, and the total number of Pauli strings, denoted as M, is contingent upon the specific form of the Hamiltonian under consideration.

(ii) **Fidelity**

   Rather than optimizing with respect to the expectation value of an operator, certain VQAs necessitate a subroutine for optimizing the state obtained from the PQC $U(\theta)$, denoted as $|\Psi\rangle_{U(\theta)}$, with respect to a target state, denoted as $|\Psi\rangle$. We then define fidelity as

the following: $F\big(\Psi, \Psi_{U(\theta)}\big) \equiv |\langle\Psi|\Psi_{U(\theta)}\rangle|^2 = \langle\mathrm{Proj}_\Psi\rangle_{U(\theta)}$, which is the expectation value of the projection operator on the target state Ψ: $\mathrm{Proj}_\Psi = |\Psi\rangle\langle\Psi|$. Conventionally, VQA involves minimizing the objective function. In this context, it is natural to use infidelity, represented as $1 - F\big(\Psi, \Psi_{U(\theta)}\big)$, or alternatively, the negative fidelity, denoted as $-\langle\mathrm{Proj}_\Psi\rangle_{U(\theta)}$.

The expression of objective functions based on fidelities is commonly utilized in state preparation algorithms in quantum optics [76, 78] and quantum machine learning [15, 31, 64, 107].

Additional objective functions can also be developed, and any cost function expressed operationally can be considered a viable option [17]. Two examples include the Conditional-Value-at-Risk (CVaR) [13] and the Gibbs objective function [83]. These two forms can be simplified to the mean energy, denoted as $\langle H \rangle$. Importantly, their performance is at least as effective as utilizing the mean energy [83, 13]. The authors of [94] extend VQE to be applicable to a large class of nonlinear problems, including nonlinear differential equations, and show, for example, that quantum circuits can be exponentially more efficient (in terms of expressivity) than classical matrix product states.

2.1.2. Parameterized quantum circuits (PQC) Next, we delve into the second critical component of a VQA, which is the quantum circuit responsible for generating the state that optimally satisfies the defined objective. As described above, it prepares the state by an act of unitary that depends on classical parameters: θ , hence, it is called the parameterized quantum circuit: PQC.

The state prepared is defined as the following: $|\Psi(\theta)\rangle = U(\theta)|\Psi_0\rangle$, where $|\Psi_0\rangle$ is some initial state, for example, $|0\rangle^{\otimes n}$, where n is the number of qubits. As remarked earlier, n should be quite small, due to the limited ability of the NISQ devices, we can only accommodate very shallow circuits. In some VQAs, it is convenient to use other initialization states. We can use a unitary operator (for example, a rotation operator) $P(\phi)$, and the initial state: $|\Psi_0\rangle = P(\phi)|0\rangle^{\otimes n}$.

Similar to the variational principle in quantum wave mechanics, which aims to obtain the wave function with the lowest possible energy, a deep understanding of the physical system can significantly assist us in creating a better initial guess. For instance, if we seek an approximation of the ground state of a potential that exhibits even parity, it's advisable to start with an initial guess that also possesses even parity. It's important to recognize that the choice of the initial state can significantly impact the performance of a VQA in terms of convergence speed and how closely the final state approximates the optimal solution for the problem at hand. This influence is particularly evident in a simple example. Hence, in most cases, your initial step should involve examining the Hamiltonian or the observable of interest to identify any inherent symmetries.

In the NISQ era, we face significant limitations in terms of the number of qubits we can effectively use. Therefore, it's essential to consider the quantum hardware on which our VQA runs. Constructing problem-inspired initial states can be computationally expensive

when using native gates. Consequently, most ansätze developed so far fall into one of two categories: problem inspired ansätze and hardware efficient ansätze [17]. The categorization depends on the structure and intended application of the ansätze. The PQC shown in FIG. 2 and FIG. 1 are examples of the second kind, hardware efficient ansätz. Note that these two circuits are also the same circuit for quantum neural network. We will now discuss both of them below.

(i) **Problem-inspired ansätze**

In the problem-inspired ansätze approach, we utilize the evolution by a unitary operator, denoted as $G(t)$, with a Hermitian generator $\hat{g}$: $G(t) = e^{-i\hat{g}t}$. These operators are derived from the properties or symmetries of the system under consideration and are used in constructing the PQC. Implementing such unitary transformations directly can be challenging. Therefore, one often employs a technique called the *Suzuki-Trotter (ST) expansion or decomposition* [120]. This method provides a general approach to approximate complex and challenging-to-implement unitary transformations. In practice, the more we know about the Hamiltonian to be *Trotterized*, the less number of gates is needed to implement this method [74]. Two mostly used ansätze of this kind is the *unitary coupled cluster* (UCC) for quantum chemistry and the *quantum approximate optimization algorithm* (QAOA).

(a) *Unitary Coupled Cluster (UCC)*

Historically, the Unitary Coupled Cluster (UCC) ansätze were among the earliest proposed and implemented approaches, as discussed in reference [123]. The core concept behind UCC is to introduce quantum correlations into the Hartree-Fock approximation. Interestingly, in 2014, it was demonstrated that representing UCC on a classical computer is computationally inefficient [137]. However, during the same year, researchers took advantage of quantum resources and successfully realized the UCC ansätze as a PQC using a photonic processor [108, 82, 49, 17].

(b) *Quantum Approximate Optimization Algorithm (QAOA)*

The Quantum Approximate Optimization Algorithm (QAOA) shares strong connections with quantum annealing and adiabatic quantum computing. Adiabatic quantum computing, in brief, leverages the adiabatic theorem in quantum mechanics, which asserts that a quantum state undergoing slow variations in a time-dependent Hamiltonian will evolve only by a phase. Consequently, one can devise a Hamiltonian that evolves over time, transitioning from an easily prepared system to a system capable of encoding problems to be solved. The energy eigenstate naturally evolves from the initially prepared state to one that solves the target problem. QAOA can be viewed as an approach that resembles Trotterized adiabatic quantum computing [17, 44].

For combinatorial optimization problems featuring strict constraints, you can introduce penalty terms in the cost function. However, this approach may not be the most efficient in practice since it can still lead to solutions that violate certain hard constraints. A variant of QAOA aimed at addressing these constraints is also

discussed in previous works [44, 17].

(ii) **Hardware efficient ansätze**

While computational studies have demonstrated that problem inspired ansätze can yield rapid convergence to a satisfactory solution state, their experimental realization can present challenges. Quantum computing devices are subject to various experimental limitations, including factors like coherence time, limited gate fidelity, and constrained gate set. To give an example on how bad the situation is, [100] illustrates that current quantum hardware is not well-suited for implementing deep and highly connected circuits, which are essential for approaches like the UCC and similar methods designed for applications beyond basic demonstrations, such as simulating more complex molecules like Hydrogen molecule. Human computation capabilities have allowed us to understand the properties of the hydrogen atom and, using manual calculations and the Born-Oppenheimer approximation, we can even compute some characteristics of hydrogen molecules. So, with problem-inspired ansätze on current devices, computers haven't significantly expanded our understanding beyond what we already knew through traditional human-driven calculations.

Alternatively, a class of hardware-efficient ansätze has been proposed to accommodate device constraints [71]. The idea is to make good use of a limited number of quantum gates and a particular qubit connection topology. Typically, the gate set used in quantum circuits includes a two-qubit entangling gate and up to three single-qubit gates. The circuit is then built by assembling blocks of single-qubit gates and entangling gates, which are applied in parallel to multiple or all qubits. Each of these blocks is often referred to as a "layer", and the ansatz circuit typically comprises multiple such layers.

The complete VQA program can be broadly conceptualized as a shallow quantum circuit. In this circuit, we iteratively apply rotation gates denoted as $R_{x,y,z}(\theta_\ell) = e^{i\theta_\ell X_\ell}$, with parameters determined by classical values θ_k, followed by the utilization of CNOT gates represented as W_k. A basic illustration of this concept is presented in FIG. 2.
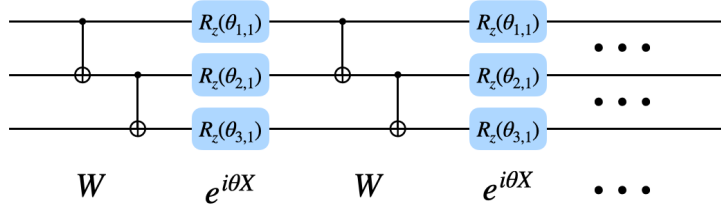
Figure 2. A basic illustration of VQA circuits.

Putting all gates together into one unitary operator, we will have

$$U = \prod_{\ell=1}^{L} W_\ell U_\ell(\theta_\ell) = \prod_{\ell=1}^{L} W_\ell \exp\left(i\theta_\ell X_\ell\right).$$ (2)

The θ_ℓ represent the variational parameters, while U_ℓ are unitary operators with X_ℓ denoting the Hermitian operators. Usually, X_ℓ takes the form of a Pauli operator, and

U_ℓ consists of rotation operators that act on individual qubits. On the other hand, W_ℓ represents the multi-qubit gates responsible for entangling qubits in the circuit. For example, W_ℓ might be a CNOT gate, CZ gate, or similar multi-qubit gates commonly used in superconducting qubits [77]. In line with this methodology, the "Alternating Layered Ansätze" is a specific instance of hardware-efficient ansätze. It comprises layers of single-qubit rotations and blocks of entangling gates that entangle only a local set of qubits. These entangling gate blocks are shifted every alternating layer in the circuit. The selection of these gates, their interconnections, and their order in the circuit significantly impact the portion of the Hilbert space that the ansatz explores and the rate at which it converges for a given problem [17].

Instead of strictly choosing between problem-inspired and hardware-efficient ansätze modalities, some PQC designers have adopted an intermediate approach. For instance, they use an exchange-type gate, which can be natively implemented in transmons, to construct a PQC that respects the symmetry of the variational problem. This approach results in ansatz circuits with particularly low parameter counts, making them suitable for quantum chemistry problems like Hydrogen and LiH molecules.

2.1.3. Measurements   Now, we move into the third stage of VQA, which is the measurement stage. The output of quantum circuits is a quantum state, and in order to optimize it on a classical computer, we must extract classical information from it. This involves estimating the expectation value of the objective function $\langle \hat{\mathcal{O}} \rangle_{U_\theta}$. The most direct approach is to apply a unitary operator to the quantum state, transforming it into the diagonal basis of the observable $\hat{O}$, and then calculate the probability of measuring the state with a corresponding eigenvalue of $\hat{O}$. For experimental details of this approach, see [77, 52]. However, this direct approach is limited by the NISQ devices currently available, as transforming the state into the diagonal basis can be computationally expensive. Below, we introduce alternative methods that is more NISQ-friendly.

(i) *Measurement of Pauli string.*
   Given that most observables can be efficiently expressed as a sum of Pauli strings, a straightforward application is to focus on measuring these Pauli strings. Transforming a state from the computational basis to the diagonal basis of σ_x or σ_y is relatively simple and can be accomplished using Hadamard gates.
   The Pauli string: $\hat{P} = \prod_{k \in K} \boldsymbol{\sigma}(k)$ where k refers to k^{th} qubit, expectation value is: $\langle \hat{P} \rangle = \left\langle \prod_{k \in K} \sigma_z(k) \right\rangle_{\tilde{U}}$, where $\tilde{U}$ refers to the product of rotations according to the specific form of the Pauli string. In practice, it's only feasible to take a finite number of single-shot measurements, denoted as N_s, of the quantum state. This enables the estimation of expectation values within a finite margin of error. For more information on error of the measurements, see [61].

(ii) *Measurement of overlaps.*
   Some VQA would require to measure a unitary's overlap with a quantum state $|\psi\rangle$: $\langle \psi | U | \psi \rangle$. Since this is not an observable, it has both real and imaginary

components. The Hadamard test is a method capable of evaluating such a quantity on a quantum computer using just one additional qubit. This involves applying a control qubit with control over the extra qubit and applying the target unitary operator U to the quantum state.

However, this method does require ample quantum resources to implement the control unitary. An alternative approach, as proposed by the authors of [99], involves decomposing U into a sum of Pauli strings and making measurements with respect to these Pauli strings individually. Another approach is applicable when U can be expressed as a product of unitaries, denoted as U_k, which act locally on a few qubits. This involves finding classical means to express U_k as $U_k = V_k^\dagger D V_k$, where D is a diagonal matrix and V_k represents unitaries that can be applied to $|\psi\rangle$ to determine the overlap. This last approach will lead us naturally to the last part of this section.

(iii) *Classical shadows.*

This technique leverages both the quantum and classical properties of particles. As multiple copies of quantum states' dynamics converge into classical behaviors over time. When dealing with an unknown quantum state represented as ρ, the objective is to predict M expectation values, denoted as $\text{Tr}(\hat{O}_i\rho)$, where $1 \leq i \leq M$ [62].

This method consists of two steps. Initially, a random unitary transformation, denoted as U, is applied to the state ρ, resulting in the transformation $\rho \rightarrow U\rho U^\dagger$. In the second step, all qubits are measured in the computational basis. These two steps are repeated multiple times using different random unitaries U. It's important to note that these random unitaries are selected to be efficiently computable on a classical computer. Through post-processing of the measurement results, it is possible to obtain a "classical shadow," which serves as a classical representation of the quantum state. Research suggests that a classical shadow of size on the order of $\log M$ is sufficient to predict all M expectation values simultaneously. This method draws inspiration from the study of shadow tomography [3], which will be explored further in section 4.1. As mentioned in the introduction, this is an example of the application of other topics in quantum machine learning (shadow tomography) on the narrow sensed quantum machine learning described before.

2.1.4. Optimization Finally, we arrive at the part where we optimize the classical parameters. This optimization process does not differ significantly from other multi-variable optimization problems, so we can employ classical optimization methods [17]. The challenge here, as mentioned earlier, is that in the NISQ era, we are constrained by the limited coherence time of quantum computers. This means that implementing deep analytical gradient descent circuits may not be feasible.

Additionally, to achieve high precision in the mean value of an observable, a substantial number of measurements are required. Due to the high sampling rate, the measurement process becomes the bottleneck in the overall runtime. Therefore, an ideal optimizer for PQC should aim to minimize the number of measurements needed to be effective. At last, it's crucial for the optimizer to be robust in noisy data environments and maintain precision even

with limited measurements taken.

(i) *Stochastic gradient descent*

The technique of gradient descent for locating local minima is a prevalent method in various machine learning applications, particularly in machine learning. While Bayesian learning is intuitively appealing, gradient descent is much more computationally practical.

The essence of this section can be summarized as follows: when presented with an objective function or loss function, denoted as $\mathcal{L}$, our goal is to identify its minima. The specific form of the loss function is not the primary focus here, but for illustration purposes, one of the most common choices is the mean squared error (MSE) loss:

$$\mathcal{L}_{\text{MSE}} \equiv \frac{1}{2}\left(z(x_\delta\,;\boldsymbol{\theta}) - y_\delta\right)^2\ . \tag{3}$$

In machine learning, the term $z(x_\delta\,;\boldsymbol{\theta})$ typically represents the final output, and y_δ corresponds to the supervised learning data. In the context of VQA, consider a simple example where $z(x_\delta\,;\boldsymbol{\theta}) = \langle\psi\rangle_{U(\theta)}$ and $y_\delta = 0$. Therefore, minimizing the loss function is equivalent to minimizing the objective function. All the parameters are encapsulated in the vector $\boldsymbol{\theta}$.

The core principle of gradient descent is to rapidly approach the minimum when we are far from it and take smaller steps when we are close to the minimum. To determine how far we are from the minimum, we rely on the gradient of the loss function. If the gradient has a large magnitude, it signifies that we are far from the minimum, whereas a small gradient indicates proximity to the minimum. Consequently, we update the parameter values based on the gradient's magnitude:

$$\theta_i(t+1) \equiv \theta_i(t) - \eta\ \left.\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\theta_i}\right|_{\theta_i=\theta_i(t)}. \tag{4}$$

This update will necessarily decrease the loss function [112]. So, the iterative application of these updates will gradually bring us closer to a local minimum. However, a significant challenge with this method is the need for a substantial number of measurements. To tackle this problem, stochastic gradient descent (SGD) offers a modification to the standard parameter update rule, where it utilizes $\mathcal{L}_{\mathcal{S}_t}$ which serves as an approximation of the loss function. For instance, you can compute the gradient using a finite number of measurements, as described in the work by [57]. In machine learning, $\mathcal{L}_{\mathcal{S}_t}$ can be thought of as the loss function computed based on a subset of the training data. These subsets within the training data are randomly divided into equal-sized partitions [112].

(ii) *Gradient-free optimization*

There are additional gradient-free optimization approaches that we can explore. Here are a few of these methods, along with their respective references:

(a) **Evolutionary Algorithms**: This technique involves data sampling to estimate the gradient of expected fitness, which is then used to adjust parameters in the direction of steepest improvement [134, 6, 139].

(b) **Reinforcement Learning**: In this method, reinforcement learning is initially applied to optimize parameters, as seen in the case of QAOA parameters. It employs a reinforcement learning framework to discover the most effective "policy" that maximizes the expected total discounted reward [46, 136, 130].

(c) **Sequential Minimal Optimization (SMO)**: In the field of machine learning, SMO has been highly effective for optimizing the parameter landscape of high-dimensional support vector machines. SMO divides the optimization task into smaller components for which analytical solutions can be derived [109, 101, 105].

Also, in [16], the authors present variational quantum algorithms for real and imaginary time evolution that are more efficient than previous proposals, where they use Sequential Minimal Optimization for time evolution.

2.2. Quantum neural tangent kernel

It is important to highlight that the circuits used in quantum machine learning are identical to those employed in variational quantum algorithms (see Fig. 2). Therefore, in this section, we introduce a theory regarding the quantum neural tangent kernel (QNTK), which investigates the dynamics of VQA circuits from first principles [87]. This can also be understood as an extension of the first-principle theory for classical machine learning [112].

Consider an initially prepared state, denoted as $|\Psi_0\rangle$, and apply the unitary operator of the circuit to it (see Eqn. (2)): $|\phi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|\Psi_0\rangle = \prod_{\ell=1}^{L} W_\ell \exp\left(i\theta_\ell X_\ell\right)|\Psi_0\rangle$. Here, ℓ represents the layer of the quantum machine learning system. The output of the quantum neural network is then defined as: $z_{i;\delta} \equiv z_i(\boldsymbol{\theta}, \boldsymbol{x}_\delta) = \langle\Psi_0(\boldsymbol{x}_\delta)| U^\dagger \mathcal{O}_i U |\Psi_0(\boldsymbol{x}_\delta)\rangle$. We use the mean squared loss (see Eqn.(3). We defined the *error factor* or *residual training error* ε as the part in the parenthesis. The value of $y_{i;\alpha}$ is given by the supervised learning database.

However, it can be easily set to 0 if one's task is to solve problems like VQE. The gradient descent equation is shown in Eqn. (4). We then consider the update of the output:

$$dz_{i;\delta} = d\varepsilon_{i;\delta} \equiv \frac{\partial z_{i;\delta}}{\partial \theta_\ell}\left(\mathcal{D}\theta_\ell\right) = \frac{dz_{i;\delta}}{d\theta_\ell}\left(-\eta\frac{\partial \mathcal{L}_\mathcal{A}}{\partial \theta_\ell}\right) = -\eta\sum_{j,\alpha,\ell}\varepsilon_{i';\alpha}\frac{\partial \varepsilon_{i;\delta}}{\partial \theta_\ell}\frac{\partial \varepsilon_{i';\alpha}}{\partial \theta_\ell} = -\eta\sum_{j,\alpha}\varepsilon_{i';\alpha}K_{\delta,\alpha}^{i,i'},$$

(5)

The variational angles θ_ℓ are then updated step by step, labeled by t, according to the derivative of the loss function with respect to θ_ℓ. η is defined as the *learning rate*, which is assumed to be a small (less than 1), positive number. The learning rate determines the size of the step that we take to update our parameters. One can calculate the update of the loss function $\mathcal{L}$ and prove that it decreases indefinitely every time we update our parameters, which is also a fact in classical machine learning [112]. In Eqn. (5) it is easy to show that the update of the output is mathematically in the exact same form as the update of the residual training error $d\varepsilon_{i;\delta}$. One then defines the quantum neural tangent kernel (QNTK):

$$K_{\delta,\alpha}^{i,i'} \equiv \sum_\ell \frac{\partial \varepsilon_{i;\delta}}{\partial \theta_\ell}\frac{\partial \varepsilon_{i';\alpha}}{\partial \theta_\ell}.$$

(6)

If the QNTK is a constant, then Eqn. (5) has a closed form solution. In the continuous limit, the residual error $\varepsilon(t)$ will be a decaying exponential with $-\eta K$ as its exponent. So we eventually will have a guaranteed convergence of the output $z_{i;\alpha}$ to the supervised data $y_{i;\alpha}$ if the QNTK $K_{\delta,\alpha}^{i,i'}$ is a constant.

However, the final assumption is not always true. To wit, with the expression of the output $z_{i,\alpha}$, one can analytically calculate the QNTK and prove that the QNTK is highly fluctuating in the variational angles $\boldsymbol{\theta}$, rather than being a constant [87]. This fluctuation implies that we have obtained representation learning. In the terminology of the kernel method (NTK) of classical machine learning, it means that the feature map itself is changing. This indicates that we are extracting features from all of our training data to update our output, and we cannot obtain a closed-form solution for $\varepsilon(t)$ in this case.

However, we can address this problem if we can determine when the QNTK is approximately a constant. If we can achieve this, we will have a partial success and can implement perturbation theory to achieve convergence.
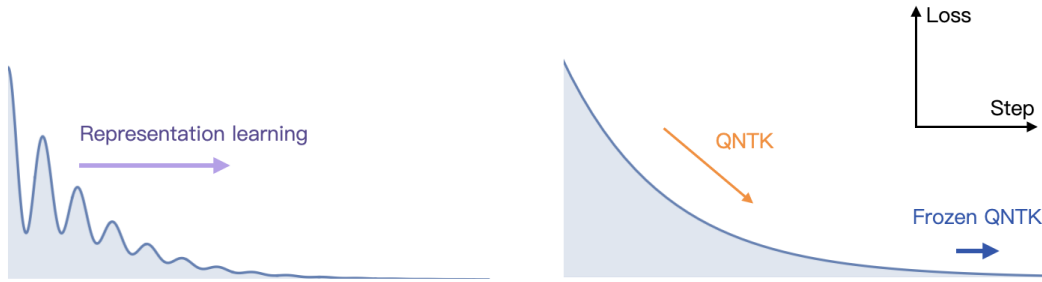

Figure 3. Demonstration of the evolution of the loss function in the continuous limit [87]. The right diagram depicts the closed-form solution when the QNTK remains constant. In contrast, the left diagram illustrates a scenario where the QNTK is nonlinear but nearly constant. In this case, representation learning is achieved, and the gradient descent equation can be addressed perturbatively.

An instance where the QNTK remains nearly constant is referred to as "lazy training". This occurs when the variational angles are nearly unchanged [87, 33]: $\theta_\ell = \theta_\ell^* + \delta\varphi$. If θ_ℓ remains approximately constant, indicating lazy training with minimal movement, the QNTK tends to be nearly constant as well. In such cases, the QNTK expression can be expanded with respect to θ_ℓ for predictable dynamics, akin to classical scenarios [112].

Another scenario involves assuming certain characteristics of the quantum neural network. Firstly, we assume that our variational ansatz U is sufficiently randomly sampled from the unitary group. This enables integration over all samples of U in terms of ensembles, leading to a closed formula for K. This assumption is known as the *k-design assumption*, commonly applied in various fields such as cryptography and black hole physics. The term "k-design" reflects the prediction accuracy up to the kth moment compared to the uniform distribution of the unitary group, indicating a completely random choice of U.

Upon performing the integral, it is deduced that the average QNTK, denoted as $\overline{K}$, is proportional to the number of layers L:

$$\overline{K} = \frac{L\,\mathrm{Tr}(\mathcal{O}_i^2)}{N^2} \propto L \,. \tag{7}$$

Hence, we can infer that the QNTK increases with the depth of our quantum neural network. Additionally, the standard deviation of QNTK, denoted as ΔK, is proportional to $\sqrt{L}$. This implies that the relative deviation becomes very small as we approach the limit where $L \gg 1$:

$$\frac{\Delta K}{\overline{K}} \sim \mathcal{O}(\frac{1}{\sqrt{L}}) \xrightarrow{L \gg 1} \frac{\Delta K}{\overline{K}} \ll 1 \,. \tag{8}$$

This phenomenon is known as the "frozen limit," suggesting that if one employs the k-design assumption, randomly samples the unitary, and ensures that the quantum neural network is sufficiently deep, a nearly frozen QNTK is obtained. This scenario is commonly referred to as a "deep quantum neural network."

As $\overline{K}$ is also proportional to N^2 (refer to Eqn. (7)), for certain operators such as the Pauli matrices, where the trace scales as N, $\overline{K}$ eventually becomes proportional to L/N. This implies that if L is not comparable to the Hilbert space dimension N, a small QNTK is obtained, leading to slow convergence. This undermines the practicality of the model, akin to an alternative statement of the barren plateaus phenomena.

Nonetheless, this model remains useful, as even when not operating at the frozen limit, it provides an approximate means to gauge the speed of gradient descent. For example, to determine which variational ansatz runs the fastest without actually executing them, one can calculate their QNTK and select the one with the largest value. It also enables the estimation of operation time and the number of iterations required for the gradient descent program using Eqn. (7), facilitating hardware design improvements. Moreover, some late-time behaviors of QNTK also been recently captured in [138].

2.3. Quantum landscapes and barren plateaus

Although VQA have been implemented in recent years to address problems in quantum chemistry and similar domains, it remains unclear how well this algorithm will perform in general or under what circumstances VQA might fail. In this section, we introduce barren plateaus as a type of no-go theorem that highlights conditions under which VQA is expected to underperform.

The process of training parameters in the Quantum Machine Learning (QML) model often involves minimizing a loss function and traversing a non-convex landscape in search of its global minimum, as extensively discussed in [27]. Quantum landscape theory, as outlined in [9], seeks to comprehend the properties of QML landscapes and explores strategies for engineering them. Notably, local minima and barren plateaus are focal points of investigation within quantum landscape theory

Akin to classical ML [27], the quantum loss landscape may contain numerous local minima. In a manner similar to classical cases, the overall non-convex optimization can become NP-hard [20]. Various approaches have been proposed to mitigate issues related

to local minima. For instance, variable structure Quantum Neural Networks (QNNs), as introduced in [19, 81], dynamically adjust the model's prior by expanding and contracting during optimization, transforming certain local minima into saddle points. Additionally, there is evidence of the overparametrization phenomenon in QML [73, 79]. In this scenario, a computational phase transition occurs as the optimization progresses, leading to the disappearance of spurious local minima when the number of parameters surpasses a critical value.

Not only are local minima a concern in QML, but there is also the intriguing phenomenon of a *barren plateau* in quantum landscapes [27, 25, 8, 60]. A barren plateau is characterized by the loss landscape becoming, on average, exponentially flat concerning the problem size. In this scenario, the valley housing the global minimum exponentially diminishes with the problem size, forming what is known as a *narrow gorge* [9]. Consequently, exponential resources, such as an increased number of shots, are needed to traverse through the landscape. This impact on resource requirements complicates the QML algorithm's complexity and can potentially undermine quantum speedup, as quantum algorithms typically aim to avoid the exponential complexity associated with classical algorithms.

The barren plateau phenomenon was first studied in deep hardware-efficient QNNs [98], where they arise due to the high expressivity of the model [8, 27]. By making no assumptions about the underlying data, deep hardware-efficient architectures aim to solve a problem by being able to prepare a wide range of unitary evolutions. In other words, the prior over the hypothesis space is relatively uninformed. Barren plateaus in this unsharp prior are caused by ignorance or the lack of sufficient inductive bias, and therefore a means to avoid them is to input knowledge into the construction of the QNN - making the design of QNNs with good inductive biases for the problem at hand a key solution. Various strategies have been developed to address these barren plateaus, such as clever initialization [126], pretraining, and parameter correlation [60]. These are also examples of adding a sharper prior to one's search over the over-expressive parameterizations of hardware-efficient QNNs. The authors of [28] studied the barren plateau problem for parameterized quantum circuits of the quantum tensor network architecture and show, for example, that classical optimization of these circuits can be exponentially more efficient than using a quantum computer.

Other mechanisms have been linked to barren plateaus [27]. Simply defining a loss function based on a global observable (i.e., observables measuring all qubits) leads to barren plateaus even for shallow circuits with sharp priors [98], while local observables (those comparing quantum states at the single-qubit level) avoid this issue [98, 125]. The latter is not due to bad inductive biases but rather to the fact that comparing objects in exponentially large Hilbert spaces requires an exponential precision, as their overlap is usually exponentially small.

While entanglement is one of the most important quantum resources for information processing tasks in quantum computers, it can also be detrimental for QML models [27]. QNNs (or embedding schemes) that generate too much entanglement also lead to barren plateaus [116, 95, 106]. Here, the issue arises when one entangles the visible qubits of the QNN (those that one measures at the QNN's output) with a large number of qubits in

the hidden layers. Due to entanglement, the information of the state is stored in non-local correlations across all qubits, and hence the reduced state of the visible qubits concentrates around the maximally mixed state. This type of barren plateau can be solved by taming the entanglement generated across the QNN.

2.3.1. Influence of background noise The existence of hardware noise during quantum computations is a distinctive feature of NISQ computing [27]. Despite this reality, many QML studies overlook noise in analytical calculations and numerical simulations, while still asserting near-term compatibility of the methods. It is imperative to consider the impact of hardware noise in QML analyses, especially for those aiming to achieve quantum advantage with currently available hardware [27].

Noise introduces errors in the information as it propagates through a quantum circuit, particularly affecting deeper circuits with longer run-times [27]. This impact extends to all aspects of quantum models, including the dataset preparation scheme and circuits used for computing quantum kernels. In the case of Quantum Neural Networks (QNNs), noise can impede their trainability [128, 127], leading to noise-induced barren plateaus where relevant features are exponentially suppressed with increasing circuit depth. The consequence is a deformation of the model's inductive bias and an effective reduction in the dimension of the quantum feature space. Despite the significant influence of quantum noise, its effects remain largely unexplored, especially regarding its impact on the classical simulability of QML models [39, 53, 27].

Addressing issues induced by noise may necessitate [27]: (1) reducing hardware error rates, (2) implementing partial quantum error correction [24], or (3) utilizing QNNs that are relatively shallow (i.e., with depth growing sub-linearly in the problem size) [128], such as Quantum Convolutional Neural Networks (QCNNs). While error mitigation techniques [124, 38, 42] can enhance the performance of QML models in the presence of noise, they may not completely resolve noise-induced trainability issues [127]. Another approach involves engineering QML models with noise-resilient properties [80, 35, 115], such as ensuring that the position of minima remains unchanged despite noise.

However, another study suggests that noise in VQAs might be beneficial for their performance. Drawing inspiration from classical machine learning, where noise is intentionally introduced to enhance the performance of the gradient descent algorithm, this phenomenon is akin to stochastic gradient descent. It has been observed that gradient descent tends to perform better under the influence of noise, as the noise helps in avoiding saddle points that could hinder the algorithm's progress [69, 70]. In the absence of noise, the algorithm may converge directly to the saddle point. However, when noise is introduced, perturbations around the points prevent convergence towards these unstable fixed points.

In the study [88], this discussion was extended to quantum noise. Through experiments, the researchers found that the noisy case outperformed the noiseless counterparts, providing a toy model illustration to support their argument. This suggests that, while implementing quantum error correction is crucial for making quantum devices fault-tolerant, in certain QML applications, eliminating all noise may not be necessary. As long as the noise remains below

a certain threshold, it may prove not detrimental but rather helpful for QML programs.

2.3.2.   Laziness Theoretically one can understand barren-plateau phenomenon as the following. A typical gradient descent algorithm will look like

$$\theta_\ell(t+1) - \theta_\ell(t) = -\eta \frac{\partial \mathcal{L}}{\partial \theta_\ell} \equiv \delta\theta_\ell \,. \tag{9}$$

The observation [98] is that if our variational ansatz is highly random, according to the k-design integral formula [111, 85, 84, 36], the derivative of the loss function is generally suppressed by the dimension of the Hilbert space N. This might lead to a situation where the variation of the loss function during gradient descent is very small, denoted as $\delta\mathcal{L} \equiv \mathcal{L}(t+1) - \mathcal{L}(t) \ll 1$ for the step t. An alternative theoretical term used to describe the quantum barren plateau, based on the large suppression in Eqn. (9), is laziness [86]. In the quantum context, the suppression comes from the dimension of the Hilbert space, while in the classical case, the suppression is associated with the width of classical neural networks.

To be more precise, laziness refers to small $\delta\theta_\mu$, and the barren plateau refers to small $\delta\mathcal{L}$. The authors of [86] demonstrate that laziness may not necessarily imply the quantum barren plateau. This insight is derived from the perspectives of overparametrization theory and representation learning theory through the QNTK [87]. In the context of quantum neural networks, *overparametrization* refers to the condition $\overline{K} \approx \mathcal{O}(1)$ in Eqn.(7).

The paper [86] emphasizes that in variational circuits with a sufficiently large number of trainable angles, gradient descent dynamics can still be efficiently executed, even in the presence of the exponential suppression of variational angle updates (laziness). On the other hand, the only limitation is the precision, where noises might affect the performance significantly. The authors also highlight that laziness is not exclusive to quantum machine learning but is observed in overparametrized classical neural networks with large widths as well. The efficiency of large-width neural networks is substantiated by the neural tangent kernel theory, and similar principles apply to their quantum counterparts.

3. Fault tolerant quantum computation (FTQC) algorithms

In the Fault-Tolerant Quantum Computation (FTQC) era, we envision a future where we have fully implemented quantum error correction. In this era, one of the most prominent and impactful applications is Shor's algorithm [117]. This algorithm has the remarkable ability to efficiently factorize large numbers, a capability that could potentially break the majority of the public-key cryptography systems currently used on the internet. Additionally, quantum computers, as originally envisioned by Richard Feynman [45], have the potential to significantly accelerate the simulation of quantum physics and chemistry.

Despite these two remarkable applications, quantum computers have historically faced a problem of having limited use cases. While these two applications are groundbreaking, there is no concrete evidence that quantum computers will revolutionize our lives in the same way that classical computers have done [1].

Recently, a series of newly proposed quantum algorithms have emerged, offering exciting possibilities for quantum computation. Among them, the Harrow-Hassidim-Lloyd (HHL) algorithm, introduced by its authors in 2008 [58], has gained significant attention. These novel algorithms not only hold the potential for significant speedups compared to classical counterparts but also show promise in addressing practical problems. These applications span various domains, including machine learning, clustering, classification, and the analysis of extensive datasets [1]. Although, significant challenges are related to many algorithms designed for quantum machine learning, including dequantization and fast interfaces between classical and quantum processors.

In this section, we will concentrate on a few selected quantum algorithms and delve into their applications.

3.1. Quantum phase estimation and quantum principle component analysis

In the realm of quantum computing, quantum phase estimation is a pivotal quantum algorithm primarily designed to estimate the phase associated with an eigenvalue of a provided unitary operator [102]. The necessity to exponentiate the density matrix within quantum principle component analysis (PCA) aligns with the requirement for phase estimation.

In a formal context, suppose a unitary operator U and its corresponding eigenvector $|\psi\rangle$. The relationship is represented as: $U |\psi\rangle = e^{i2\pi\theta} |\psi\rangle$. Here, without loss of generality, we assume $0 \leq \theta < 1$. and the objective of Quantum Phase Estimation (QPE) is to accurately determine the value of θ while minimizing the number of operations required. The concept involves generating a binary approximation of θ through the application of quantum Fourier transformation. The preparatory steps are illustrated in FIG. 4.
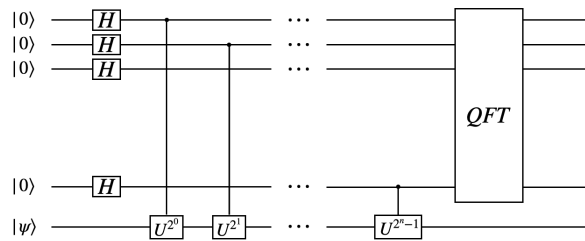

Figure 4. Circuits for quantum phase estimation.

Beginning with $|\psi\rangle$ and n qubits initialized in the $|0\rangle$ state: $|0\rangle^{\otimes n} \otimes |\psi\rangle$. First one applies Hadamard gates to all $|0\rangle$ qubits: $H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{2^{n/2}} (|0\rangle + |1\rangle)^{\otimes n} = \frac{1}{2^{n/2}} \sum_{k=0}^{2^{n}-1} |k\rangle$. One then decomposes k : $k = \sum_{j=0}^{n-1} k_j 2^j$. The next step involves applying the unitary operator U to $|\psi\rangle$ while maintaining control over the other qubits. Practically, one applies U^{2^j} on $|\psi\rangle$, only if $k_j = 1$, for $0 \leq j \leq n-1$. Finally, this sequence of operations results in the state:

$$\frac{1}{2^{n/2}} \sum_{k=0}^{2^{n}-1} |k\rangle |\psi\rangle \rightarrow \frac{1}{2^{n/2}} \sum_{k=0}^{2^{n}-1} |k\rangle \, U^{k} |\psi\rangle = |k\rangle \, U^{k_0 2^0} U^{k_1 2^1} \dots U^{k_{n-1} 2^{n-1}} |\psi\rangle = \frac{1}{2^{n/2}} \sum_{k=0}^{2^{n}-1} e^{i2\pi k\theta} |k\rangle |\psi\rangle \ .$$
(10)

The eigenstate $|\psi\rangle$ is not needed at this stage. We proceed with the quantum Fourier transformation on the qubits $|k\rangle$:

$$\frac{1}{2^{n/2}} \sum_{k=0}^{2^{n}-1} e^{i2\pi k\theta} |k\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{k=0}^{2^n-1} \exp\left\{ \frac{-i2\pi k}{2^n} \left(x - 2^n \theta\right) \right\} |x\rangle \ .$$
(11)

Now, one estimates θ by rounding $2^n \theta$ to the nearest integer. If $2^n \theta$ can be expressed as an integer, then one will obtain a precise result after performing measurements. On the other hand, If θ cannot be accurately expressed up to n^{th} order binary, QPE will have an error. So the larger n is, the smaller the error is. Because we have more space to approximate θ . The good news is the error of QPE estimation is always bounded [102]. By approximating θ, it's possible to find the eigenvalue of the unitary operator. Consequently, since unitary operators can always be expressed as an exponential of a Hermitian operator, one can also determine the eigenvalue of the Hermitian matrix. This is the principle behind how quantum PCA calculates both the eigenvectors and eigenvalues of the density matrix.

One of the most fascinating applications in quantum machine learning is Quantum Principal Component Analysis. This method, akin to the popular Principal Component Analysis (PCA) used by companies like Netflix, enables the analysis of users' preferences to offer tailored film recommendations. In a mathematical sense, the data provided by users is represented as vectors: v_j , in a d dimensional vector space. PCA's objective is to identify the ideal axes for grouping this data into clusters. Another application could involve generating these vectors based on the fluctuations in stock prices for all stocks in the market, analyzing changes from discrete times t_i to t_{i+1} [18].
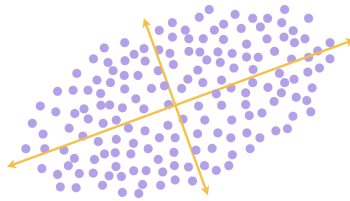

Figure 5. A simple illustration of principle component analysis (PCA).

Classically, this program is done as the following [18]. First, one calculate the covariance matrix: $\boldsymbol{C} = \sum_j \boldsymbol{v}_j \boldsymbol{v}_j^T$, where T represents the transverse matrix. In principle, one can understand the covariance matrix as a manifestation of the correlation between different component of the data. Now, one can diagonalize the covariance matrix and find its eigenvectors c_k and corresponding eigenvalue e_k : $\boldsymbol{C} = \sum_k e_k \boldsymbol{c}_k \boldsymbol{c}_k^\dagger$.

If only a few of the eigenvalues c_k are large, and the remainder are small or zero, then the eigenvectors corresponding to those eigenvalues are called the principal components of C . In more mathematical terms, if the diagonalized covariance matrix is sparse, then the non-zero eigenvalues are called the principle components. Each principal component represents an underlying common trend or form of correlation in the data, and decomposing a data vectors v in terms of principal components,

If only a few of the eigenvalues c_k are significantly large, and the rest are considerably small or zero, then the eigenvectors linked to these eigenvalues are termed the principal components of C In more technical terms, if the diagonalized covariance matrix is sparse, the non-zero eigenvalues represent the principal components. Each principal component characterizes an underlying common trend or form of correlation within the data. Decomposing a data vector v into these principal components through the equation: $v = \sum_k \tilde{v}_k c_k$, enables compression of data representation and the anticipation of future behavior. Classical algorithms for PCA possess computational complexity and query complexity scaling as $\mathcal{O}(d^2)$.

In the quantum version of this program, the initial step involves mapping these vectors v_j into quantum states: $|v_j\rangle$. This mapping is facilitated by quantum random access memory (QRAM), a concept that will be further detailed in section 3.4. A d dimensional vector can be encoded in $\log d$ qubits. Assuming every vector v_j is equally probable or is randomly selected from a dataset containing N vectors the resulting density matrix is: $\rho = \frac{1}{N} \sum_j |v_j\rangle\langle v_j|$. This density matrix essentially serves as the quantum equivalent of the covariance matrix C in the classical scenario. The subsequent step for quantum principal component analysis involves implementing two methods: density matrix exponentiation and quantum phase estimation. The former maps the density matrix ρ into a unitary operator: $U_{\mathrm{DME}} = e^{-i\theta\rho}$, which enables us to utilize QPE to evaluate the eigenvalues and eigenstates.

The latter method aims to find the eigenvectors and eigenvalues of the density matrix written in U_{DME}. By repeatedly sampling data and using the two aforementioned techniques, one can take any quantum version of a data vector $|v_j\rangle$ and decompose it into its principal components: $|v_j\rangle \rightarrow \sum_k v_k |c_k\rangle$. It's important to note that quantum principle component analysis scales as $\mathcal{O}(\log^2 d)$ in both computational complexity and query complexity, which represents a significant speedup compared to the classical version. In the study conducted in [67], the researchers present experimental evidence showcasing quantum advantage in PCA. However, a crucial prerequisite for this advantage is the availability of quantum data sourced from physical experiments, capable of generating density matrices. This contrasts with classical devices relying on QRAM for data processing. And the quantum advantage is measured in the number of queries to get the density matrix.

3.2. Classical algorithm framework for dequantizating QML models

Despite the development of exciting quantum algorithms in QML, there are still open problems whether known QML algorithms provide new exponential speedups over classical algorithms for practically relevant instances of machine learning problems [34, 40]. This

uncertainty is rooted in the challenge of identifying the source of this speedup. Some arguments suggest that QML's exponential acceleration is solely attributed to its state preparation assumptions. Notably, the concept of "dequantization" has emerged, involving the development of classical algorithms that mimic the sampling assumptions of QML programs. Authors in this domain have created classical algorithms, often referred to as the dequantization of QML programs, which closely resemble their quantum counterparts but exhibit only polynomial differences in complexity. One notable example of dequantization involves the quantum recommendation system.

The work by Kerenidis and Prakash on the quantum recommendation system [72] was noteworthy for addressing the caveats associated with the HHL algorithm, as pointed out by Scott Aaronson [1]. They presented a comprehensive quantum algorithm directly comparable to classical algorithms. Initially, Kerenidis and Prakash's quantum algorithm demonstrated exponential speedup compared to the best-known classical algorithms, but its provable exponential nature was uncertain. In 2019, Tang [121] introduced a classical algorithm that dequantized the quantum recommendation systems algorithm with only polynomially slower runtime, providing clarity on the provable nature of the speedup. These works highlighted the insight that the data structure fulfilling state preparation assumptions can also fulfill ℓ^2-norm sampling assumptions (as defined in section 2.2 of [121]). More precisely, this is a $\log$-dimension time classical randomized linear algebra algorithms in the sample and query input model, which is an analogue of quantum state preparation assumptions coming from, for example, in most cases, QRAM). Consequently, a classical algorithm aiming to "match" the quantum algorithm can leverage these assumptions.

In 2020, the authors of [32] introduced an algorithmic framework for quantum-inspired classical algorithms focusing on close-to-low-rank matrices. This work generalized the series of results initiated by Tang's quantum-inspired algorithm for recommendation systems [121]. The motivation for these classical algorithms stemmed from quantum linear algebra algorithms and the quantum singular value transformation (SVT) framework [47]. The proposed classical algorithms for SVT exhibited runtime independent of input dimension under suitable quantum-inspired sampling assumptions. The results provided compelling evidence that, in the corresponding QRAM data structure input model, quantum SVT does not deliver exponential quantum speedups.

In 2021, the authors of [122] introduced a new input model termed *SQ access*, representing a form of l2-norm sampling assumption. They developed a model to dequantize two influential QML algorithms, namely quantum principal component analysis [93] and quantum supervised clustering [92]. In essence, they provided classical algorithms that, with classical SQ access assumptions replacing quantum state preparation assumptions, replicated the bounds and runtime of the corresponding quantum algorithms, albeit with a polynomial slowdown.

On the other hand, in [67] the authors point out that the assumption in [122] does not work in the quantum physics environment setup, and it is possible that there are still quantum advantages with exponential separation. In [67], their quantum advantages are measured in terms of the number of queries to access the quantum state requiring principle

component analysis. In [122], It presupposes the capability to retrieve any element of the exponentially large density matrix with exponentially precise accuracy, within a time frame that is polynomial. Achieving the capability already requires exponential numbers of queries to access the density matrix. See further detailed discussions in [37]. In summary, the pursuit of identifying classical analogs for quantum machine learning applications presents a fascinating avenue for exploration. Investigating dequantization enables the identification of the constraints inherent in both quantum and classical machine learning, and aids in delineating the distinctions between quantum and classical algorithms.

3.3. Harrow-Hassidim-Lloyd algorithm

In this section, we will delve into the intricacies of the HHL algorithm, as detailed in [58]. The fundamental setup remains consistent:

$$A\,|x\rangle = |b\rangle \ , \tag{12}$$

where A is an $N \times N$ Hermitian matrix, and $|x\rangle$ and $|b\rangle$ are normalized vectors. We assume that A is Hermitian which does not compromise the generality of the algorithm, as the space can always be expanded to make it true. For $|u_i\rangle$ the eigen vectors of A, we can decompose A and $|b\rangle$, such that

$$|x\rangle = A^{-1}\,|b\rangle = \sum_{i,j=0}^{N-1} \lambda_i^{-1}|u_i\rangle\langle u_i|b_j\,|u_j\rangle = \sum_{i,j=0}^{N-1} \lambda_i^{-1}b_j\delta_{ij}\,|u_i\rangle = \sum_{i=0}^{N-1} \lambda_i^{-1}b_i\,|u_i\rangle \ . \tag{13}$$

Now, let's delve into the HHL algorithm circuit, which step by step accomplishes what we've discussed in Eqn. (13). We'll track the evolution of qubit states as follows.
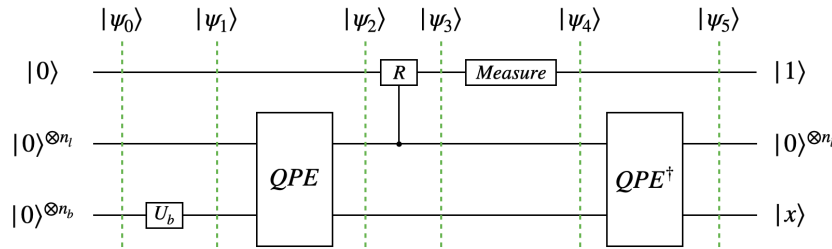

Figure 6. Circuits for HHL algorithm [11].

With the entire process divided into 5 stages [58, 11], we begin with the following initialization: $|\psi_0\rangle = |0\rangle^{\otimes n_b}\,|0\rangle^{\otimes n_l}\,|0\rangle$.

The goal is to encode the information of $|b\rangle$ in the first register: $|0\rangle^{\otimes n_b}$ and encode λ_j in the second register: $|0\rangle^{\otimes n_l}$. These can be done with a unitary operator U_b and quantum phase estimation which determines the values of λ_j and $|u_j\rangle$: $|\psi_2\rangle = \sum_{j=0}^{N-1} b_j\,|u_j\rangle\,|\lambda_j\rangle\,|0\rangle$.

The key step is to rotate the third register qubit with an angle with a control on the clock register and do a post-selection on the third register of obtaining $|1\rangle$:

$$|\psi_3\rangle = \sum_{j=0}^{N-1} b_j |u_j\rangle |\lambda_j\rangle \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}} |0\rangle + \frac{C}{\lambda_j} |1\rangle \right) \xrightarrow{\text{post-select}} |\psi_4\rangle = \sum_{j=0}^{N-1} b_j |u_j\rangle |\lambda_j\rangle \frac{C}{\lambda_j} |1\rangle \ ,$$

(14)

where constant C is the normalization factor. If we obtain $|0\rangle$, hen we will have to repeat the above procedure until we obtain the desired result. Now we have our desired λ_j^{-1}, so we can apply the inverse of the quantum phase estimation procedure and return to our initialized second register and we have the following identification:

$$|\psi_5\rangle = C \sum_{j=0}^{N-1} b_j \lambda_j^{-1} |u_j\rangle |0\rangle^{\otimes n_l} |1\rangle = C |x\rangle |0\rangle^{\otimes n_l} |1\rangle \ . \tag{15}$$

The state output of the first register is our desired solution state or vector. The whole post-selection process is for us to be sure that the state in the first register is now proportional to the solution vector, and from there, we can complete our algorithm. For a more detailed discussion of the HHL algorithm, please refer to the original paper [58]. The HHL algorithm takes $\mathcal{O}((\log N)^2)$ quantum steps to output $|x\rangle$, compared with $\mathcal{O}(N \log N)$ steps required to find the vector x using the best-known method on a classical computer. HHL does achieve a significant speedup, and so does the quantum SVM compared to its classical counterpart.

Before we become overly enthusiastic about this algorithm and its broad applications, it's important to note the following caveats, which can be crucial in practice [1].

(i) HHL can handle various types of information. However, when dealing with classical information, it's essential to efficiently load the vector $b = (b_1, \ldots, b_n)$ into the quantum computer's memory. This process involves transforming:

$$\boldsymbol{b} \to |b\rangle = \sum_{i=0}^{N-1} b_i |u_i\rangle \ , \tag{16}$$

must occur rapidly. If the preparation of the state $|b\rangle$ requires exponential steps, the significant speedup gained by HHL in subsequent steps might be negated. Theoretically, this might be achieved through quantum random access memory (QRAM), which will be discussed in section 3.4. Nevertheless, it is crucial that b is reasonably uniform. For evident reasons, this challenge is referred to as the input problem [1, 18].

(ii) The quantum computer must also have the capability to apply unitary transformations of the form e^{-iAt} for various values of t. If the matrix A is *sparse*—meaning it contains at most s nonzero entries per row, where $s \ll n$—and if there exists a quantum random access memory (QRAM) that efficiently stores, for each i, the locations and values of row i's nonzero entries, then it is known that one can apply e^{-iAt} in a time that grows nearly linearly with s.

(iii) The matrix A not only needs to be invertible but also robustly invertible, or "well-conditioned." Otherwise, the significant speedup provided by the algorithm may be lost.

(iv) Finally, there's the output problem, which is the converse of the input problem [18]. The challenge of encoding classical information into quantum states also extends to reading the output quantum state $|x\rangle$ into classical information x when HHL is finished. Learning the value of any specific entry x_i by measurement will, in general, require repeating the algorithm roughly N times, negating the significant speedup.

To summarize, HHL is an algorithm designed to solve a system of linear equations approximately in logarithmic time. It serves as a template for other quantum algorithms, as long as one can address and mitigate the various caveats associated with its practical implementation [1]. If these challenges are properly handled, HHL can find applications in various real-world scenarios.

3.3.1. Large scale machine learning models in fault tolerant era There are trials about applications of the HHL algorithm and their implementation, particularly at a large scale, in Quantum Machine Learning (QML) programs. One observation is that, the mathematical description of the gradient descent program involves solving a set of difference equations, which, in the continuous limit, transforms into ordinary differential equations (ODEs).

The paper [90] introduces a technique known as **Carleman linearization**, demonstrating that non-linear ODEs can be linearized into an infinite set of linear ODEs that can be truncated to a chosen level of approximation. This allows the approximate solution of challenging non-linear ODEs by solving the easier linearized versions. The authors establish that if the original ODEs are non-linear and *dissipative*, then the error resulting from the truncation in Carleman linearization can be controlled. This implies that a non-linear, dissipative system can be efficiently linearized.

In the realm of quantum computing, the HHL algorithm is known to efficiently solve linear ODEs, as expressed in Eqn. (12), providing a significant speedup compared to classical algorithms. Consequently, the combination of Carleman linearization and the HHL algorithm holds the promise of yielding a significantly efficient algorithm for solving non-linear ODEs in comparison to classical algorithms.

Dissipativeness plays a crucial role in various machine learning tasks. Many machine learning processes, including gradient descent, exhibit dissipative behavior. Treating gradient descent as dynamics reveals that it operates as an open first-order system outside the scope of Lagrangian mechanics [89]. Biological processes in the brain also exemplify the significance of dissipation, where forgetting becomes essential for making room to remember new information [89].

The findings of [89] suggest the potential for an efficient algorithm to solve stochastic gradient descent. Generalizing the techniques for differential equations to *ordinary difference equations*, it is observed that in the learning process, the Hessian eigenvalues characterize the degree of dissipation. Towards the end of training, there are roughly equal numbers of positive and negative eigenvalues, signifying comparable dissipative and non-dissipative modes. However, at the beginning of training, the number of dissipative modes significantly exceeds the number of non-dissipative modes, creating highly asymmetric Hessian eigenvalues. This

phase is recognized as highly dissipative, consistent with the intuition that the system learns rapidly in the initial stages. The ODE solver described earlier is applicable during this phase, enabling the solution of machine learning tasks. Towards the end of training, when positive and negative eigenvalues become comparable, the algorithm may not be suitable for solving ODEs in general. An exception is when approaching a local minimum, where the Hessian matrix is positive definite, indicating a dissipative system suitable for the ODE solver. In summary, this quantum algorithm might help machine learning processes significantly, and if the Hessian becomes symmetric towards the end, classical algorithms can be employed for further computation.

Overcoming the input and output challenges in quantum machine learning is a complex task. While Quantum Random Access Memory (QRAM) is a promising solution for efficient data input, it is still in the developmental stage. Addressing the input challenge in quantum machine learning can be tackled using a classical algorithm known as *"pruning"*. Typically, towards the end of training in many machine learning algorithms, the matrix of training parameters becomes highly sparse, meaning that a significant number of training parameters become zero. This sparsity presents an opportunity to effectively manage the input problem in quantum machine learning.

Dealing with the output problem involves leveraging the sparsity that often occurs in parameter matrices at the end of training, making it potentially easier to download sparse quantum states to classical computers. Additionally, classical shadow techniques, with their polynomial scaling, provide a method for extracting information from dense quantum states, if we know the sparsity of the matrix. These areas are actively researched, and advancements in quantum computing technologies and algorithms may offer new solutions in the future.

Indeed, the vision of leveraging quantum algorithms, particularly with large-scale implementations of algorithms like the HHL, holds the promise of addressing the increasing costs associated with classical algorithms. The potential gains in efficiency and computational speed offered by quantum computers could revolutionize machine learning and other computational tasks. However, it's important to emphasize that realizing these advancements requires the implementation of full quantum error correction to ensure fault tolerance. Overcoming the challenges associated with error correction remains a critical aspect of bringing quantum computing applications to fruition. Moreover, some further explorations about machine learning foundations should be given along this line, like improving the algorithms from solving gradient descent to solving the matrix multiplication problems in large-scale classical neural networks for back propagation, where quantum linear algebra techniques or parallel computing from analog photonic devices might be helpful for those tasks [14].

3.4. Quantum random access memory

Now, let's explore the challenges associated with the "input problem" and the "output problem". Classical data needs to be inputted into a quantum computer before undergoing processing [18]. Referred to as the 'input problem', this step is generally performed with

minimal overhead but can pose a significant bottleneck for certain algorithms. Similarly, the 'output problem' arises when retrieving data after it has been processed on a quantum device. Much like the input problem, the output problem can lead to a substantial operational slowdown.

In particular, when considering the application of algorithms like HHL, least squares fitting, quantum support vector machines, and related approaches, substantial amounts of classical data might be loaded into a quantum system. This loading process can demand significant time, as acknowledged in the literature [1, 18]. While the use of a quantum random access memory (QRAM) [54, 56] might, in principle, address this issue, the associated costs might be prohibitive for large-scale data problems [10].

QRAM can be conceptualized as an architecture designed for the implementation of quantum *oracles* [54]. Consider a computational problem where the input is represented by a classical data vector x. An oracle, often described as a black box, is a tool that can be interrogated to disclose information about x. Although the oracle reveals details about x upon request, the specific method it employs to obtain this information is not explicitly specified.
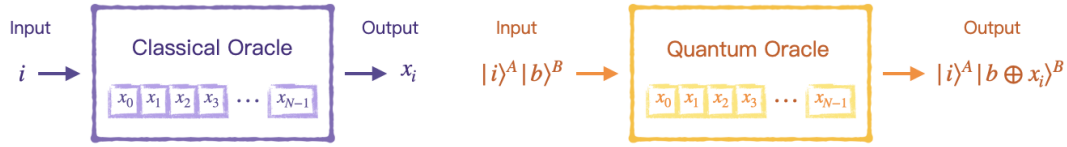

Figure 7. Classical and quantum data-lookup oracles [54].

Oracles can exist in either classical or quantum forms, as illustrated in FIG. 7. A basic example of a classical oracle is the data-lookup oracle. This type of oracle is activated by providing it with an index i as input, and it subsequently outputs the corresponding vector element x_i. A natural extension of this classical data-lookup oracle is the quantum data-lookup oracle. In the quantum scenario, both the inputs and outputs of the oracle query are represented as quantum states. The query itself is implemented by a unitary operation, denoted as $\mathcal{O}_x^{(\mathrm{DL})}$, which performs the mapping [54]:

$$\mathcal{O}_x^{\mathrm{DL})} |i\rangle^A |b\rangle^B = |i\rangle^A |b \oplus x_i\rangle^B , \qquad (17)$$

where the notation b represents any computational basis state and $\oplus$ denotes addition modulo 2. The superscripts A and B refer to two quantum registers, the state of register A indicates which element to look up, and the query encodes this element into the state of register B. It's important to note that, while the inputs and outputs of the query are quantum, the data being queried is classical. This characteristic makes quantum oracles serve as a bridge between classical data and quantum algorithms [54].

However, merely loading classical data into quantum memory is not sufficient; the process needs to be fast [1]. QRAM is designed to address this requirement. In QRAM, the quantum state summarizing the vector uses $\log d$ qubits, and the QRAM operation involves $\mathcal{O}(d)$ operations distributed over $\mathcal{O}(\log d)$ steps, which can be performed in parallel [18].

For QRAM, a quantum superposition of different addresses $|\psi_{\text{in}}\rangle$ serves as input, and the QRAM produces an entangled state $|\psi_{\text{out}}\rangle$ where each address is correlated with the corresponding memory element [54]:

$$|\psi_{\text{in}}\rangle = \sum_{i=0}^{N-1} \alpha_i \, |i\rangle^A \, |0\rangle^B \xrightarrow{\text{QRAM}} |\psi_{\text{out}}\rangle = \sum_{i=0}^{N-1} \alpha_i \, |i\rangle^A \, |x_i\rangle^B \ , \tag{18}$$

where N is the size of the data vector x, and the superscripts A and B represent "address" and "bus," respectively.

There are two popular designs for QRAM: fanout QRAM and bucket-brigade QRAM. Their difference primarily lies in how they utilize the address qubit. All QRAM designs consist of quantum routers, which serve to route the input qubit to the right if the router qubit is in state $|1\rangle$ and to the left if the router qubit is in state $|0\rangle$. In fanout QRAM, all router qubits are first flipped according to the address qubit. For example, if we want to query the information stored at site $i = 5$, the address qubits will be in the computational basis: $|5\rangle = |1\rangle \, |0\rangle \, |1\rangle$. Then, we need a QRAM with a depth of 3 and flip all the router qubits with a CNOT gate controlled by the address qubits, as illustrated in the figure below.
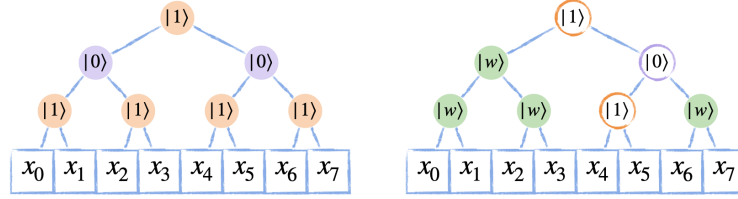

Figure 8. The left figure is a picturesque illustration of fanout QRAM [54] to retrieve information stored at $i = 5$. On the other hand, the right figure is a picturesque illustration of bucket-brigade QRAM [54] to retrieve information stored at $i = 5$.

Then we can route the bus qubit, and since we stored classical information at x_i, it is possible for us to copy it to the bus qubit and then route it all the way out before we initialize the QRAM for future use.

The bucket-brigade QRAM is a bit different in the first stage. We don't flip all the router qubits in the layer corresponding to the address qubits. Instead, we route the address qubits into the QRAM one by one, with the router qubits initialized at some state $|w\rangle$. Since we route in the address qubits in order, we naturally carve out the path that will lead to the information stored in x_5.

There are debates about whether QRAM is actually practical since it is time-saving but also hardware-consuming. Recently, there are several different QRAM designs that are hardware-efficient. One example is developed by the authors of [55]. They made use of circuit quantum acoustodynamic (cQAD) systems [103, 51] and constructed quantum gates by applying external drives with different frequencies.

Furthermore, due to its logarithmic depth nature, when QRAM is stored with a large number of data, the speed for querying it will ultimately be bounded by the speed of light. The problem is how many qubits QRAM can store before we hit the boundary set by locality.

This problem is addressed in [129], where the authors considered locality and found that some 2D or 3D QRAM designs can handle up to 10^{20} qubits before hitting the constraint line. This might be already a sufficient number of qubits to perform machine learning training programs. As an result, QRAM might be still reasonable to build in the future.

Although it looks like that QRAM might be implementable, it is also important that there is no definitive evidence or proof in the hardware especially in the large scale up to date. So whether we can achieve exponential speed ups that some quantum machine learning programs promises might be still unknown and controversial. On the other hand, for some of the QML programs, one can go around QRAM by implementing a quantum subroutine which prepares the quantum states that encodes classical information [18].

4. Statistical learning theory

In this section, our aim is to approach a quantum counterpart to statistical learning theory, initiating a systematic framework for delving into the fundamentals of quantum machine learning. A crucial aspect intertwined with statistical learning theory is shadow tomography, addressing a challenge rooted in the distinctive nature of quantum measurements—destructiveness. When concluding a Quantum Machine Learning (QML) program and obtaining a quantum state, extracting classical information becomes a non-trivial endeavor. The challenge arises from the inevitable collapse of the state during measurement, limiting access to the original state. Working with a single copy of an unknown quantum state, denoted as ρ, poses an inherent difficulty, as clever strategies cannot approximate a classical description of ρ through direct measurement. The overarching task of reconstructing a description of a D-dimensional quantum mixed state ρ, given multiple copies of ρ, falls under the domain of *quantum state tomography* [2]. On the other hand, works such as [21, 22] studied statistical complexity in the context of quantum machine learning, which also closely relates to this topic.

One significant development inspired by shadow tomography is known as *classical shadow*. As discussed earlier in section 2.1.3, this approach holds promise for addressing the output or downloading problem. Classical shadow infers the output state ρ by performing random measurements, averaging the obtained classical data, and then inverting the results to retrieve the quantum state. This method extensively leverages basics of probability theory to control errors, making it an efficient quantum-to-classical data converter [41]. Utilizing classical shadow, one can devise classical machine learning algorithms that exhibit only polynomial differences with full fault-tolerant QML programs, particularly when focusing on average error [66]. Moreover, this approach holds significant applications in QML for quantum data or quantum simulators, especially when a readout is required at the end.

4.1. Shadow tomography

Shadow tomography was a proposal first came up with by Scott Aaronson couple of years ago [2]. In this final section, we will review some key points of it and also a little about its recent

developments.

The foundation of tomography lies in the well-known fact of the destructive nature of measurements in quantum mechanics. When a quantum state or qubit, expressed as $\alpha\,|0\rangle + \beta\,|1\rangle$, undergoes a measurement, it collapses into either $|0\rangle$ or $|1\rangle$ with probabilities α^2 and β^2, respectively. Regardless of the outcome, the original state is lost and impossible to recover, a consequence of the No-Cloning Theorem, which prohibits the direct copying of quantum information.

The complexity increases significantly when dealing with n qubits. For a pure state of n qubits, obtaining an approximate description requires dealing with 2^n complex numbers. This exponential growth poses a challenge, as it would necessitate exponential classical bits to represent n qubits. Even for a relatively modest system of 1000 qubits, it would require 2^{1000} complex numbers for an approximate description. To put this into perspective, this exceeds the total number of atoms in the observable universe, estimated to be around 10^{80}. This realization, first acknowledged in the 1980s, played a crucial role in the conceptualization of quantum computing by visionaries like Richard Feynman [45] and others. But again measurement will yield at most n bits.

The quantum state tomography challenge involves a machine capable of generating n-qubit quantum states and producing identical copies of a specified state. Focusing on a mixed quantum state represented by a $D \times D$ density matrix, denoted as ρ, the goal is to gain insights into this density matrix. The key question is to determine the minimum number of copies of the state ρ necessary to effectively tackle this challenge. The answer to this question is approximately $\mathcal{O}(D^2)$ copies of ρ, which was established in 2016 [104]. This implies that the number of required copies scales polynomially with D, the dimension of the Hilbert space, which is still exponentially large. For instance, learning about 100 qubits would necessitate around 2^{200} copies of the state, an impractical and astronomically large number. Consequently, the feasibility of quantum state tomography becomes increasingly limited as the number of qubits grows, and one of the world records stands at about 10 qubits [118], involving millions of measurement settings. This underscores the significant challenges and limitations in quantum state tomography for larger quantum systems.

The concept of *shadow tomography*, introduced by Scott Aaronson [2, 3], is motivated by the following scenario. We are provided with an unknown D-dimensional mixed state ρ and a set of observables, $\mathcal{O}_1, \ldots, \mathcal{O}_M$. Assuming these observables can only take two values, such as *yes* or *no*, and considering that these measurements fully characterize the state, we aim to recover the "shadow" cast by ρ on the measurements $\mathcal{O}_1, \ldots, \mathcal{O}_M$. The term "shadow tomography" was suggested by Steve Flammia, emphasizing our goal of retrieving the "shadow" rather than the entire density matrix of ρ. The objective is to determine the minimum number of states ρ needed to estimate the expectation value $\Pr[\mathcal{O}_i] = \mathrm{Tr}(\mathcal{O}_i \rho)$ for all these observables.

Our intuition suggests that approximately $\mathcal{O}(D^2)$ copies would be adequate. We can bypass the entire shadow tomography setup and simply obtain a well-approximated version of the state ρ. Once we possess the state, accurately determining any expectation values becomes feasible. Furthermore, it seems intuitive that having roughly $\mathcal{O}(M)$ copies of the

state should enable straightforward shadow tomography. By conducting measurements on all M observables, we can derive the expectation values. Although the state is lost after each measurement, the availability of numerous copies remains. The latter will be practical if M is small, which means that we only care a small number of measurements or observables. But in many cases, both D and M are enormous. So the crucial question revolves around whether there exist a systematic way of accomplishing shadow tomography can be accomplished with a time complexity of $\mathrm{poly}(\log D, \log M)$.

In 2018, it was demonstrated in [2] that a protocol exists capable of achieving shadow tomography with a time complexity of $\mathrm{poly}(\log D, \log M)$.

Theorem 4.1. *Shadow Tomography Theorem*

   Shadow tomography is solvable using only $\tilde{\mathcal{O}}\left(\log^4 M \cdot \log D/\varepsilon^4\right)$ copies of the state ρ, where the $\tilde{\mathcal{O}}$ hides a higher order factor. The procedure is fully explicit.

In simpler terms, this protocol estimates the acceptance probability of all measurements, denoted as $\mathcal{O}_1, \ldots, \mathcal{O}_M$, with an additive error of $\pm\varepsilon$, utilizing only approximately $\tilde{\mathcal{O}}\left(\log^4 M \cdot \log D/\varepsilon^4\right)$ copies of ρ. While it's acknowledged that this is not the most efficient method and there is room for improvement, this protocol marked a significant milestone by demonstrating the theoretical feasibility of shadow tomography.

The implications of Theorem 4.1 include the following:

(i) For an n-qubit state $|\psi\rangle$, by measuring approximately $n^{\mathcal{O}(1)}$ copies of $|\psi\rangle$, one can understand its behavior on **every** quantum circuit with at most $p(n)$ gates, for any fixed polynomial p.

(ii) Given any $n^{\mathcal{O}(1)}$-qubit quantum program ρ, having $n^{\mathcal{O}(1)}$ copies of ρ allows us to estimate its acceptance probability on **every** n-bit input x.

To comprehend the workings of shadow tomography, a key concept is "gentle measurement" [4]. If a measurement of a mixed state ρ yields a certain outcome with a probability greater than or equal to $1 - \varepsilon$, then after the measurement, there exists a state ρ' that is $\sqrt{\varepsilon}$-close to ρ in trace distance. In simpler terms, if a state is close to an eigenstate of some observable, it is highly likely that the state will essentially collapse into that eigenstate, which isn't much different from the original state before the measurement occurred. An even stronger statement holds: applying M such measurements in succession, each accepting with a probability greater than or equal to $1 - 2M\sqrt{\varepsilon}$, ensures that the error grows linearly. Thus, if a sequence of measurements is applied, and each one does not significantly damage ρ, the cumulative effect of the entire session (assuming not too many measurements, i.e., small M) will also not significantly damage ρ.

The concept of gentle measurement is often paired with *amplification*. If one has many identical copies of a state ρ, and the measurement on one copy yields a certain outcome with a 90 percent probability, then with k copies of the state, an *amplified measurement* can be applied. This involves coherently measuring k copies of ρ and taking a **majority** to reduce the error to $1/\exp k$. Therefore, one can use amplification initially to reduce ε or the probability of a bad measurement to an exponentially small value with respect to k, the number of copies

measured. Subsequently, gentle measurement can be employed, ensuring that the damage to the state ρ each time a measurement is applied is only of an exponentially small amount.

One obvious question might be why gentle measurement does not solve our problem of shadow tomography immediately. Gentle measurement only provides a solution under a special case known as a **promise gap**. The gentle measurement will give us an answer under the following circumstance: given measurements $\mathcal{O}_1, \ldots, \mathcal{O}_M$ and real numbers $C_1, \ldots, C_M$, and being promised that $\Pr[\mathcal{O}_i \text{ accepts } \rho]$ is either at least C_i or at most $C_i - \varepsilon$ for each i, we can decide for each i using only $\mathcal{O}\left(\log M/\varepsilon^2\right)$ copies of ρ. This provides a good zone for gentle measurements to figure out which side of the two outcome measurements our state is close to, allowing gentle measurement to work. The problem arises when we are in the intermediate zone: $C_i - \varepsilon \geq \Pr[\mathcal{O}_i \text{ accepts } \rho] \geq C_i$. In this case, we will not be able to use this trick and accomplish shadow tomography. In other words, when there's no promise, we can never rule out that we're on the knife-edge between acceptance and rejection, making measuring ρ dangerous, since measurement there will not be gentle and destroy our state. And because our objective limits our copies of ρ to only $\log M$, we can't afford to destroy that many copies of the state.

The whole point of shadow tomography is to address this problem [2, 3]. The actual proof of Theorem 4.1 involves a combination of several tricks. The first one was intended to solve a different yet similar problem. Imagine Alice, who knows the state ρ, needs to send its information or description to Bob with respect to its behavior on measurements of observables $\mathcal{O}_1, \ldots, \mathcal{O}_M$, which we call $E_1, \ldots, E_M$. In contrast to shadow tomography, Alice does know the state at the beginning, but she has to send the information about the measurements in much fewer bits.

The following program will solve this problem, resembling a machine learning program. Initially, since Bob knows absolutely nothing, he will take a guess, and let's set it to the *maximally entangled state*: ρ_0. Then Alice helps Bob improve by repeatedly telling him a measurement with respect to an observable $\mathcal{O}_{i(t)}$ on which Bob's current guess ρ_{t-1} badly fails. Bob lets ρ_t be the state obtained by starting from ρ_{t-1}, then performing the measurement of $\mathcal{O}_{i(t)}$ and **postselecting** on getting the right outcome. After a few rounds, Bob has improved his state. To clarify, Bob is not trying to learn ρ at all but only ρ's behavior. He might end up with a state that is very far away in terms of trace distance from ρ yet still improves with respect to all the measurements that we care about. So the extreme situation is that Bob would guess randomly and obtain a very different state, yet this state has nothing to do with the measurements to improve upon. Then Bob is done on day one. The key point is for boosting weights type reasons; this process must converge after $T = \mathcal{O}(\log D)$ iterations, at some state ρ_T that behaves like ρ in terms of $E_1, \ldots, E_M$, even if it's far from ρ in trace distance.

Can we use this postselection learning protocol to solve the problem of shadow tomography? Unfortunately, it's still not enough by itself. Here, we relied on Alice to know the state ρ, and to know which measurement to be used that is best fit for postselection and updating Bob's guess. However, in the shadow tomography situation, there is no Alice. There is no one who knows the state and can tell you useful measurements to condition on. Rather,

you just have these copies of a ρ, and we have to figure it out for ourselves.

Another protocol to be combined with the previous one is called the **Quantum OR Bound** [59].

Theorem 4.2. *Quantum OR Bound*

Let ρ be an unknown mixed state, and let $E_1 ,\dots ,E_M$ be known 2-outcome measurements. Suppose we are promised that either

(i) *__there exists__ an i such that $\Pr[E_i$ accepts $\rho] \geq C$; or else*

(ii) $\Pr[E_i$ accepts $\rho] \leq C - \varepsilon$, *__for all__ $i \in [M]$.*

Then we can decide which, with high probability, given only $\mathcal{O}\left(\log M/\varepsilon^2\right)$ copies of ρ .

So what's new about shadow tomography is the combination of two protocols, **postselection learning** and the **Quantum OR Bound**. Scott Aaronson [2] found that you can use the Quantum OR Bound protocol as a subroutine to repeatedly search for informative measurements. Again, you start with the maximally mixed state and use the quantum OR bound repeatedly to search for measurements that can best improve your guess, round by round, to shape it into the one that has a similar outcome in terms of measurements with the target state ρ. Putting everything together, one will end up with Theorem 4.1.

The result in Theorem 4.1 is not the upper or lower bound; it simply shows that shadow tomography can be done in $\log M$ steps. Nowadays, the best lower bound people can prove is $\mathcal{O}(\log M/\varepsilon^2)$, which holds even if we are trying to learn a classical distribution [4]. So it is still unclear whether we need quantum Hilbert dimension D in Theorem 4.1. In [3], the authors were able to improve upon the $\log M$ dependence using a brand-new connection between gentle measurement and differential privacy. The sample complexity they have achieved is $k = \mathcal{O}\left(\log^2 M \cdot \log^2 D/\varepsilon^8\right)$.

Inspired by shadow tomography, [66] introduced the classical shadow protocol, as discussed in section 2.1.3. Thus, quantum machine learning is a rapidly developing field that encompasses various topics in physics and computer science. Although sub-fields may appear distinct, they all fall under the umbrella of quantum machine learning, influencing and potentially finding applications in one another. As a result, as Scott Aaronson states at the end of his paper [1], despite decades of research in quantum computing, researchers still marvel at the fact that the laws of quantum physics enable us to solve classical problems significantly faster than today's computers seem capable of. Therefore, it should perhaps not surprise us that, in machine learning and elsewhere, unlocking the full potential of quantum speedups requires significant effort.

4.2. Classical shadow formalism and random measurements

Drawing inspiration from the principles of shadow tomography, the classical shadow formalism has emerged as an intriguing topic with the potential to serve as an efficient quantum-classical information converter. A series of works, such as [63, 66, 41], have contributed to establishing a rigorous framework for reasoning about the randomized

measurement paradigm, a key component of classical shadow. These works not only delve into the conceptual aspects but also provide methods to derive error bounds for quantum information extraction based on randomized measurements, leveraging probability theory.

The challenges associated with the quantum-classical information interface, particularly in the context of NISQ algorithms (discussed in sections 2.1 and 2.2), as well as in the FTQC era for quantum machine learning algorithms (explored in section 3.3.1), highlight the necessity of efficient readout mechanisms. Researchers are actively addressing the need for reliable ways to input and extract information from quantum systems to classical ones, as the success of quantum machine learning programs relies on overcoming these challenges. If the process of loading classical data into a quantum computer or transforming quantum data into classical form takes an significant amount of time, the promised speed-ups of quantum algorithms become impractical. Therefore, not only is it crucial to find methods for transforming classical data into quantum form and vice versa, but the efficiency of these processes, including achieving significant speed-up, is paramount, given the exponential growth of Hilbert space degrees of freedom.

The most direct and comprehensive solution to the input problem is Quantum Random Access Memory (QRAM), as detailed in section 3.4. Additionally, alternative techniques like pruning, discussed in section 3.3.1, offer potential solutions to circumvent the pressing requirement for QRAM. Classical shadow, see also section 2.1.3, on the other hand, is focused on addressing the output problem. In this scenario, we possess a quantum state that encapsulates information trained by our Quantum Machine Learning (QML) program, and the goal is to decode this quantum state into classical information for straightforward interpretation.

To present this challenge in a broader context, the task is to efficiently estimate the expectation values of multiple structured observables. Specifically, the aim is to estimate the expectation value of various individual terms in Hamiltonians, where these terms share a well-defined structure, such as being Pauli operators. The proposed strategy for addressing this challenge involves diverting attention from the intricate details of the terms and instead uniformly sampling Pauli measurements. In simpler terms, the goal is to estimate the expectation values of Pauli observables, with the selection of the Pauli basis for each qubit being determined randomly during the measurement process [41].

This concept can be grasped intuitively through the lens of the infinite monkey theorem [43]. The theorem posits that a monkey randomly hitting keys on a typewriter for an infinite amount of time will almost certainly type any given text, including the complete works of William Shakespeare, and will likely produce every possible finite text an infinite number of times. Let's apply this idea to predicting an arbitrary low-weight Pauli observable, represented by the purple blocks in Fig. 9. The white blocks signify identity matrices, which are not of primary concern. The crucial aspect is the presence of three specific Pauli operators adjacent to each other in Fig. 9. Now, if we randomly assign Pauli measurements to each qubit, we are free to choose any Pauli measurements for the white blocks on the left side of Fig. 9. However, for the purple blocks, we must use the same Pauli measurements as on the left, or else we won't gain meaningful information about the statistics. To achieve this, random sampling

is employed. Importantly, the performance is not adversely affected, as the probability of obtaining such a low-weight Pauli string is exponentially small, but this exponentiation is in the weight of the Pauli, not in the number of qubits [43]. Additionally, if single-shot measurements are considered, there will be an oversampling factor.

Figure 9. A picturesque illustration of the infinite monkey theorem applied to randomized measurements.

The crucial aspect is that, since everything is done randomly, an effective union bound can be applied to estimate numerous low-weight Pauli observables. The union bound scales logarithmically with the number of terms. Consequently, if one randomly samples each Pauli measurement, the expected total number of measurements required to predict L low-weight observables with accuracy ϵ each is only logarithmic in L, thanks to the union bound [41].

To formally understand the classical shadow formalism, first we construct a quantum channel which transforms the quantum data into the classical data. We start with an unknown m qubit quantum state ρ and do a certain random unitary rotation: U , for example an independent single qubit Clifford rotation. Then we do a computational basis measurement on all of them. Since we did measurement at last, so the output of this channel is actually, honest to god, classical information. This, in Fig. 10, means that $U^\dagger|b\rangle\langle b|U$ although written in quantum language, is classical data. Hence, altogether, what this channel does is to transform the quantum state we got from the quantum computer and mapped it into classical information which we will do further computation.
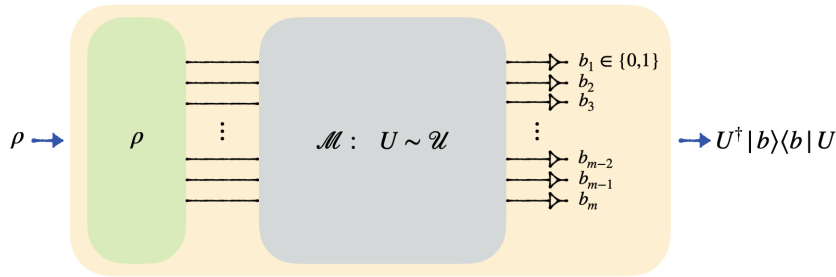

Figure 10. A quantum to classical channel.

The subsequent step involves averaging the random unitary sampling (through an integral over U), depending on the chosen ensemble, and also averaging over the computational basis outcomes (by summing over b) as per Born's rule. If the ensemble is well-behaved, such as Haar random, then the result after the averaging step can be efficiently computed. Moreover, since the result is classical information rather than quantum, it can be easily inverted to obtain the state ρ. Another example may be found in [23], where the authors consider classical shadows with unitary ensembles that are invariant under multiplication by Pauli operators. Consequently, the state ρ is obtained by performing random measurements to acquire classical

data, taking average, and then inverting the result to retrieve the state. The lingering question pertains to the convergence speed of this program to the desired expectation value. Drawing inspiration from shadow tomography, the authors of [62] demonstrated that the number of required snapshots or single measurements in this program scales logarithmically with the number of terms needed to predict: $N \gtrsim \log(L) \max_j ||\mathcal{O}_j||^2_{\text{shadow}}/\epsilon^2$, where $\mathcal{O}_j$ is the observable to be measured, and ϵ denotes the desired accuracy.

In 2021, the authors of [119] conducted an experimental implementation of classical shadow, demonstrating its superiority over maximum likelihood in the low sampling regime. This framework's applicability extends to the prediction of observable polynomials, as discussed in [41]. The strategy involves expressing polynomials as linear combinations of tensor products and substituting the tensor products with independent shadows, enhancing efficiency in probing entanglements in many-body systems. Notably, this method exhibits robustness in the presence of noise, leveraging random unitaries as detailed in [30]. The authors of [75] propose an error-mitigated classical shadow estimation scheme. This result complements [30]. [135] introduced an error-mitigated fermionic classical shadow scheme. Additionally, [65] explores techniques for derandomizing the procedure. In classical computer science, there exists a standard approach to transform a probabilistic argument into a deterministic strategy, maintaining comparable effectiveness without reliance on randomness.

4.2.1. Machine learning with classical shadow   In summary, classical shadows serve as an efficient quantum-to-classical converter, enabling the extraction of classical information from quantum states. This information can be used to predict observable expectation values. When considering the Variational Quantum Eigensolver (VQE) in a classical context, it resembles a kernel method. The process involves initiating with parameters that define the circuit, creating a map in a high-dimensional space. These parameters are then used to map to a state generated by the circuit, constructing an objective function with the Hamiltonian to acquire classical information. Moreover, classical shadows have demonstrated significant performance gains for VQE [41], prompting exploration into their potential implementation for more sophisticated machine learning programs.

Consider the problem of predicting ground state properties, where we assume a family of parameterized Hamiltonians, denoted as $H(\boldsymbol{x})$, with $\boldsymbol{x}$ being a classical input characterizing the Hamiltonian (e.g., Heisenberg Hamiltonian with $\boldsymbol{x}$ representing couplings). The goal is to learn how to predict ground state properties, such as observable expectation values, for ground states associated with Hamiltonians that were not part of the training data.

Classically, this problem is challenging. However, leveraging quantum training data and classical shadow measurements simplifies the task. Here's how the process unfolds:

(i) **Learning Phase**:
   - Random input parameters are sampled, and the associated ground states are prepared in the laboratory.
   - Classical shadow measurements are performed on these ground states for randomly chosen single qubits.

- The training data comprises classical inputs ($\boldsymbol{x}$) associated with classical shadow single-shot measurements of the corresponding ground states.
- The goal is to train a classical kernel method using these classical shadows.

(ii) **Prediction Phase**:
- In this phase, a Fourier-type kernel function is employed to convolve all the classical shadows observed during the learning phase.
- This convolution provides an approximation of the ground state for Hamiltonians not directly observed.

The central concept involves using classical shadows as a connection between quantum training data and classical predictions. This approach facilitates the learning of ground state properties for Hamiltonians that were not part of the initial training set. The goal is to establish an efficient representation for future Hamiltonians, enabling the computation of observable expectation values and making predictions. This program has been demonstrated to be effective. In [68], the authors established that for a small average case prediction error (averaged over all possible inputs), a scenario common in machine learning, the required amount of training data is not extensive. Instead, it scales quasi-polynomially in the number of inputs, not the number of qubits.

In the study conducted by [66], the authors delved deeper into the efficacy of this method. They utilized a quantum device to generate data, employed classical shadows to effectively utilize the quantum data, and then applied classical machine learning algorithms for training. They compared this approach (classical machine learning with quantum data leading to classical shadows) to a hypothetical full-scale quantum machine learning protocol. The latter runs on a quantum computer and has coherent access to training data through QRAM. This is the most powerful program that we can envision in the future of QML. The results indicated that, when focusing on average error, there is no significant separation between quantum and classical training data size in terms of sample complexity. However, for worst-case prediction error, [66] presented concrete examples demonstrating an exponential separation in complexity. In the research presented in [67], the authors explore analogous approaches to learning. They investigate how classical and quantum agents, given their distinct methods of receiving, processing, and storing information, differ in efficiency when aiming to characterize a physical system or state ρ. The study demonstrates that quantum agents can achieve exponential advantages in experiments, particularly in terms of the number of queries required to obtain ρ.

4.3. Quantum machine learning for quantum data and quantum simulators

The most immediate application of quantum machine learning lies in handling quantum data [18], i.e., the states produced by quantum systems and processes. Traditional quantum machine learning algorithms, as discussed earlier, identify patterns in classical data by transforming it into quantum states and manipulating those states using basic quantum linear algebra operations. These same algorithms can be directly employed on quantum states of light and matter to uncover inherent features and patterns. The quantum methods of analysis

often prove to be more efficient and insightful than classical approaches when dealing with data obtained from quantum systems.

For instance, in the case of multiple copies of a system represented by an $N \times N$ density matrix, quantum principal component analysis can determine eigenvalues and reveal corresponding eigenvectors in a time complexity of $\mathcal{O}\left((\log_2 N)^2\right)$. This stands in contrast to the $\mathcal{O}(N^2)$ measurements required for classical devices to perform tomography on the density matrix and the $\mathcal{O}(N^2)$ operations needed for classical PCA. This quantum analysis of quantum data can be efficiently conducted on the relatively modest-scale quantum computers expected to be available in the coming years. As mentioned before, [67] also demonstrated the quantum advantages of learning a physical state ρ through a quantum agent for example quantum sensor or simulator discussed above.

Quantum simulators serve as powerful tools for analyzing quantum dynamics. These are essentially "quantum analog computers" capable of emulating the dynamics of specific quantum systems. Quantum simulators can be specialized devices designed for simulating particular classes of quantum systems or general-purpose quantum computers. The approach involves connecting a trusted quantum simulator to an unknown system, adjusting the simulator's model to counteract the unknown dynamics, and efficiently learning the dynamics of the unknown system using approximate Bayesian inference [18, 48, 132, 133]. This significantly reduces the number of measurements required for simulation, offering an exponential improvement.

Notably, the universal quantum emulator algorithm [96] enables the reconstruction of quantum dynamics, and the quantum Boltzmann training algorithm [18] allows for the reconstruction of states with a time complexity logarithmic in the dimension of the Hilbert space. This is significantly faster than reconstructing dynamics through classical tomography.

While using a quantum computer to characterize a quantum system [132, 133] or input states for quantum PCA poses technical challenges, as it doesn't require QRAM, it holds promise for near-term applications of quantum machine learning [132, 133, 18], offering the potential for significant speedups in device characterization.

5. Conclusion and discussion

This review highlights that both current NISQ-era quantum devices and future fully fault-tolerant quantum computers (FTQC) hold significant promise for applications in machine learning and data analysis [17, 18, 87, 89]. Over the past decade, quantum computing has witnessed notable advancements in applications, experimental validations, and theoretical findings. The quantum computation field, especially in NISQ applications, has seen an almost exponential growth in the number of research papers. Various factors contribute to this trend, including substantial enhancements in quantum hardware [17, 67]. Given that quantum computing is a relatively young scientific discipline, there is ample opportunity for groundbreaking research and discoveries. The presence of theoretical, practical, and experimental challenges, many of which are addressed in this review, further underscores the motivation for an open-source approach in the field.

As mentioned in [17], we expect experimental pursuit in the NISQ era would focus on the design of quantum hardware with a larger number of qubits, and gates with lower error rates capable of executing deeper circuits. Along the way, one of the goals is to demonstrate quantum advantage for practical use cases. If the NISQ paradigm is not powerful enough to exhibit any quantum advantage, theoretical pursuits would be required to understand its limitations. The prime direction of the NISQ and near-term era is to engineer the best possible solution with the limited quantum resources available. The tools and techniques invented during this period could be valuable in the fault- tolerant era as well. To conduct a successful demonstration of quantum advantage, the right blend of the two crucial components: hardware and algorithms designs, is required [17]. First, hardware development is the key. The design of quantum computers with more qubits, lesser error rates, longer coherence times, and more connectivity between the qubits will be one of the top priorities in the NISQ era. Intensive research in new qubits developments, quantum optimal control and material discovery will be indispensable for both universal programmable quantum computers or special purpose ones. Secondly, to harness the potential of noisy but powerful quantum devices, we expect breakthroughs on the algorithm frontier. Algorithms with realistic assumptions, as the ones mentioned in [17], regarding device capabilities will be favored. To lessen the effect of noise, progress towards the design of error suppression, mitigation and correction methods is expected.

While QML has been proposed as a potential avenue for achieving scientific value in the near term using NISQ devices, questions arise regarding its applicability in the future [27]. Researchers envision two distinct post-NISQ eras. The first, termed "partial error corrected," assumes that quantum computers will possess a sufficient number of physical qubits (a few hundred) and low error rates, allowing for a small number of fully error-corrected logical qubits. In this era, users can allocate qubits between error-corrected and non-error-corrected subsets. The subsequent era, labeled the "fault-tolerant era," will emerge when quantum hardware features a large number of error-corrected qubits. In this era, algorithms like those discussed in section 3.3.1 could notably enhance the scalability and sustainability of classical large-scale machine learning models. Notably, works such as [90] offer solid theoretical assurances and intersections with cutting-edge classical machine learning research. This approach diverges from the variational quantum algorithms mindset, exemplified by [87], by aiming to enhance classical machine learning through a key quantum step that acts as a bottleneck for classical training. Developments in shadow tomography and related fields [2, 41, 66] demonstrate the potential for systematically analyzing QML and gaining a deeper understanding of how quantum speedup is achieved. As demonstrated experimentally in [67], with quantum technology such as quantum sensor, quantum memory, and quantum computer, our ability to learn about the physical world might be exponentially improved. Hence, there is confidence that in the future, researchers will have a much more robust understanding of how quantum computing achieves speedup and can develop more powerful applications for machine learning problems.

Acknowledgement

We thank Hsin-Yuan Huang and Kanav Setia for helpful discussions. JL is supported in part by International Business Machines (IBM) Quantum through the Chicago Quantum Exchange, and the Pritzker School of Molecular Engineering at the University of Chicago through AFOSR MURI (FA9550-21-1-0209).

References

[1] S. Aaronson. Read the fine print. *Nature Physics*, 11(4):291–293, 2015.

[2] S. Aaronson. Shadow tomography of quantum states. In *Proceedings of the 50th annual ACM SIGACT symposium on theory of computing*, pages 325–338, 2018.

[3] S. Aaronson. Shadow tomography of quantum states. *SIAM Journal on Computing*, 49(5):STOC18–368–STOC18–394, 2020. doi: 10.1137/18M120275X. URL `https://doi.org/10.1137/18M120275X`.

[4] S. Aaronson and G. N. Rothblum. Gentle measurement of quantum states and differential privacy, 2019.

[5] D. Amaro, C. Modica, M. Rosenkranz, M. Fiorentini, M. Benedetti, and M. Lubasch. Filtering variational quantum algorithms for combinatorial optimization. *Quantum Science and Technology*, 7(1):015021, Jan. 2022. ISSN 2058-9565. doi: 10.1088/2058-9565/ac3e54. URL `http://dx.doi.org/10.1088/2058-9565/ac3e54`.

[6] A. Anand, M. Degroote, and A. Aspuru-Guzik. Natural evolutionary strategies for variational quantum computation. *Machine Learning: Science and Technology*, 2(4):045012, jul 2021. doi: 10.1088/2632-2153/abf3ac. URL `https://doi.org/10.1088%2F2632-2153%2Fabf3ac`.

[7] A. Andreassen, I. Feige, C. Frye, and M. D. Schwartz. Junipr: a framework for unsupervised machine learning in particle physics. *The European Physical Journal C*, 79:1–24, 2019.

[8] A. Arrasmith, M. Cerezo, P. Czarnik, L. Cincio, and P. J. Coles. Effect of barren plateaus on gradient-free optimization. *Quantum*, 5:558, Oct. 2021. ISSN 2521-327X. doi: 10.22331/q-2021-10-05-558. URL `http://dx.doi.org/10.22331/q-2021-10-05-558`.

[9] A. Arrasmith, Z. Holmes, M. Cerezo, and P. J. Coles. Equivalence of quantum barren plateaus to cost concentration and narrow gorges. *Quantum Science and Technology*, 7(4):045015, Aug. 2022. ISSN 2058-9565. doi: 10.1088/2058-9565/ac7d06. URL `http://dx.doi.org/10.1088/2058-9565/ac7d06`.

[10] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, and P. V. Srinivasan. On the robustness of bucket brigade quantum ram. *New Journal of Physics*, 17(12):123010, Dec. 2015. ISSN 1367-2630. doi: 10.1088/1367-2630/17/12/123010. URL `http://dx.doi.org/10.1088/1367-2630/17/12/123010`.

[11] H. J. M. J. au2, A. Zaman, and H. Y. Wong. Step-by-step hhl algorithm walkthrough to enhance the understanding of critical quantum computing concepts, 2023.

[12] B. Barak and K. Marwaha. Classical algorithms and quantum limitations for maximum cut on high-girth graphs. *arXiv preprint arXiv:2106.05900*, 2021.

[13] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner. Improving variational quantum optimization using cvar. *Quantum*, 4:256, 2020.

[14] S. Becker, D. Englund, and B. Stiller. An optoacoustic field-programmable perceptron for recurrent neural networks. *arXiv preprint arXiv:2309.01543*, 2023.

[15] M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Information*, 5(1):45, 2019.

[16] M. Benedetti, M. Fiorentini, and M. Lubasch. Hardware-efficient variational quantum algorithms for time evolution. *Physical Review Research*, 3(3), July 2021. ISSN 2643-1564. doi: 10.1103/physrevresearch.3.033083. URL `http://dx.doi.org/10.1103/PhysRevResearch.3.033083`.

[17] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, et al. Noisy intermediate-scale quantum (nisq) algorithms (2021). *arXiv preprint arXiv:2101.08448*, 2021.

[18] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[19] M. Bilkis, M. Cerezo, G. Verdon, P. J. Coles, and L. Cincio. A semi-agnostic ansatz with variable structure for variational quantum algorithms. *Quantum Machine Intelligence*, 5(2), 2023. ISSN 2524-4914. doi: 10.1007/s42484-023-00132-1. URL `http://dx.doi.org/10.1007/s42484-023-00132-1`.

[20] L. Bittel and M. Kliesch. Training variational quantum algorithms is np-hard. *Phys. Rev. Lett.*, 127:120502, Sep 2021. doi: 10.1103/PhysRevLett.127.120502. URL `https://link.aps.org/doi/10.1103/PhysRevLett.127.120502`.

[21] K. Bu, D. E. Koh, L. Li, Q. Luo, and Y. Zhang. Statistical complexity of quantum circuits. *Physical Review A*, 105(6), June 2022. ISSN 2469-9934. doi: 10.1103/physreva.105.062431. URL `http://dx.doi.org/10.1103/PhysRevA.105.062431`.

[22] K. Bu, D. E. Koh, L. Li, Q. Luo, and Y. Zhang. Effects of quantum resources and noise on the statistical complexity of quantum circuits. *Quantum Science and Technology*, 8(2):025013, Feb. 2023. ISSN 2058-9565. doi: 10.1088/2058-9565/acb56a. URL `http://dx.doi.org/10.1088/2058-9565/acb56a`.

[23] K. Bu, D. E. Koh, R. J. Garcia, and A. Jaffe. Classical shadows with pauli-invariant unitary ensembles. *npj Quantum Information*, 10(1), Jan. 2024. ISSN 2056-6387. doi: 10.1038/s41534-023-00801-w. URL `http://dx.doi.org/10.1038/s41534-023-00801-w`.

[24] D. Bultrini, S. Wang, P. Czarnik, M. H. Gordon, M. Cerezo, P. J. Coles, and L. Cincio. The battle of clean and dirty qubits in the era of partial error correction. *Quantum*, 7:1060, July 2023. ISSN 2521-327X. doi: 10.22331/q-2023-07-13-1060. URL `http://dx.doi.org/10.22331/q-2023-07-13-1060`.

[25] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1), Mar. 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-21728-w. URL `http://dx.doi.org/10.1038/s41467-021-21728-w`.

[26] M. Cerezo, K. Sharma, A. Arrasmith, and P. J. Coles. Variational quantum state eigensolver. *npj Quantum Information*, 8(1):113, 2022.

[27] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles. Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576, 2022.

[28] E. Cervero Martín, K. Plekhanov, and M. Lubasch. Barren plateaus in quantum tensor network optimization. *Quantum*, 7:974, Apr. 2023. ISSN 2521-327X. doi: 10.22331/q-2023-04-13-974. URL `http://dx.doi.org/10.22331/q-2023-04-13-974`.

[29] H.-H. Chang. An introduction to error-correcting codes: From classical to quantum, 2006.

[30] S. Chen, W. Yu, P. Zeng, and S. T. Flammia. Robust shadow estimation. *PRX Quantum*, 2(3), Sept. 2021. ISSN 2691-3399. doi: 10.1103/prxquantum.2.030348. URL `http://dx.doi.org/10.1103/PRXQuantum.2.030348`.

[31] S. Cheng, J. Chen, and L. Wang. Information perspective to probabilistic modeling: Boltzmann machines versus born machines. *Entropy*, 20(8):583, 2018.

[32] N.-H. Chia, A. Gilyén, T. Li, H.-H. Lin, E. Tang, and C. Wang. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC '20. ACM, June 2020. doi: 10.1145/3357713.3384314. URL `http://dx.doi.org/10.1145/3357713.3384314`.

[33] L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming, 2020.

[34] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig. Quantum machine learning: a classical perspective. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2209): 20170551, Jan. 2018. ISSN 1471-2946. doi: 10.1098/rspa.2017.0551. URL `http://dx.doi.org/10.1098/rspa.2017.0551`.

[35] L. Cincio, K. Rudinger, M. Sarovar, and P. J. Coles. Machine learning of noise-resilient quantum circuits. *PRX Quantum*, 2(1), Feb. 2021. ISSN 2691-3399. doi: 10.1103/prxquantum.2.010324. URL `http://dx.doi.org/10.1103/PRXQuantum.2.010324`.

[36] J. Cotler, N. Hunter-Jones, J. Liu, and B. Yoshida. Chaos, complexity, and random matrices. *Journal of High Energy Physics*, 2017(11), Nov. 2017. ISSN 1029-8479. doi: 10.1007/jhep11(2017)048. URL `http://dx.doi.org/10.1007/JHEP11(2017)048`.

[37] J. Cotler, H.-Y. Huang, and J. R. McClean. Revisiting dequantization and quantum advantage in learning tasks. *arXiv preprint arXiv:2112.00811*, 2021.

[38] P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio. Error mitigation with clifford quantum-circuit data. *Quantum*, 5:592, Nov. 2021. ISSN 2521-327X. doi: 10.22331/q-2021-11-26-592. URL `http://dx.doi.org/10.22331/q-2021-11-26-592`.

[39] A. Deshpande, P. Niroula, O. Shtanko, A. V. Gorshkov, B. Fefferman, and M. J. Gullans. Tight bounds on the convergence of noisy random circuits to the uniform distribution. *PRX Quantum*, 3(4), Dec. 2022. ISSN 2691-3399. doi: 10.1103/prxquantum.3.040329. URL `http://dx.doi.org/10.1103/PRXQuantum.3.040329`.

[40] V. Dunjko and P. Wittek. A non-review of quantum machine learning: trends and explorations. *Quantum Views*, 4:32, 03 2020. doi: 10.22331/qv-2020-03-17-32.

[41] A. Elben, S. T. Flammia, H.-Y. Huang, R. Kueng, J. Preskill, B. Vermersch, and P. Zoller. The randomized measurement toolbox. *Nature Reviews Physics*, 5(1): 9–24, Dec. 2022. ISSN 2522-5820. doi: 10.1038/s42254-022-00535-2. URL `http://dx.doi.org/10.1038/s42254-022-00535-2`.

[42] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan. Hybrid quantum-classical algorithms and quantum error mitigation. *Journal of the Physical Society of Japan*, 90(3):032001, 2021.

[43] T. J. Evans, R. Harper, and S. T. Flammia. Scalable bayesian hamiltonian learning, 2019.

[44] E. Farhi, J. Goldstone, and S. Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[45] R. Feynman, A. Hey, and R. Allen. *Lectures On Computation*. Advanced book program. Basic Books, 1996. ISBN 9780201489910. URL `https://books.google.com.tw/books?id=-olQAAAAMAAJ`.

[46] A. Garcia-Saez and J. Riu. Quantum observables for continuous control of the quantum approximate optimization algorithm via reinforcement learning, 2019.

[47] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC '19. ACM, June 2019. doi: 10.1145/3313276.3316366. URL `http://dx.doi.org/10.1145/3313276.3316366`.

[48] C. E. Granade, C. Ferrie, N. Wiebe, and D. G. Cory. Robust online hamiltonian learning. *New Journal of Physics*, 14(10):103013, Oct. 2012. ISSN 1367-2630.

doi: 10.1088/1367-2630/14/10/103013. URL `http://dx.doi.org/10.1088/1367-2630/14/10/103013`.

[49] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature communications*, 10(1):3007, 2019.

[50] L. K. Grover. A fast quantum mechanical algorithm for database search, 1996.

[51] M. V. Gustafsson, T. Aref, A. F. Kockum, M. K. Ekström, G. Johansson, and P. Delsing. Propagating phonons coupled to an artificial atom. *Science*, 346(6206):207–211, 2014. ISSN 1095-9203. doi: 10.1126/science.1257219. URL `http://dx.doi.org/10.1126/science.1257219`.

[52] H. HAFFNER, C. ROOS, and R. BLATT. Quantum computing with trapped ions. *Physics Reports*, 469(4):155–203, dec 2008. doi: 10.1016/j.physrep.2008.09.003. URL `https://doi.org/10.1016%2Fj.physrep.2008.09.003`.

[53] S. Hakkaku, Y. Tashima, K. Mitarai, W. Mizukami, and K. Fujii. Quantifying fermionic nonlinearity of quantum circuits. *Physical Review Research*, 4(4), Nov. 2022. ISSN 2643-1564. doi: 10.1103/physrevresearch.4.043100. URL `http://dx.doi.org/10.1103/PhysRevResearch.4.043100`.

[54] C. T. Hann. *Practicality of Quantum Random Access Memory*. PhD thesis, Yale University, 2021.

[55] C. T. Hann, C.-L. Zou, Y. Zhang, Y. Chu, R. J. Schoelkopf, S. Girvin, and L. Jiang. Hardware-efficient quantum random access memory with hybrid quantum acoustic systems. *Physical Review Letters*, 123(25), 2019. ISSN 1079-7114. doi: 10.1103/physrevlett.123.250501. URL `http://dx.doi.org/10.1103/PhysRevLett.123.250501`.

[56] C. T. Hann, C.-L. Zou, Y. Zhang, Y. Chu, R. J. Schoelkopf, S. M. Girvin, and L. Jiang. Hardware-efficient quantum random access memory with hybrid quantum acoustic systems. *Phys. Rev. Lett.*, 123:250501, Dec 2019.

[57] A. W. Harrow and J. C. Napp. Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms. *Physical Review Letters*, 126(14), apr 2021. doi: 10.1103/physrevlett.126.140502. URL `https://doi.org/10.1103%2Fphysrevlett.126.140502`.

[58] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[59] A. W. Harrow, C. Y.-Y. Lin, and A. Montanaro. Sequential measurements, disturbance and property testing. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, Jan. 2017. doi: 10.1137/1.9781611974782.105. URL `http://dx.doi.org/10.1137/1.9781611974782.105`.

[60] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *PRX Quantum*, 3(1), Jan. 2022. ISSN

2691-3399. doi: 10.1103/prxquantum.3.010313. URL `http://dx.doi.org/10.1103/PRXQuantum.3.010313`.

[61] H.-Y. Huang, K. Bharti, and P. Rebentrost. Near-term quantum algorithms for linear systems of equations. *arXiv e-prints*, art. arXiv:1909.07344, Sept. 2019. doi: 10.48550/arXiv.1909.07344.

[62] H.-Y. Huang, R. Kueng, and J. Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, jun 2020. doi: 10.1038/s41567-020-0932-7. URL `https://doi.org/10.1038%2Fs41567-020-0932-7`.

[63] H.-Y. Huang, R. Kueng, and J. Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, June 2020. ISSN 1745-2481. doi: 10.1038/s41567-020-0932-7. URL `http://dx.doi.org/10.1038/s41567-020-0932-7`.

[64] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean. Power of data in quantum machine learning. *Nature Communications*, 12 (1), may 2021. doi: 10.1038/s41467-021-22539-9. URL `https://doi.org/10.1038%2Fs41467-021-22539-9`.

[65] H.-Y. Huang, R. Kueng, and J. Preskill. Efficient estimation of pauli observables by derandomization. *Phys. Rev. Lett.*, 127:030503, Jul 2021. doi: 10.1103/PhysRevLett.127.030503. URL `https://link.aps.org/doi/10.1103/PhysRevLett.127.030503`.

[66] H.-Y. Huang, R. Kueng, and J. Preskill. Information-theoretic bounds on quantum advantage in machine learning. *Physical Review Letters*, 126(19):190505, 2021.

[67] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill, et al. Quantum advantage in learning from experiments. *Science*, 376(6598):1182–1186, 2022.

[68] H.-Y. Huang, R. Kueng, G. Torlai, V. V. Albert, and J. Preskill. Provably efficient machine learning for quantum many-body problems. *Science*, 377(6613), Sept. 2022. ISSN 1095-9203. doi: 10.1126/science.abk3333. URL `http://dx.doi.org/10.1126/science.abk3333`.

[69] P. Jain and P. Kar. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3–4):142–336, 2017. ISSN 1935-8245. doi: 10.1561/2200000058. URL `http://dx.doi.org/10.1561/2200000058`.

[70] C. Jin, P. Netrapalli, R. Ge, S. M. Kakade, and M. I. Jordan. On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points, 2019.

[71] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *nature*, 549(7671):242–246, 2017.

[72] I. Kerenidis and A. Prakash. Quantum recommendation systems, 2016.

[73] B. T. Kiani, S. Lloyd, and R. Maity. Learning unitaries by gradient descent, 2020.

[74] I. D. Kivlichan, J. McClean, N. Wiebe, C. Gidney, A. Aspuru-Guzik, G. K.-L. Chan, and R. Babbush. Quantum simulation of electronic structure with linear depth and connectivity. *Physical review letters*, 120(11):110501, 2018.

[75] D. E. Koh and S. Grewal. Classical shadows with noise. *Quantum*, 6:776, Aug. 2022. ISSN 2521-327X. doi: 10.22331/q-2022-08-16-776. URL http://dx.doi.org/10.22331/q-2022-08-16-776.

[76] J. S. Kottmann, M. Krenn, T. H. Kyaw, S. Alperin-Lea, and A. Aspuru-Guzik. Quantum computer-aided design of quantum optics hardware. *Quantum Science and Technology*, 6(3):035010, 2021.

[77] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver. A quantum engineer's guide to superconducting qubits. *Applied physics reviews*, 6(2), 2019.

[78] M. Krenn, M. Erhard, and A. Zeilinger. Computer-inspired quantum experiments. *Nature Reviews Physics*, 2(11):649–661, 2020.

[79] M. Larocca, N. Ju, D. Garcia-Martin, P. J. Coles, and M. Cerezo. Theory of overparametrization in quantum neural networks. *Nature Computational Science*, 3 (6):542–551, 2023. ISSN 2662-8457. doi: 10.1038/s43588-023-00467-6. URL http://dx.doi.org/10.1038/s43588-023-00467-6.

[80] R. LaRose and B. Coyle. Robust data encodings for quantum classifiers. *Physical Review A*, 102(3), Sept. 2020. ISSN 2469-9934. doi: 10.1103/physreva.102.032420. URL http://dx.doi.org/10.1103/PhysRevA.102.032420.

[81] R. LaRose, A. Tikku, E. O'Neel-Judy, L. Cincio, and P. J. Coles. Variational quantum state diagonalization. *npj Quantum Information*, 5(1), 2019. ISSN 2056-6387. doi: 10.1038/s41534-019-0167-6. URL http://dx.doi.org/10.1038/s41534-019-0167-6.

[82] J. Lee, W. J. Huggins, M. Head-Gordon, and K. B. Whaley. Generalized unitary coupled cluster wave functions for quantum computation. *Journal of chemical theory and computation*, 15(1):311–324, 2018.

[83] L. Li, M. Fan, M. Coram, P. Riley, S. Leichenauer, et al. Quantum optimization with a novel gibbs objective function and ansatz architecture search. *Physical Review Research*, 2(2):023074, 2020.

[84] J. Liu. Spectral form factors and late time quantum chaos. *Physical Review D*, 98 (8), Oct. 2018. ISSN 2470-0029. doi: 10.1103/physrevd.98.086026. URL http://dx.doi.org/10.1103/PhysRevD.98.086026.

[85] J. Liu. Scrambling and decoding the charged quantum information. *Physical Review Research*, 2(4), Oct. 2020. ISSN 2643-1564. doi: 10.1103/physrevresearch.2.043164. URL http://dx.doi.org/10.1103/PhysRevResearch.2.043164.

[86] J. Liu, Z. Lin, and L. Jiang. Laziness, barren plateau, and noise in machine learning, 2022.

[87] J. Liu, F. Tacchino, J. R. Glick, L. Jiang, and A. Mezzacapo. Representation learning via quantum neural tangent kernels. *PRX Quantum*, 3(3):030323, 2022.

[88] J. Liu, F. Wilde, A. A. Mele, L. Jiang, and J. Eisert. Noise can be helpful for variational quantum algorithms. *arXiv preprint arXiv:2210.06723*, 2022.

[89] J. Liu, M. Liu, J.-P. Liu, Z. Ye, Y. Wang, Y. Alexeev, J. Eisert, and L. Jiang. Towards provably efficient quantum algorithms for large-scale machine-learning models. *arXiv preprint arXiv:2303.03428*, 2023.

[90] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proceedings of the National Academy of Sciences*, 118(35), 2021. ISSN 1091-6490. doi: 10.1073/pnas.2026805118. URL `http://dx.doi.org/10.1073/pnas.2026805118`.

[91] S. Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.

[92] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum algorithms for supervised and unsupervised machine learning, 2013.

[93] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, July 2014. ISSN 1745-2481. doi: 10.1038/nphys3029. URL `http://dx.doi.org/10.1038/nphys3029`.

[94] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch. Variational quantum algorithms for nonlinear problems. *Physical Review A*, 101(1), Jan. 2020. ISSN 2469-9934. doi: 10.1103/physreva.101.010301. URL `http://dx.doi.org/10.1103/PhysRevA.101.010301`.

[95] C. O. Marrero, M. Kieferová, and N. Wiebe. Entanglement induced barren plateaus, 2021.

[96] I. Marvian and S. Lloyd. Universal quantum emulator, 2016.

[97] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[98] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9 (1), Nov. 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-07090-4. URL `http://dx.doi.org/10.1038/s41467-018-07090-4`.

[99] K. Mitarai and K. Fujii. Methodology for replacing indirect measurements with direct measurements. *Physical Review Research*, 1(1), aug 2019. doi: 10.1103/physrevresearch.1.013006. URL `https://doi.org/10.1103%2Fphysrevresearch.1.013006`.

[100] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, et al. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, 3(3): 030503, 2018.

[101] K. M. Nakanishi, K. Fujii, and S. Todo. Sequential minimal optimization for quantum-classical hybrid algorithms. *Physical Review Research*, 2(4), oct 2020. doi: 10.1103/physrevresearch.2.043158. URL https://doi.org/10.1103%2Fphysrevresearch.2.043158.

[102] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. doi: 10.1017/CBO9780511976667.

[103] A. O'Connell, M. Hofheinz, M. Ansmann, R. C. Bialczak, M. Lenander, E. Lucero, M. Neeley, D. Sank, H. Wang, M. Weides, J. Wenner, J. M. Martinis, and A. Cleland. Quantum ground state and single-phonon control of a mechanical resonator. *Nature*, 464(7289):697—703, April 2010. ISSN 0028-0836. doi: 10.1038/nature08967. URL https://doi.org/10.1038/nature08967.

[104] R. O'Donnell and J. Wright. Efficient quantum tomography, 2015.

[105] M. Ostaszewski, E. Grant, and M. Benedetti. Structure optimization for parameterized quantum circuits. *Quantum*, 5:391, jan 2021. doi: 10.22331/q-2021-01-28-391. URL https://doi.org/10.22331%2Fq-2021-01-28-391.

[106] T. L. Patti, K. Najafi, X. Gao, and S. F. Yelin. Entanglement devised barren plateau mitigation. *Physical Review Research*, 3(3), July 2021. ISSN 2643-1564. doi: 10.1103/physrevresearch.3.033090. URL http://dx.doi.org/10.1103/PhysRevResearch.3.033090.

[107] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.

[108] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):4213, 2014.

[109] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft, April 1998.

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

[111] D. A. Roberts and B. Yoshida. Chaos and complexity by design. *Journal of High Energy Physics*, 2017(4), Apr. 2017. ISSN 1029-8479. doi: 10.1007/jhep04(2017)121. URL http://dx.doi.org/10.1007/JHEP04(2017)121.

[112] D. A. Roberts, S. Yaida, and B. Hanin. *The principles of deep learning theory*. Cambridge University Press Cambridge, MA, USA, 2022.

[113] J. Roffe. Quantum error correction: an introductory guide. *Contemporary Physics*, 60(3):226–245, July 2019. ISSN 1366-5812. doi: 10.1080/00107514.2019.1667078. URL http://dx.doi.org/10.1080/00107514.2019.1667078.

[114] J. J. Sakurai and J. Napolitano. *Modern Quantum Mechanics*. Quantum physics, quantum information and quantum computation. Cambridge University Press, 10 2020. ISBN 978-0-8053-8291-4, 978-1-108-52742-2, 978-1-108-58728-0. doi: 10.1017/9781108587280.

[115] K. Sharma, S. Khatri, M. Cerezo, and P. J. Coles. Noise resilience of variational quantum compiling. *New Journal of Physics*, 22(4):043006, Apr. 2020. ISSN 1367-2630. doi: 10.1088/1367-2630/ab784c. URL `http://dx.doi.org/10.1088/1367-2630/ab784c`.

[116] K. Sharma, M. Cerezo, L. Cincio, and P. J. Coles. Trainability of dissipative perceptron-based quantum neural networks. *Physical Review Letters*, 128(18), May 2022. ISSN 1079-7114. doi: 10.1103/physrevlett.128.180505. URL `http://dx.doi.org/10.1103/PhysRevLett.128.180505`.

[117] P. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994. doi: 10.1109/SFCS.1994.365700.

[118] C. Song, K. Xu, W. Liu, C.-p. Yang, S.-B. Zheng, H. Deng, Q. Xie, K. Huang, Q. Guo, L. Zhang, P. Zhang, D. Xu, D. Zheng, X. Zhu, H. Wang, Y.-A. Chen, C.-Y. Lu, S. Han, and J.-W. Pan. 10-qubit entanglement and parallel logic operations with a superconducting circuit. *Physical Review Letters*, 119(18), Nov. 2017. ISSN 1079-7114. doi: 10.1103/physrevlett.119.180511. URL `http://dx.doi.org/10.1103/PhysRevLett.119.180511`.

[119] G. Struchalin, Y. A. Zagorovskii, E. Kovlakov, S. Straupe, and S. Kulik. Experimental estimation of quantum state properties from classical shadows. *PRX Quantum*, 2:010307, Jan 2021. doi: 10.1103/PRXQuantum.2.010307. URL `https://link.aps.org/doi/10.1103/PRXQuantum.2.010307`.

[120] M. Suzuki. Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Communications in Mathematical Physics*, 51(2):183–190, June 1976. doi: 10.1007/BF01609348.

[121] E. Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st annual ACM SIGACT symposium on theory of computing*, pages 217–228, 2019.

[122] E. Tang. Quantum principal component analysis only achieves an exponential speedup because of its state preparation assumptions. *Physical Review Letters*, 127(6), Aug. 2021. ISSN 1079-7114. doi: 10.1103/physrevlett.127.060503. URL `http://dx.doi.org/10.1103/PhysRevLett.127.060503`.

[123] A. G. Taube and R. J. Bartlett. New perspectives on unitary coupled-cluster theory. *International Journal of Quantum Chemistry*, 106(15):3393–3401, Jan. 2006. doi: 10.1002/qua.21198.

[124] K. Temme, S. Bravyi, and J. M. Gambetta. Error mitigation for short-depth quantum circuits. *Physical Review Letters*, 119(18), Nov. 2017. ISSN 1079-7114. doi: 10.1103/physrevlett.119.180509. URL `http://dx.doi.org/10.1103/PhysRevLett.119.180509`.

[125] A. V. Uvarov and J. D. Biamonte. On barren plateaus and cost function locality in variational quantum algorithms. *Journal of Physics A: Mathematical and Theoretical*,

54(24):245301, May 2021. ISSN 1751-8121. doi: 10.1088/1751-8121/abfac7. URL `http://dx.doi.org/10.1088/1751-8121/abfac7`.

[126] G. Verdon, M. Broughton, J. R. McClean, K. J. Sung, R. Babbush, Z. Jiang, H. Neven, and M. Mohseni. Learning to learn with quantum neural networks via classical neural networks, 2019.

[127] S. Wang, P. Czarnik, A. Arrasmith, M. Cerezo, L. Cincio, and P. J. Coles. Can error mitigation improve trainability of noisy variational quantum algorithms?, 2021.

[128] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, 12(1), Nov. 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-27045-6. URL `http://dx.doi.org/10.1038/s41467-021-27045-6`.

[129] Y. Wang, Y. Alexeev, L. Jiang, F. T. Chong, and J. Liu. Fundamental causal bounds of quantum random access memories. *arXiv preprint arXiv:2307.13460*, 2023.

[130] M. M. Wauters, E. Panizon, G. B. Mbeng, and G. E. Santoro. Reinforcement-learning-assisted quantum optimization. *Physical Review Research*, 2(3), sep 2020. doi: 10.1103/physrevresearch.2.033446. URL `https://doi.org/10.1103%2Fphysrevresearch.2.033446`.

[131] D. Wecker, M. B. Hastings, and M. Troyer. Progress towards practical quantum variational algorithms. *Physical Review A*, 92(4):042303, 2015.

[132] N. Wiebe, C. Granade, C. Ferrie, and D. Cory. Hamiltonian learning and certification using quantum resources. *Physical Review Letters*, 112(19), 2014. ISSN 1079-7114. doi: 10.1103/physrevlett.112.190501. URL `http://dx.doi.org/10.1103/PhysRevLett.112.190501`.

[133] N. Wiebe, C. Granade, and D. G. Cory. Quantum bootstrapping via compressed quantum hamiltonian learning. *New Journal of Physics*, 17(2):022005, Feb. 2015. ISSN 1367-2630. doi: 10.1088/1367-2630/17/2/022005. URL `http://dx.doi.org/10.1088/1367-2630/17/2/022005`.

[134] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, and J. Schmidhuber. Natural evolution strategies, 2011.

[135] B. Wu and D. E. Koh. Error-mitigated fermionic classical shadows on noisy quantum devices, 2023.

[136] J. Yao, M. Bukov, and L. Lin. Policy gradient based quantum approximate optimization algorithm, 2020.

[137] M. H. Yung, J. Casanova, A. Mezzacapo, J. McClean, L. Lamata, A. Aspuru-Guzik, and E. Solano. From transistor to trapped-ion computers for quantum chemistry. *Scientific Reports*, 4(1), 1 2014. doi: 10.1038/srep03589.

[138] B. Zhang, J. Liu, X.-C. Wu, L. Jiang, and Q. Zhuang. Dynamical phase transition in quantum neural networks with large depth. *arXiv preprint arXiv:2311.18144*, 2023.

[139] T. Zhao, G. Carleo, J. Stokes, and S. Veerapaneni. Natural evolution strategies and variational monte carlo. *Machine Learning: Science and Technology*, 2(2):02LT01,

dec 2020. doi: 10.1088/2632-2153/abcb50. URL `https://doi.org/10.1088%` `2F2632-2153%2Fabcb50`.